AGH University of Science and Technology in Krakow

Faculty of Electrical Engineering, Automatics, Computer Science

and Biomedical Engineering

DEPARTMENT OF APPLIED COMPUTER SCIENCE

University of Caen Normandy

Laboratory GREYC

MODELS, AGENTS, DECISION RESEARCH GROUP

PHD THESIS

Temporal Planning with Fuzzy Constraints and Preferences

AUTHOR: Krystian Jobczyk

SUPERVISORS: Prof. DHR Maroua Bouzid,
Prof. dr hab. Antoni Ligęza

Krakow 2017

Akademia Górniczo-Hutnicza w Krakowie

Wydział Elektroniki, Automatyki, Informatyki

i Inżynierii Biomedycznej

KATEDRA INFORMATYKI STOSOWANEJ

Uniwersytet Dolnej Normandii w Caen

Laboratorium GREYC

GRUPA 'MODELE, AGENCI, DECYZJE'

ROZPRAWA DOKTORSKA

Planowanie Temporalne z Ograniczeniami Typu Rozmytego i Preferencjami

AUTOR: Krystian Jobczyk

PROMOTORZY: Prof. DHR Maroua Bouzid,
Prof. dr hab. Antoni Ligęza

Kraków 2017

Acknowledgements

Dziękuję moim rodzicom za ich cierpliwość, wsparcie i gotowość do znoszenia moich co raz to nowych pomysłów i za wiarę w sens ich realizacji.

Dziękuję Profesorowi Antoniemu Ligęzie za Jego mądrość, naukowe i emocjonalne wsparcie, cierpliwe znoszenie mojego (trudnego momentami) indywidualizmu, a także za stworzenie niezwykłej atmosfery wolności badawczej, w której idee tej pracy mogły dojrzewać oraz za troskliwe poprowadzenie mojej rozprawy.

Dziękuję Jurkowi Karczmarczukowi–emigracyjnemu 'Wujowi' za Jego wszechstronne wsparcie, mnogość porad naukowych, życiowych i za namiastkę polskości na emigracji, a take za odważne czuwanie, by praca ta mogła zostać sfinalizowana.

Dziękuję moim drogim współtowarzyszom z pokoju 318/319: Michałowi, Szymonowi i Mateuszowi, Krzysiom: Kuttowi i Kluzie – z tym ostatnim w roli głównej i wyróżnionej – za stworzoną przez nich przyjacielską, a wręcz domową atmosferę pracy i wymiany myśli na wszystkie tematy świata. Krzysiowi Kluzie dziękuję ponadto za niezliczone rozmowy i żarty poza godzinami pracy, żywą pamięć i troskę i przeczytanie manuskryptu pracy, dzięki czemu uniknąłem wielu niedopatrzeń i błędów. Dziękuję także 'Profesorom Adrianom': Weronice i Markowi za przesympatyczne i prawdziwie intelektualne dyskusje o naszych doktoratowych i życiowych sprawach.

Dziękuję Szymonowi i Michałowi za Jego przyjaźń, wsparcie i sekundowanie z oddali moim postępom. Dziękuję Kubie za miłe chwile, związane często z Jego chęcią zrozumienia tego, co w pracy i specyfiki tego, czym jest informatyka.

Je voudrais remercier Maroua — ma directrice — pour elle aid de organiser mon sojour au debut et le financement de ma these et pour tout les experiences differents, que etaient l'ecole de la vie pour moi. Je voudrais remercier les membres du MAD pour les discussiones et remarques durant les runions du group de travail durant mon sejour a Caen, aussi.

Finallement, je voudrais remercier mes collegues de l'aumonarie avec Pere Nicalas (especialement: Charles-Eric, Funny, Paul, Jean Baptiste, Quantin) pour votre sympathie et pour les milliers des ide de l'organisation du tempts Caen et Herouville-St. Claire.

Ich möchte herzlich Hannes Leitgeb – meinen freundlichen Chef während meines Aufenthaltes in MCMP in München als ein Nachwuchswissenschaftler – für seine lebendige Unterstützung meiner informatischen Anspr̈che und seine glanzvolle Empfehlung bedanken. Ich möchte auch herzlich meinen Kollege Mario Huber fr seine lebendige Interesse für meine Dissertation und alle stimmulierte Gespräche und die wunderschöne Zeit während meines Aufentheltes in Vancouver im Sommer 2016.

Io voglio ringraziare anche Prof. Angelo Montanari della universita di Udine per le discussione durante ECAI'14 in Prague et per la corespondenzia scientifica. In particolare, ringrazio per una spiegazione delle relazioni tra subsistemi della logica di Halpern-Shoham e un linguaggio regolare comme l'inspirazione per una constructione teoretica del driver

hybrido per la pianificazione temporale.

Figure 1: Salvador Dali, La persistencia de la mamoria

———————— Pracę tę dedykuję:
————Moim Rodzicom, Szymonowi, Krzysiowi, Kubie————
————oraz Panu Doktorowi Wojciechowi Makarewiczowi ————

Contents

0.1 What Is the Thesis About? 10
0.2 Motivation 10
0.3 The Problem Formulation 11
0.4 Objectives of the Thesis 12
0.5 Structure and Content of the Thesis. 13
0.6 How to Read the Thesis? 15
0.7 A Short Justification of the Thesis Approaches 15

I Introduction 1: Temporal Planning as an Extension of Classical Planning. The AI Approach 21

0.8 Plan and Planning 23
0.8.1 Planning Domains, Plans, Actions and Planning Problems 24
0.9 Planning as Searching in Graphs 29
0.9.1 Planning as Forward Search 29
0.9.2 Planning as Backward Search 32
0.10 Planning as Satisfiability 33
0.10.1 Unit-propagation and Davis-Putnam Procedure 34
0.11 Planning with Temporal Operators 35
0.12 Planning Languages. 38
0.12.1 Linear Temporal Logic (LTL) 38
0.12.2 PDDL and PDDL+ 39
0.13 State of the Art and History of Research in this Area 40

II Introduction 2: Temporal Constraints, Preferences and Fuzzy Temporal Constraints 42

0.14 Temporal Constraints – a Brief Introduction 44
0.15 Qualitative Temporal Constraints 44
0.15.1 Point Algebra 45
0.15.2 Allen's Interval Algebra 45
0.15.3 Halpern-Shoham Logic 47
0.15.4 State of the Art 48

0.16 Quantitative Temporal Constraints, or Temporal Constrains Satisfaction Problem and its specifications . . . 49
0.16.1 Temporal Constrains Satisfaction Problem – a Detailed Specification 49
0.16.2 Simple Temporal Problem . . . 51
0.16.3 Preferences . . . 52
0.17 Two Depictions of Fuzzy Temporal Constraints of Allen's Sort . . . 54
0.17.1 Terminological Framework . . . 55
0.17.2 De Cock-Schockaert's School Approach to Fuzzified Allen's Relations 58
0.17.3 Fuzzy Allen's Relations in Ohlbach's Integral-based Depiction . . . 60
0.17.4 State of the Art and History of Research in this Area . . . 66

1 CONTRIBUTIONS: Temporal Planning Problems Approaches: Critique and Taxonomy 69
1.1 Temporal Planning with Fuzzy Constraints and Prefereces: Analysis of Selected Problems . . . 70
1.2 The Temporal Travelling Salesman Problem . . . 74
1.2.1 Informal Defining of TTSP . . . 74
1.2.2 Formal Definition of TTSP. . . . 75
1.3 Multi-Agent Schedule-Planning Problem . . . 79
1.3.1 A Multi-Agent Schedule-Planning Problem – a More Practical Depiction 79
1.3.2 General Formulation of a Multi-Agent Schedule-Planning Problem . . 81
1.3.3 Types of Temporal Constraints of PM-AS-PP . . . 82
1.4 Multi-Agent Schedule-Planning Problem *versus* Temporal Traveling Salesman Problem – Their Modeling and Representations . . . 85

2 Temporal Planning with Fuzzy Temporal Constraints and Preferences. The Real Analysis-Based Depiction 88
2.1 Introductory Remarks . . . 89
2.1.1 Motivation of Current Analyses . . . 89
2.2 Mathematical Foundation of the Convolution-based Approach to Fuzzy Allen's Relations . . . 90
2.2.1 Motivation of the Convolution-based Approach . . . 90
2.2.2 Terminological Framework of the Discussion . . . 91
2.2.3 Mathematical Foundation in Terms of Real and Abstract Analysis . . 93
2.2.4 Algebraic Aspects of the Convolution-based Fuzzy Allen's Relations . 99
2.3 Modeling of Fuzzy Temporal Constraints and Preferences . . . 101
2.3.1 Fuzzy Temporal Constraints in a New Representation . . . 101
2.3.2 Preferences in a New Representation. . . . 107
2.4 Fuzzy Temporal Constraints and Preferences in STRIPS and Davis-Putnam Procedure . . . 109
2.4.1 Fuzzy Temporal Constraints and Preferences in STRIPS . . . 109
2.4.2 Towards Temporal-Preferential STRIPS . . . 110
2.4.3 Fuzzy Temporal Constraints and Preferences in Davis-Putnam Procedure 113

2.5 Concluding Remarks . . . 116

3 Computable and programming-wise aspects of temporal planning with fuzzy constraints and preferences 117

3.1 Introduction . . . 118

3.1.1 Motivation of the Chapter . . . 118

3.1.2 Objective and Novelty of this Chapter . . . 118

3.2 Computational Aspects of Temporal Planning with Fuzzy Temporal Constrains and Preferences . . . 119

3.2.1 The Extended STRIPS in Use . . . 119

3.2.2 Extended Putnam-David Procedure in Use . . . 122

3.3 Programming-wise Aspects of Multi-Agent Schedule-Planning Problem . . . 127

3.4 Concluding remarks . . . 132

4 Temporal Planning with Fuzzy Temporal Constraints and Preferences. The Logic-based Depiction 133

4.1 Introduction . . . 134

4.1.1 Motivation of this Chapter . . . 134

4.1.2 Objectives and Novelty of this Chapter. . . . 135

4.1.3 The Problem Definition. . . . 136

4.2 Preliminaries . . . 136

4.3 Multi-Valued Preferential Logic (MVPL) . . . 138

4.3.1 Multi-Valued Preferential Logic (MVPL) – Syntax . . . 139

4.3.2 Interval-based Semantics for MVPL. . . . 139

4.4 Multi-Valued Preferential Halpern-Shoham logic (PHS) and its Subsystem $\mathrm{PHS^L}$ 141

4.4.1 Multi-Valued Preferential HS-logic (PHS). . . . 141

4.4.2 Subsystem $\mathrm{PHS^L}$ for Preferences and (Delayed) Actions. . . . 142

4.4.3 Modeling of the Salesman Situation from PTSP-problem. . . . 143

4.5 Conclusions . . . 145

5 A General Method of the Hybrid Controller Construction for Temporal Planning with Preferences 146

5.1 Introduction . . . 147

5.2 Motivation of Current Analyses . . . 147

5.2.1 Objectives and Novelty of this Chapter. . . . 147

5.3 Preliminaries . . . 148

5.4 Problem Formulation and a General Algorithm of the Controller Construction 150

5.5 The Controller Construction for Temporal Planning . . . 151

5.5.1 From Triangulation to the Finite Transition System (FTS) . . . 151

5.5.2 From FTS System to the First Büchi Automaton . . . 153

5.5.3 From LTL and $HS^{L,D}$ to the Second Büchi Automaton . . . 153

5.5.4 Product Automaton $\mathcal{A}_{FTS} \times \mathcal{A}_{LTL,HS^D}$ . . . 155

5.6 Part II: Implementation . . . 156

5.6.1 The Robot Environment and its $LTL \bigcup HS^{D,L}$-specification 156
5.6.2 From $LTL \cup HS^{L}$ to the Second B́uchi Automaton 158
5.6.3 The Product Automaton 159
5.6.4 PROLOG-description of the Product Automaton 163
5.7 Conclusions 163

6 Towards a Synthesis 165
6.1 Integral-based Approach as a Support of the Logical Approach 165

7 Conclusions 169
7.1 Concluding Remarks 169
7.2 Directions of Future Research 170
7.2.1 List of the Papers of the Author of the Thesis with a Referrence to its Content. 171

A Annexe 1 – Fuzzy Logic of Integrals for Allen's relations 173
A.0.2 Requirements of the Construction and Notational Remarks. 173
A.0.3 Syntax. 174
A.0.4 Semantics. 175
A.1 The Pavelka-Hajek-style Completeness and Some Metalogical Properties of FCL-FAIR 177
A.2 FCL-FAIR in Finite Models 178

A Annexe 2 – Fuzzy Allen's relations in a convolution-based depiction in algorithms. 180

A Annex 3 – The STPU Solution in the Integral-based Specification 184
A.0.1 Simple Temporal Problems with Uncertainty (STPUs) 184
A.0.2 STPPU, or Simple Temporal Problem under Uncertainty with Preferences 185
A.0.3 Integrals and Specification of Solution of STPU's 186

A Annexe 4 – Metalogical properties of Preferential Halpern-Shoham Logic (PHS) 191
A.1 A detailed presentation of fibred semantics 191
A.2 Some Metalogical Results: Model checking Problem for $\mathrm{PHS}^{\mathrm{L}}$ and Satisfiability 192
A.3 Satisfiability Problem. 193
A.4 State of the Art 194

A Annexe 5 – Proofs of Some Geometric Properties of Robot Environment 196
A.1 Proof of Some Facts 196

A Annexe 6 – Planning as Graph-Search – More Advanced Algorithm 198
A.0.1 Graph Planning and its Short Analysis 198
A.1 Some Formal Properties of STP. 202

A Annexe 7 – Lindstrøm Characterizing Theorem and its Simplified Proof 205
A.1 Terminological Background 205
A.1.1 Languages 205
A.1.2 Structures 205
A.1.3 Two Further Definitions 206
A.1.4 Theorems 206
A.2 Proof of Lindstrøm Theorem 207

A Annex 8–The Chosen Mathematical Concepts Used in the Thesis 209
A.1 Measure theory 209
A.2 Probability theory 209
A.3 Algebra 211

Abstract.This chapter provides an overview of this PhD-thesis: the thesis motivation, its objectives and its structure. Finally, some road map how to read this thesis is added. All these points are prefaced by explanatory remarks what is the main thesis focus.

0.1 What Is the Thesis About?

This thesis is aimed at the discussing of *temporal planning with fuzzy constraints and preferences.* Planning constitutes one of the form of rational behavior and reasoning – in particular. Intuitively, we often refer *planning* to a deliberation process that chooses and organizes *actions* in order to achieve goals that are desired or required [1, 2, 3]. Actions are executed in a given *domain.* Making actions often modifies initial domains. For example, an action 'move' may change a robot/satellite position.

A natural extension of (classical) planning is *temporal planning.* Informally speaking, temporal planning may be viewed as classical planning involved in 'timing'. 'Timing' can be understood in many ways: as a duration of action performing or as temporal constraints imposed on the action materialization. *Temporal planning* is aimed at different issues (see: [3]). The most typical are the following ones:

- time optimization of action execution,
- types of temporal constraints that may be imposed on action execution,
- representation of temporal constraints,
- construction of plans which respects temporal constraints that are required.

In general, temporal constraints are divided into two classes: *the qualitative and quantitative temporal constraints.*

In order to reinforce reality of investigations, temporal planning is often considered together with a new component called 'preferences'. They introduce some piece of rationality to temporal planning. Preferences may be imposed on action execution, task performing or on a choice of different components such as: situations, solutions, etc. Unfortunately, temporal planning sometimes forms an acting under uncertainty. The notion of 'uncertainty' may refer to different situations in temporal planning. It may mean that our preferences were set imprecisely or that temporal constrains – imposed on action performing – is not rigid. All these situations constitute a subject of *temporal planning with fuzzy constrains and preferences.*

0.2 Motivation

Nowdays, there exist at least three different approaches to planning in Artificial Intelligence and in close-related branches of computer science. One of them is a planning paradigm "planning as satisfiability of formulas' - see: [1, 2, 3]. The main alternative approach to planning via graph search was introduced and broadly developed in [4, 5, 6] and recently supported by some techniques of the graph-plan analysis from [7, 8, 9]. There is also a couple of typical problems - naturally associated to planning. One of such problem is an old Hamiltonian optimization problem – commonly known as the *Traveling Salesman Problem* (TSP) - solved by M.M. Flood in [10] and developed by Menger in [11] I. Heller in [12]. This problem may be briefly characterized as follows: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

An alternative problem – associated to planning (and scheduling) is the so-called *Multi-Agent Problem* (MAP). This problem – or even better – a class of similar problems may be represented by the so-called *Nurse Rostering Problem* – discussed in [13, 14, 15, 16] and specified as follows: having a set of actions and agents find a schedule that satisfies all constraints imposed on these actions.

Meanwhile, a taxonomy of temporal constraints (such as these in TSP or MAP) is heterogeneous. In fact, as mentioned, they are usually divided into two groups: the *qualitative* and the *quantitative* ones. The quantitative ones are usually rendered in terms of *Temporal Constraints Satisfaction Problem* and *Simple Temporal Problem* – as its specification when numeric constraints are restricted to a single interval – due to [17, 18, 19, 20].

Unfortunately, these planning problems, planning paradigms and the taxonomy of temporal constraints are involved in different difficulties and they show some essential shortcomings or lacks. Namely, they may be specified as follows.

PlanParadigms:
1. Neither methods of the the first paradigm (such as STRIPS), nor of the second one (such as Davis-Putnam procedure) seem to have enough sufficient expressive power to exploit them in contexts of temporal planning with different constraints and preferences.
2. At least, an expressive power of these methods is not clear.
3. Finally, these planning paradigms – are rather considered purely methodologically as a reservoir of planning methods – developed, somehow, independently of concrete planning problems as their subject 'specification'.

PlanProblems:
1. No temporal extensions of these problem (MAS and TSP) are known and no approach to a construction of a taxonomy of temporal extensions of MAS and TSP is proposed.
2. In a consequence, no approach to modeling and representation of these extensions are known in a specialist literature.
3. Finally, majority of research on TSP (partially on MAS, as well) refers to these problems as to the discrete ones (for example, in a graph-based representation), putting aside a real time aspects in their solving.

TempConst:
1. The representation of temporal constraints does not overcome a dualism: 'quantitative-qualitative' and that there is no 'bridge' between a conceptualization of quantitative and qualitative temporal constrains.
2. In addition, majority of works refer to Allen's algebra as a sufficient conceptual basis to represent temporal constraints. Meanwhile, these fuzzy temporal constraints – integrated with preferences in particular – require a more sophisticated representation (than Allen's algebra). This new representation should be sensitive for combined formulas (representing both temporal constraints and preferences).
3. Finally, approaches to the representation of Allen's algebra oscillate more around meta-logical features of Halpern-Shoham logic such as PSPACE-completeness, decidability etc. than around a question: 'How to make Allen's interval relations computably useful and operationally explorable?'.

It seems that overcoming these difficulties and dualism towards a more holistic approach may support a more successful and general investigations in a temporal planning. This conviction constitutes a motivation factor of the thesis analysis as well.

0.3 The Problem Formulation

Due to above observations – this PhD-thesis is aimed at solving the following research problems:

1. *How to introduce a simple taxonomy of problems in temporal planning and scheduling with fuzzy constrains and preferences, which satisfies the following conditions:*
   - *it is (relatively) exhaustive,*
   - *it contains a precise description of basis problems of each,*

2. *How to manage and model fuzzy temporal constrains and preferences – due to this taxonomy – in order to either:*
    - *grasp a computational side of them or,*
    - *extend an expressive power of planning methods and tools or,*
    - *give new formal tools for plan monitoring.*

0.4 Objectives of the Thesis

According to these motivation factors – the objectives of the thesis may be summarized as follows:

Goal: The main goal of the thesis is a depth-analysis and exploration up-to date the conceptual tissue of temporal planning with fuzzy constraints and preferences to:

- propose a new mathematical (formal) foundations for the components of temporal planning with these constraints and
- elaborate new ways of representation and modeling of these components.

This main goal may be decomposed in the following subgoals (as their conjunction) – chronologically ordered as below:

Go1 To identify main difficulties and weaknesses of current approaches to temporal planning and to representation of temporal constraints and preferences.

Go2 To propose new temporal version of Traveling Salesman Problem and Multi-Agent Problem as a basis for a subject-specification of temporal planning.

Go3 To give a mathematical foundation for temporal constraints with fuzzy Allen's relations in a distinguished role in terms of algebra and real and abstract analysis.

Go4 To propose a new hybrid approach to fuzzy temporal constraints and preferences on a base of the convolution-based representation of fuzzy Allen's interval relations in the context of Temporal Multi-Agent Problem.

Go5 To incorporate this new approach to propose new fuzzy temporal and preferential extensions of STRIPS and Davis-Putnam procedure.

Go6 To propose a logic-based approach to fuzzy temporal constraints and preferences in the context of Traveling Salesman Problem.

Go7 To incorporate this new approach in plan supervising – for the construction of the hybrid plan controller.

Go8 To propose a synthesis of these two approaches: the convolution-based and the logic-based one via complementation the construction of the plan controller by a convolution representation of the agent's move trajectories.

By the way, the computational and programming-wise (in terms of PROLOG-solvers) aspects of Multi-Agent Schedulo-Planning Problem are undertaken in the thesis.

As mentioned, this PhD-thesis is amied at the representing and modeling of different components of temporal planning with fuzzy constraints and preferences. It is also focused on in-depth-analysis up-to-date of teh conceptau tissue of these problems. Therefore, it is focused neither on finding optimal solutions of the problems from this area, nor on experimental results and analysis from this area.

0.5 Structure and Content of the Thesis.

Although considerations of the thesis oscillate around an issue of temporal planning with different components and around methods of their modeling, contributions of this PhD-work may be clearly divided into two complementary parts.

1. **Part I (of Contributions)** is aimed at:
   - *representing and modeling* temporal planning and fuzzy temporal constraints and preferences in terms of convolution-based models and
   - *investigating* some computational and programming-wise aspects of this convolution-based approach.
2. **Part II (of Contributions)** is aimed at:
   - *representing* components of temporal planning – preferences, temporal constraints – in terms of some Multi-Valued Halpern-Shoham logic,
   - *modeling* them by means of a newly proposed interval-based fibring semantics, and
   - *putting forward* a general method of the hybrid plan controller construction exploiting the proposed approach.

The detailed content of further thesis chapters is the following:

Introduction 1 . This chapter forms a conceptual and (partially) historical introduction to issues of temporal planning as a unique extension of classical planning. Classical planning is described in different paradigms. In particular, classical planning as graph-searching (for example, as based on STRIPS) and as satisfiability (via Davis-Putnam procedure) were discussed. Finally, temporal planning is presented as an extension of classical planning by temporal aspects of acting.

Introduction 2 . This chapter forms an introduction to temporal and fuzzy temporal constrains and and their taxonomy. It also describes preferences as a separate component of temporal reasoning and temporal planning. Fuzzy temporal constraints are divided into two classes: the quantitative and the qualitative ones. The quantitative temporal constraints are briefly discussed in terms of Constraints Satisfaction Problem and its specification in the so-called *Simple Temporal Problems* (STP). The qualitative fuzzy temporal constraints – the main focus of this chapter – are properly discussed in terms of fuzzy Allen's relations. Two different approaches to their representation are presented here: Ohlbach's integral-based depiction and DeCock-Schockaer's depiction in terms of relational calculus and t-norms.

Contributions. Chapter 1. This chapter has an intermediate character between 'Introduction 1' and 'Introduction 2' and further parts of the PhD-thesis. Different difficulties of earlier approaches to temporal planning and fuzzy temporal constraints are detected and briefly discussed. One of the difficulty is a lack of a subject-specification of temporal planning, which is usually seen in a more methodological way. It forms a motivating factor to propose an outline of a small taxonomy of a subject-problems for temporal planning. Two classes of problems are introduced: the problems of the class of *Temporal Traveling Salesman Problem* (TTSP) and the problems of Multi-Agent schedule-Planning Problem (MA-SP-P). Both the paradigmatic problems (TTSP and MA-SP-P) are also defined in detail. Finally, some hints how to represent and model them are put forward in this chapter.

Contributions. Chapter 2. This chapter introduces a new mathematical approach to fuzzy temporal constrains and preferences. At first, fuzzy Allen's relations are represented by norms of the appropriate convolutions of the Lebesgue integrable functions – in a polemic reference to Ohlbach's ideas. Secondly, a new holistic approach to fuzzy temporal constraints – on a base of the convolution representation of fuzzy Allen's relations – is elaborated. This new holistic approach forms a combination of the quantitative and the qualitative fuzzy temporal constraints. The first ones are the constraints of MA-SP-P. The are encoded in the appropriate fuzzy intervals. The qualitative ones are just fuzzy Allen's relations imposed on these fuzzy intervals.

Next sections of the chapter present the temporal and preferential extensions of STRIPS and of Davis-Putnam procedure in a theoretic depiction. In addition, some metalogical features of the extensions are also discussed.

The qualitative temporal constraints are represented by Allen's relations and they are imposed on the quantitative ones. This combination allows us to introduce a new definition of *fuzzy temporal constrains* and *preferences* on a base of the last one.

Contributions. Chapter 3. Investigations of this chapter forms a conceptual continuation of investigations of chapter 2 and they refer to computational and programming-wise aspects of fuzzy temporal constraints and thero representation. At first, the convolution-based depiction of fuzzy Allen's relations is applied to STRIPS and Davis-Putnam procedure in the appropriate temporal and preferential extension. Secondly, the PROLOG-solvers for chosen cases of the Multi-Agent Schedule-Planning Problem are presented. Analyses of this chaper are carried out in the subject context of Multi-Agent Schedule-Planning Problem.

Contributions. Chapter 4. This chapter addresses an alternative, algebraic-logical approach to representation of temporal constrains and preferences. These components are rendered in terms of a new Multi-Valued (Preferential) Halpern-Shoham logic. A 'fuzziness' is introduced here by preferences. This formal system is further interpreted in some interval-based fibred semantics. It allows us to consider combined formulas representing both preferences and actions – temporally constrained. Investigations of this chapter oscillates around Traveling Salesman Problem and its modeling.

Contributions. Chapter 5. This chapter describes a general method of the hybrid plan controller construction and it extends a purely theoretic investigations of chapter 4 towards an application area. The controller construction runs as follows.

1. At first, the robot motion environment is specified in Linear Temporal Logic (LTL) extended by Halpern-Shoham Logic (HS).
2. This LTL+HS-description is encoded by the appropriate Büchi automaton and it represents a required, planned situation.
3. Next, the second Büchi automaton is constructed for a real situation of the robot task performing.

These two automata form a construction basis for their product automata. Its representation in terms of PROLOG plays a role of a desired plan controller.

Contributions. Chapter 6. This chapter describes an attempt of a synthesis of earlier approaches to fuzzy temporal constraints and preferences. It is discussed here how the analysis-based and the logic-based representations might complement each other in the plan controller construction. For example, trajectories of agent moves in a logic-based description may be interpreted as the appropriate functions in Sobolev spaces.

Contributions. Chapter 7. This chapter contains concluding remarks and announces a promising direction of a future research.

Appendixes. The thesis contains also 8 Appendixes. Appendixes 1-7 contain more advanced results from a thematic scope of the thesis, such as metalogical features of fuzzy logic systems for Allen's relations. Appendix 8 contains a couple of mathematical definitions used in the proper body of the thesis.

0.6 How to Read the Thesis?

Contributions of the thesis – as it has been signalized – do not have any simply linear structure, but rather a *bi-linear*. It follows from the fact that the proposed taxonomy of temporal planning problems may be viewed as containing two classes of problems. The first class contains problems of Multi-Agent Schedule-Planning Problem-type. (M-AS-P). The second one – problems of Temporal Traveling Salesman Problem-type (TTSP). Problems of the first class show to be better graspable in analytic terms of the convolution-based approach. Problems of the TTSP-type – show to be suitable to be represented more logically, in terms of Preferential Halpern-Shoham logic.

In consequence, we have the following two lines of reasoning in the thesis.

A **The first line** leads from the chapter 'Detecting difficulties' by chapter 1 (presenting a convolution-based approach to fuzzy temporal constraints) up to chapter 2 (presenting computational aspects of the convolution-based approach in temporal planning).

B **The second one** leads from the same chapter 'Detecting difficulties' by chapter 3 (describing fuzzy temporal constraints and preferences in terms of Preferential Halpern-Shoham logic) up to chapter 4 (describing a general method of the plan controller construction as an implementation of chapter 3).

These two reasoning lines are visually presented in Figure 2.
For that reason, chapter 3 *should not* be considered as a continuation or extension of investigations of chapter 1 (or chapter 2). In fact, it presents an alternative and complementary approach to approaches presented in chapters 1 and 2.

0.7 A Short Justification of the Thesis Approaches

Two-dimensionality of the thesis analysis has just explained. It remains to justify a mathematical thesis approach and considering of temporal planning in mixed contexts.

Justification for mathematical approach, but against 'mathematism'. An idea to exploit mathematical tools and methods in the thesis approach stems from the author's belief that problems of AI – problems of planning in particular – should be operationally and computably graspable and expected results should be precisely measurable. This postulate seems to be feasible in a framework of any mathematical approach. The same mathematical approach may deliver a formal and precise conceptualization in a framework of a logical representation (of actions, temporal constrains or preferences). Finally, a better understanding of logical and mathematical foundations of temporal planning may be useful for AI-appliers – as it allows us to better understand what we do in a broad area of temporal planning.

Nevertheless, this work does not constitute any manifesto of 'mathematism'. In fact, *purely* mathematical approaches – based on a mathematical idealization – are often insufficient in temporal planning or they have reached limits of applications. In addition, a purely mathematical description seems to sometimes deliver too 'static' framework to describe the whole dynamism of temporal planning, its problems and their specification. Some examples for this fact may be found in this work as well.

Justification of considering temporal planning in mixed contexts. There are (at least) two reasons against considering temporal planning as a separate activity in AI.

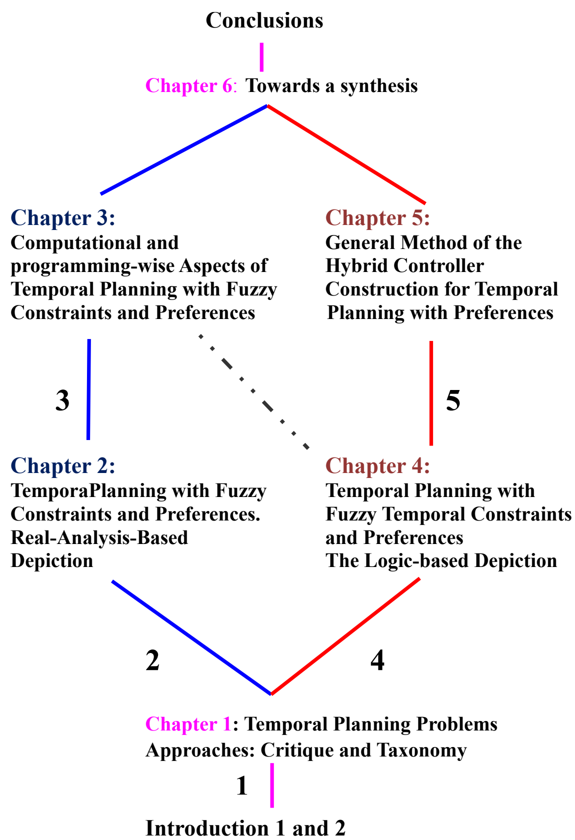


Figure 2: A road map of the thesis. Numbers 1 - 6 presents a chronology of thesis chapters.

1. At first, temporal planning (classical planning as well) may be much better emphasized and elucidated in mixed contexts – together with scheduling and plan monitoring.

2. Secondly, temporal planning seems to be especially suitable to be combined with many forms of temporal reasoning such as scheduling or plan supervising.

3. Finally, temporal planning itself forms a slightly theoretic concept and it just requires some form of complementation in order to become a more realistic acting.

For that reasons, a conceptual framework was not only determined by the single question 'Which action to choose?', but also by questions: 'How to associate a given set of actions to a schedule?', 'How to supervise performing of a given plan?' or 'How to represent fuzzy temporal contexts imposed on a given set of actions?'.

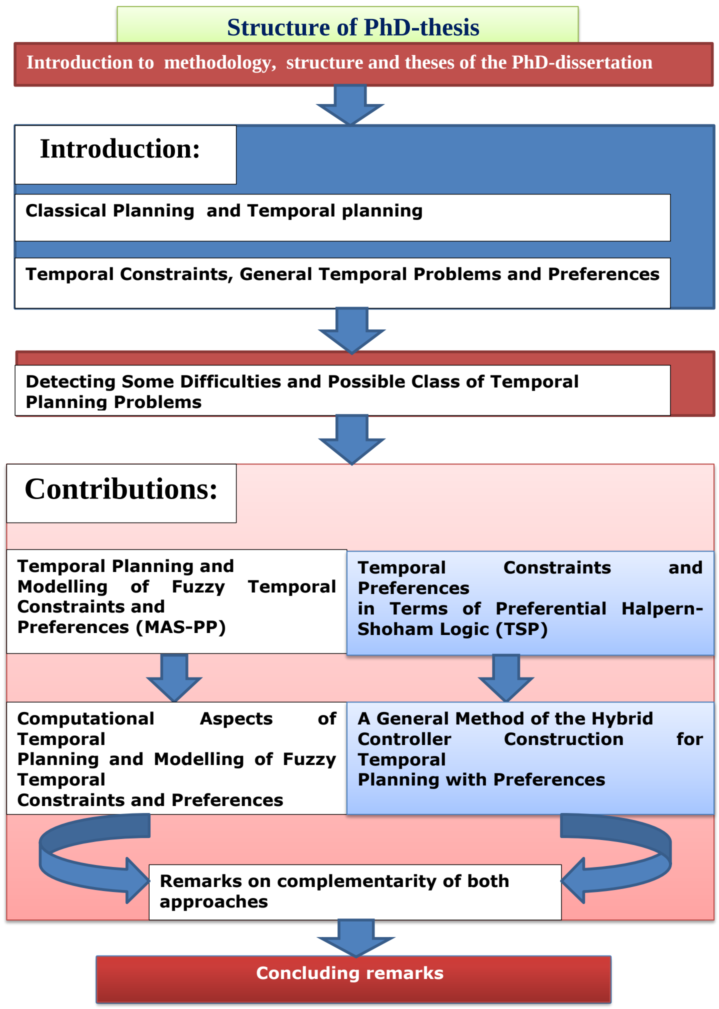


Figure 3: The structure of the PhD-thesis

List of Figures

1 Salvador Dali, La persistencia de la mamoria 4
2 A road map of the thesis. Numbers 1 - 6 presents a chronology of thesis chapters. 16
3 The structure of the PhD-thesis 17
4 Visual presentation of a 'conceptual map' for temporal planning 24
5 A crane in a block world 27
6 Goal g, applicable actions and a new goal obtained via inverse $\gamma^{-1}(g, a)$ for blocks A,B,C. . . . 28
7 STRIPS-method for a simple situation of acting in a block world. Detected preconditions are only applicable to actions: PICKUP(A) and STACK(A, B), so – due to STRIPS-algorithm – these actions determine the required plan. 31
8 Goal g, applicable actions and a new goal obtained via inverse $\gamma^{-1}(g, a)$ for blocks A, B, C. . . 33
9 Davis-Putnam procedure for Example 66 36
10 Davis-Putnam procedure for Example 7 36
11 Planning situation with 3 point-actions 46
12 13 basis Allen relations between intervals. 46
13 Composition of any two basic relations of Allen-type. The picture may be found under: http://www.ics.uci.edu/alspaugh/cls/shr/allen. 47
14 A Salesman Traveling between French cities 48
15 Temporal constraint network for the Titanic catastrophe 51
16 A poinwise Kripke-based semantics for preferences. In a world t (R-accessible from u) is satisfied all (ϕ), what is possible in u. 54
17 13 basis Allen relations between intervals. 55
18 Core, support and x-coordinates of a fuzzy interval 56
19 Difference between Riemann and Lebesgue integrals. 57
20 A visual presentation of relations $L^{\leq}_{(\alpha,\beta)}(., b)$ and $L^{\preceq}_{(\alpha,\beta)}(., b)$. 60
21 A point 'p' before a fuzzy interval. A visualization of a point-interval 'before' relation. (Interval j is considered here as dependent on the argument $t - x$ – according to the arrow direction on the diagram. The sense of this depiction will be explained in detail in chapter 2 of 'Contrubutions'). 61
22 A point 'p' is blown up to the new fuzzy interval $i(x)$. 62
23 Fuzzy 'before'-relation of the interval-interval type dependently on 'size' of intervals. This relation may be represented by the dark lines. 64

1.1 The visualization of the TTSP for Visiting Professor in terms of the proposed conceptualization. 77

2.1 The main idea of defining fuzzy temporal constraints and preferences. 103
2.2 Intervals (d_i, z_j) and function $f((d_i, z_j)$ (dark line on a picture) determining a fuzzy interval. In fact, this picture illustrates a more general situation, when $\sum_{i=1}^{m} X(i)_{d,z,n,a}$ is not necessary a constant function and f is not necessary continuous. 105

2.3 Fuzzy temporal constraints (FTC) determined by the convolution for Allen's 'meet'-relation over a common (blue) region HTC. 106
2.4 The blue common triangle represents HTC. FTC are represented by a fuzzy region 'around' the triangle – determined by the broken line. Finally, preferences are represented by a common region of both intervals and the pink region cut by the preferential linear function y (the dark line). The interval $i(x)$ encodes (the constraints of) the planned situation. The normalized values of F-functions are found on the vertical axis.The interval $j(x)$ encodes the real situation. 108

3.1 Actions a_5 and a_6 with temporal constrains imposed on them (the blue region) and fuzzy temporal constrains determined by the convolution (black broken line). 120
3.2 Davis-Putnam procedure for a basic planning case 123
3.3 Actions `Load(B)` and ¬ `Load(B)` with associated values 0.8 and 0.2 (*resp.*) 124
3.4 Davis-Putnam procedure in a fuzzy case. In this case we do not reject `Load(B)`∧¬ `Load(B)`, which gives us a model μ with a true degree v = 0.2. 125
3.5 Actions `Load(B)` and ¬ `Load(B)` with associated values 0.8 and 0.2 (*resp.*) and the preferential line, which orders to reject ¬ `Load(B)`. 126
3.6 Fragment of solution of the initial example with fuzzy temporal constraints and preferences. The red line leads to the rejected solution. The green ones – to the accepted ones. 126

4.1 Illustration of the Traveling Salesman Problem in the considered case. 136
4.2 An example of IBIS-system 137
4.3 $\mathcal{U}$nr is the unraveling of $\mathcal{F}$ 138
4.4 An illustration of *IBIS* from Example 27 138
4.5 Two discrete intervals I_1 and I_2 having 10 common points with an example of computing a value $\|I_1 \precsim_K I_2\|$ for some K. 140
4.6 A fibring mapping which should not necessary be a function 143

5.1 Fragment of Büchi automaton with states s_0, s_1 and s_2 over an alphabet Σ. 149
5.2 An outline of the procedure of the hybrid plan controller construction 151
5.3 An example of a triangulation of some polygonal robot environment 152
5.4 The polygonal environment of the robot motion with 4 rooms. The blue broken line illustrates the planned trajectory of the robot move from a room no. 1 to the room no. 4. The red one illustrates the deviated trajectory of the robot move. 157
5.5 Fragment of the Büchi automaton with states for a closure of the LTL-formula Take(A). . . . 159
5.6 Fragment of the Büchi automaton with states for a closure of the LTL-formula $Take(A) \rightarrow \langle L\rangle go(R_3)$. 160
5.7 Fragment of the Büchi automaton with states for a closure of the LTL-formula a$Move(R_1^A, R_3)$.160
5.8 Fragment of the Büchi automaton with states for a closure of the LTL-formulas $Take(A)$ and $Move(R_1^A, R_3)$. 161
5.9 Fragment of the Büchi automaton for $Move(R_1^A, R_3)$ and for a fragment of a formula $take(A) \rightarrow \langle L\rangle go(R_3)$. 161
5.10 Fragment of the Büchi automaton for the real task performing for the closure of LTL-formula $Move(R_1^A, R_2)$. 162

6.1 Robot polygonal environment (with rooms A and B) as a Hilbert space of Lebesgue integrable trajectories (green lines) and control functions (red line). 166

A.1 An example of STPU 185
A.2 An observation task of a satellite as an example of STPPU 186
A.3 STPU for the sub-problem of the Multi-Agent Schedulo-Planning Problem 187

A.4 Functions Ax and Bx (partially) defining the fuzzy intervals i and j 188
A.5 An outline of the function $\frac{2x^3}{2x^3-3ax^2}$ 189
A.6 The fragment of a graph of the function $F(x-B)$, which we are interested in – as a visual representation of our problem solution. 189

A.1 $\mathcal{U}$nr is the unraveling of $\mathcal{F}$ 192

A.1 The second step of the Graph-plan algorithm search 201
A.2 The Salesman's traveling between cities in Germany 203

Part I

Introduction 1: Temporal Planning as an Extension of Classical Planning. The AI Approach

Introduction 1: Temporal Planning as an Extension of Classical Planning. The AI Approach

0.8 Plan and Planning

Classical Planning [3], as considered in *Artificial Intelligence* (AI) and Knowledge Representation (KR), forms a broad research domain located at the intersection of such areas as: Temporal Reasoning, State-Space Search, Graph-Search and Heuristic Search. It may be briefly specified by a title of the Lifschitz's work [2] as *reasoning about actions and plans.*[1] In fact, (classical) planning[2] consists in finding a finite sequence of actions that transform an initial state into a final one, called a *goal state*. Each action has a predefined set of conditions which must be satisfied in order to execute it. These conditions are commonly called action *preconditions*. Each action causes some effects of its execution.

From a purely methodological point of view, classical planning forms a heterogeneous research domain – as it exploits a variety of (sometimes) mutually incompatible approaches and methods. The main methodological paradigms of classical planning are: the so-called *Satisfiability-based* approach[3], planning as search (in particular: in graphs) and planning via Markov Decision Processes (MDD). The satisfiability-based paradigm is based on a logical specification of planning domain and goals and is closely related to planning as satisfiability. From operational point of view, it may be supported by (heuristic) graph-search methods. In contrast, the Markov's process-based planning oscillates around such concepts as: utility functions, probability and optimization problems[4].

Temporal planning constitutes some further extension of classical planning, where temporal aspects of objects, events and actions are also considered. Informally speaking, temporal planning can be considered as activity close to classical planning, but time concerning issues, such as execution times and temporal constraints, are also taken into account.

A somewhat rough conceptual map of the planning domain sub-areas – as considered by contemporary AI – is presented in Fig. 0.8.

The term 'timing' in temporal planning may be rendered in many ways: as a duration of action performing, as a temporal 'distance' between intervals associated to actions or – as temporal constraints imposed on the action performing. Nevertheless, temporal aspects of planning usually refer to the fact that actions are assigned some duration (as a time of their execution).

Independently of these (sometimes sophisticated) differences, a general conceptual planning framework is determined by the following components (see, for example, [3]):

A actions,

B planning domains,

C plans,

D planning problems,

E initial states and goal states.

[1] This perspective of planning was presented in papers such as [21, 22, 23]. (Classical) planning as satisfiability was defined in[24], as a graph analysis in Blum's works: [4, 5, 6]. Classical planning in a more automated-based approach was widely discussed in some extensions in two monographs of Ghallab-Nau-Traverso in [3] and Harisson in [25].

[2] Sometimes – as in [3] – classical planning is considered as a slight generalization of the so-called *set-theoretic* planning. The difference between them can be grasped in the perspective of languages for their representation. In set-theoretic planning a finite set of propositions (atoms) of a propositional calculus can play the role of a planning language. In classical planning, we admit a language of a predicate calculus as a planning language – often restricted to a finite set of formulas.

[3] In [3], this approach is rather called 'Model-Checking approach'. Meanwhile, the model-checking problem can be defined as follows: given a model of a system, check whether this model meets a given specification. Model-checking was invented and described in the 80's by E. Clark and A. Emerson in [26, 27, 28] and introduced to planning research in [29, 30].

[4] In planning via Markov Decision Processes, a planning domain is modeled as a transition system, where probabilities are assigned to state transitions, goals are represented by the so-called *utility functions* and planning is considered as an optimization problem – see: [3], pp. 379-386. Whereas Markov decision processes are known since 19th century between mathematicians, they were incorporated by researchers in AI in the 90th in works of A. Cassanda and T. Dean: [31, 32, 33].

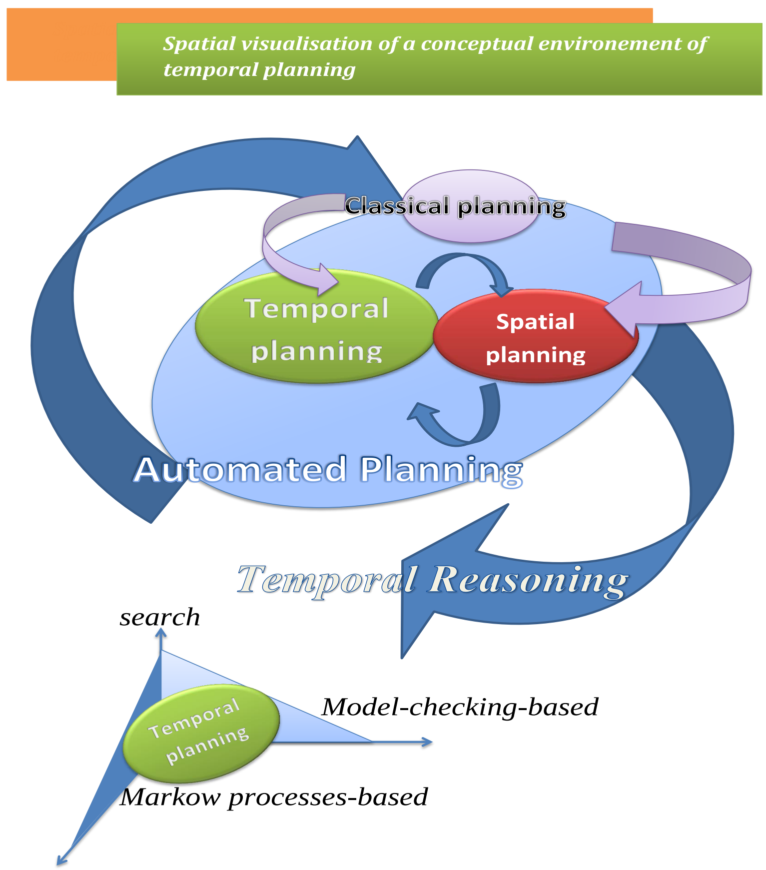


Figure 4: Visual presentation of a 'conceptual map' for temporal planning

These components are often defined in the appropriate planning language – both a propositional one or a predicate one, as well as the modal ones. It allows us to represent initial and final states as propositions of a given planning language. A detailed description of the planning components will be provided below.

0.8.1 Planning Domains, Plans, Actions and Planning Problems

The core concept of planning (in both classical and temporal approach) is the notion of *action*. Actions – as atomic components of plans are somehow associated to planning domains and planning goals. Planning domains are defined by a set of states $\mathcal{S}$, set of actions $\mathcal{A}$ and by possible *observations* that refer to the domain components. Some of the states can be distinguished as *initial states* of the domain. In order to move between states, a *transition function* is defined. Detailed definitions of these concepts are as presented below.

Actions and planning domains

Assume that a non-empty set $\mathcal{A}$ (of entities called later 'actions') is given. In *classical planning* each action $a \in \mathcal{A}$ forms a triple:

$$a = (\mathsf{name(a)}, \mathsf{precond(a)}, \mathsf{effects(a)}) \tag{1}$$

where $\mathsf{name(a)}$ is an action name, $\mathsf{precond(a)}$ – the set of action *preconditions* and $\mathsf{effects(a)}$ – a set of action *effects*[5]. Preconditions are such conditions which must be satisfied to perform a. They could be determined as such ones by the effect of another actions prior to the execution of a. Effects are all changes (new states) in the action performing environment, all results of the action execution, see: [34][6].

In *temporal planning*, each action a has its own *duration*, which may be unlimited or bounded. The action duration may be represented by initial and final points of its duration or by the whole time interval (open or closed) – associated to a.

Example 1 *Assume a trivial planning situation of a robot r in a one-block world (B). Assume that r takes the block B from a room P_j to the corridor $Corr$. This action specification may be rendered as follows (in some propositional fragment of a descriptive language; for intuition, mnemotechnical notation P^O denotes the fact that object O has location P).*

locations: (room P_j, corridor $Corr$)
action: (robot r takes a block B from P_j to the corridor $Corr$)
$take(r, B, P_j, Corr)$

preconditions: non-empty(P_i), empty($Corr$):
$P_j^B, \neg Corr^B$[7]

effects: empty(P_i), non-empty($Corr$):
$\neg P_i^B, Corr^B$.

Assume that some non-empty domain $\mathcal{S}$ of states (the so-called state-space; in fact $\mathcal{S} \subseteq 2^L$ constitutes a set of definitions of states in some language L ([3], pp.20-21) is given and let $s \in S$ be some arbitrary state. We say that an action a is *applicable* to the state s if and only if $\mathsf{precond(a)} \subseteq s$ (see: [3], p. 20)).

Definition 1 ***(Domain of planning)*** *([3], p. 31). Let $\mathcal{L}$ be a first-order language that has finitely many predicates and symbols*[8]*.*

A nondeterministic planning domain is a tuple $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \gamma \rangle$, such that:

1. *$\mathcal{S} \subseteq 2^{\mathbf{B}(\mathcal{L})}$ is a set of states, where $\mathbf{B}(\mathcal{L})$ is the set of ground atoms of $\mathcal{L}$;*
2. *$\mathcal{A}$ is the set of actions, or a set of triples $a = (\mathsf{precond(a)}, \mathsf{effects}^-(a), \mathsf{effect}^+(a))$, where sets $\mathsf{effect}^-(a)$ and $\mathsf{effect}^+(a)$ are called negative and positive effects of a (resp.),*
3. *$\gamma(s, a) = (s - \mathsf{effects}^-(a)) \cup \mathsf{effects}^+(a)$ if $a \in \mathcal{A}$ is applicable to $s \in \mathcal{S}$, otherwise γ is undefined; in fact, γ describes the state transformation after a being applied to s,*
4. *$\mathcal{S}$ is closed under γ, that is $\gamma(s, a) \in \mathcal{S}$ for each action a applicable to s*[9]*.*

[5]If a planning language is given, elements of both sets are rendered by literals of this language.

[6]In [34], names are omitted for a cost of two temporal factors: the action start and the action end. This approach – useful in approaches integrating planning with scheduling or a plan monitoring, seems to be inappropriate for a use of defining of actions because of a lack of a distinction between temporal planning and a classical planning.

[7]Since *Corr* is a location, $\neg Corr^B$ should be understood as a notation of the fact that B is not located in the *Corr*.

[8]In this role some other languages such as *Linear Temporal Language* – introduced by A. Pnuelli in [35] or languages of a modal, an attributive or a descriptive logic, see:[3]. Nevertheless, the mutual relationships between first-order logic and modal logic (and LTL, as well) is complicated. See: the first-order definability criteria of van Benthem.

[9]D. Gabbay in [2] considers a slightly different definition of a planning domain. Instead of γ (defined as above) he considers a transition function $\triangle : \mathcal{S} \times \mathcal{A} \mapsto 2^S$, which associates to each state $s \in S$ and to each $a \in \mathcal{A}$ a *set* $\triangle(s, a) \subseteq S$ of next states. Note that this approach forms some generalization of the presented approach as it introduces more nondeterminism (instead of a concrete state – a set *set* $\triangle(s, a)$ is known only.

A domain is **finite**, if both sets $\mathcal{S}$ and $\mathcal{A}$ are finite. This definition should be further refined if the planning language $\mathcal{L}$ is known up to details. In such a case, we usually consider states $s \in \mathcal{S}$ as a subset of propositions $p_1, p_2 \ldots \in \mathcal{L}$ and $\mathcal{S} \subseteq 2^L$.

Due to a long-term convention widely accepted in the domain of AI planning [3] – a classical planning is associated to the first order language. Nevertheless, some other languages – even the propositional ones – such as: a language of Linear Temporal Logic (LTL)[10] or a language of Descriptive Logic or Planning Domain Description Language (PDDL) and its extensions (PDDL+) [11] can play a role of planning languages. The problem of mutual relationships between these languages and a language of first-order logic forms an intriguing meta-logical issue, but it still is under investigation and the discussion seems far from being closed [12].

Example 2 *Consider a crane in the block world as depicted in Fig.2. A planning domain contains 3 states* s_0, s_1, s_2 *– built up from propositions of a given language* $\mathcal{L}$*; the details may be as follows*[13]*:*

$\mathcal{L} = \{$A-on-platform, B-on-platform, C-on-platform, A-on-B, C-on-A$\}$
$S = \{s_0, s_1, s_2\}$, where:

- $s_0 = \{$A-on-platform, B-on-platform, C-on-platform$\}$,
- $s_1 = \{$B-on-platform, C-on-platform, A-on-B$\}$,
- $s_2 = \{$B-on-platform, A-on-B, C-on-A$\}$.

We admit actions of the following type:

- *take(*x, platform) – take an object x from the platform, for $x \in \{A, B, C\}$.
- *put(x,y)* – put x on y.

Due to this – a set of all admissible actions is as follows:

$$\mathcal{A} = \{take(A,\ platform)\ take(B{,}platform),\ take(C,\ platform),\ put(A, B), put(A, C),\\ put(B, A), put(B, C),\ put(C, A), put(C, B)\}.$$

In practice, the following restricted set of actions is required to reach the final situation (state s_2) – as depicted in Fig.2. $\mathcal{A}_1 = \{take(A{,}platform),\ put(A, B), take(C{,}platform),\ put(C, A)\}$.

Plans and planning problems

A plan is a (finite) set of actions to be performed on a planning domain in some contexts in order to reach the required goal state. Generally, different conventions are used for a task of defining a plan. Some approaches to defining plans are based on a concept of observability with planning domains of the form $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \gamma \rangle$. In such a framework a plan is formally defined as follows:

[10] LTL was introduced to computer science by A. Pnuelli in [35].

[11] PDDL was introduced in [36] and developed by M. Fox's school in [37, 38, 39, 40].

[12] Some criteria of the first-order definability of a modal logic-based language – that also applies to LTL as a temporal-modal system – were formulated by R. Goldblatt in [41]. This author gave some example of first-order undefinable formulas of a modal logic. The relationships between first-order logic and systems of description logic are more clear – as many of them are defined as decidable fragment of the first-order logic (FOL) – see: [42]. Finally, a very important and sophisticated criterion of the first-order definability was formulated in a pioneering paper [43] of P. Lindstrøm from 1969. It was shown there that each language capable of expressing finiteness has an expressive power stronger than first-order logic. This fact should not be, however, mistaken with a fact that a first order logic can implicitly say about finite and infinite sets of objects, actions etc. The simplified proof of Lindstrøm Theorem may be found in Appendix 7.

[13] It has been said that states of planning domain are built up via identification of them with propositions of a given planning language. For simplicity of a depiction, a transition function presentation will be omitted.

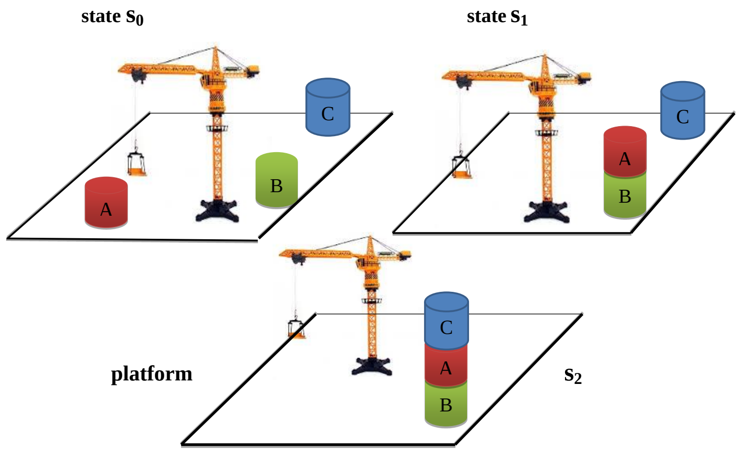


Figure 5: A crane in a block world

Definition 2 ***Plan.*** *([3], p. 21) Assume that a planning domain $\mathcal{D} = \langle S, \mathcal{A}, \gamma \rangle$ is given. A plan is any sequence of actions from $\mathcal{A}$:*

$$\pi = \langle a_1, a_2 \ldots, a_k \rangle \text{, where } k \geq 0. \tag{2}$$

Definition 3 ***Length of plan.*** *([3], p. 21) The length of the plan is $|\pi| = k$, the number of actions.*

Example 3 *Assume once again a planning situation for a crane in the block world as depicted in Figure 5. Its task is to construct a tower 'C-A-B' – as depicted in Figure 5. The plan for performing is given by the following sequence:*

$$\pi = \langle take(A, platform), put(A, B), take(C, platform), put(C, A) \rangle. \tag{3}$$

Obviously, the length of plan π equals 4.

If we enrich a planning domain by an initial planning state s_0 and by a planning goal g, we are already able to render a *classical planning problem.*

Definition 4 ***(Planning problem.)*** *A classical planning problem forms a triple $\mathcal{P} = \langle \mathcal{D}, s_0, g \rangle$, where:*

- *$\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \gamma \rangle$ is a planning domain, defined as earlier,*
- *$s_0 \in \mathcal{S}$ is the initial state,*
- *$g \in \mathcal{S}$ is to be the final planning state.*

If one exchanges a planning domain $\mathcal{D}$ for a set of planning operators, then one can define a *planning statement* as a tuple $\langle \mathcal{A}, s_o, g \rangle$. Since actions (planning operators) should be referred to planning goals, we need a notion of a *relevance* of an action a to a goal g in order to express such a reference. Informally speaking, a is relevant to g if a can produce a state which satisfies g (due to [3], p. 31). More formally:

Definition 5 *An action a is relevant to a goal g if:*[14]

[14]See: [3], p. 31)

- *it makes at least one of g's propositions true:*
  $g \cap \mathsf{effects}^{+}(\mathsf{a}) \neq \emptyset$ *(positive effects of a do not conflict with g).*

- *it does not make any g's proposition false:*
  $g \cap \mathsf{effects}^{-}(\mathsf{a}) = \emptyset$

One can also express this relevance by means of an inverse operator γ^{-1} as follows:

Definition 6 *If a is relevant to g, then:*

$$\gamma^{-1}(g, a) = (g - \mathsf{effects}(\mathsf{a})) \cup \mathsf{precond}(\mathsf{a})^{15}\,. \tag{4}$$

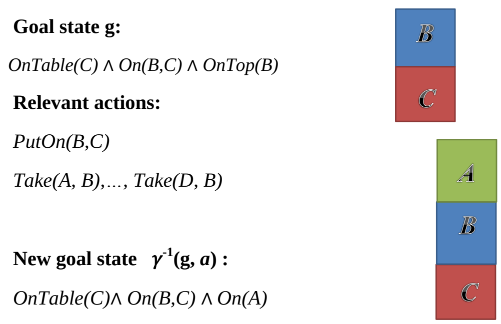


Figure 6: Goal g, applicable actions and a new goal obtained via inverse $\gamma^{-1}(g, a)$ for blocks A,B,C.

Example 4 *Consider a situation of the same robot in the block world, where blocks A,B,C form a tower A-B-C – as depicted in Figure 6. The simple robot task is to take the block A from the tower top. The goal g, new goal $\gamma^{-1}(g, a)$ and a possible set of relevant actions are depicted in Figure 6.*

Since $\mathsf{effect}^{+}\Big(PutOn(B, C)\Big) = On(B, C)$, so *PutOn(B,C)* is *relevant* to $g = OnTable(C) \wedge On(B, C) \wedge OnTop(B)$ as:

$$\Big(OnTable(C) \wedge On(B, C) \wedge OnTop(B)\Big) \cap On(B, C) \neq \emptyset. \tag{5}$$

Simultaneously,

$$\gamma^{-1}(g, take(A, B)) = (g - \mathsf{effects}\Big(\mathsf{take(A, B)}\Big)) \cup \mathsf{precond}\Big(\mathsf{take(A, B)}\Big) = \tag{6}$$

$$= \Big(OnTable(C) \wedge On(B, C) \wedge OnTop(B) - OnTop(B)\Big) \cup \Big(OnTop(A) \wedge On(A, B)\Big). \tag{7}$$

$$= \Big(OnTable(C) \wedge On(B, C) \wedge OnTop(A) \wedge On(A, B)\Big) \tag{8}$$

– as depicted in Figure 6.

[15] This definition in terms of the invese operator γ^{-1} may be, somehow, troublesome because of an ambiguity of its results. In fact, γ^{-1} may give two different states. Unambiguity holds only if γ is a bijection. Unfortunately, this difficulty was not discussed in [3]. Fortunately, it does not play any important role in the current analysis.

0.9 Planning as Searching in Graphs

Independently of the set-theoretic representation – planning may be considered as a graph-search procedure. Each graph is comprehended as an algebraic structure $\langle V, E\rangle$, where $V \neq \emptyset$ is a set of states called nodes and $E \subseteq V \times V$ is a binary relation between nodes. The elements of E are called *edges* or *arcs*. In planning, nodes represent states and edges – actions (operations) between them. Generally speaking, the planning graph-based approach constitutes a powerful model and tool for planning and scheduling in numerical domains. The main difference between this approach and the set-theoretic classical planning is that here the focus is on (physical) states and state-transitions. A consistent set of such transitions forms a path in an appropriate graph.

Independently of a fact that this planning paradigm forms a promising and illustrative (but not easy) approach to planning, we put aside its detailed description for a cost of a presentation of older planning methods of *forward search* and *backward search*. It follows from two reasons.

1 At first, these two methods – although naturally associated to this planning paradigm – are relatively independent of it because of their universal character.

2 Secondly, no special graph-based planning methods will be exploited in contributions – although the concept of graph itself will be exploited in representation of Temporal Traveling Salesman Problem.

0.9.1 Planning as Forward Search

One of the simplest and historically primary planning methodology in the search-based paradigm is the forward search, introduced in [44]. The idea of the forward-search is deceptively simple. Namely, for a given planning problem $(\mathcal{D}, s_0, g)$ a plan π is constructed iteratively as follows. For s_0 (different from g) we obtain an empty plan.

1 If a state s is identical to g, then we return some (non-empty) plan π.

2 Otherwise we consider a set called *applicable* of actions that their preconditions are true in a (currently considered) state s.

   A If this set is empty, the plan cannot be found in this path.

   B Otherwise, we can nondeterministically choose an action a from *applicable* and add this action to the end of the earlier plan π. In this way we obtain a new plan $\pi.a$.

In algorithmic form, these operations may be rendered as follows:

applicable $\leftarrow \{a|$ a is a ground instance of an operator in $\mathcal{A}$,
    and `precond(a)` is satisfied in $s\}$
 if $non - applicable$ **then return** `failure`
 choose nondeterministically any action $a \in$ *applicable*
$s \leftarrow \gamma(s, a)$
$\pi \leftarrow \pi.a$

Simultaneously, after reaching the new state $\gamma(a, s)$, the whole procedure is repeated. We stop it if such a state s_k that $\gamma(s_k, a_k) = g$ is achieved. In this moment, we take the sequence $\pi_{fin} = \langle a_1, a_2, \ldots, a_k\rangle$ as a required plan. These ideas are presented in a compact form in the following `Forward-search` algorithm (see: [3], p. 70).

```
Forward-search(𝒜, s, goals)
s ← s0
π ← the empty plan
    loop
    if s satisfies g then return π
    applicable← {a| a is a ground instance of an operator in 𝒜 precond(a)⊆ s0}
            and precond(a) is satisfied in s}
     if non − applicable then return failure
     choose nondeterministically any action a ∈ applicable
    s ← γ(s, a)
    π ← π.a
```

The main difficulty with this forward-search algorithm is a problem how to improve efficiency reducing the search space (even for a cost of losing of the algorithm completeness). The STRIPS-method (abbreviation from: Stanford Research Institute Problem Solver) was chronologically one of the first attempt to overcome this difficulty [16].

Although STRIPS is not necessary associated to the forward-search (it may be based on a backward-search, as well), it is convenient to describe it in the context of the forward-search in a version based on it. However, one needs to underline that STRIPS should be seen as a technique or a rule applied to the state descriptions to produce new state descriptions. It informs how states changes in planning and which preconditions must be satisfied to execute actions (see: [44], pp. 277-280, 298-99.).

STRIPS as based on forward search procedure (as the so-called *F-rule*) works according to the following rules (see:[44, 3])– beginning from the initial state s_0 to a goal g.

1. This algorithm works if a set of *states* is not empty,
2. Then we choose a state $s \in$*states*.
    - If $g \subseteq s$, then we take $\pi(s)$ (a plan in a state s) as a desired plan.
    - Otherwise, we take a set $E(s)$ of actions applicable to s.
        (a) if $E(s)$ is empty – we remove s from *states*,
        (b) if does not – we choose an action $a \in E(s)$ and exchange s for $s^{'}$ by removing effects of a from s. Then a current plan $\pi(s^{'}) = \pi(s).a$ (The action a is added at the end.)
        (c) The same procedure is repeated for a set $E(s^{'})$, etc. until g will be achieved.

[16] STRIPS reduces the size of search space with respect to the classical `forward-search` algorithm as it refers added actions to the planning goal.

Elements of this procedure may be represented in the STRIPS-algorithm as follows.

`STRIPS-algorithm`$(\mathcal{A}, s_0,$ g)
begin
`state`$= \{s_0\}$
$\pi(s_0) = \langle\rangle$
$E(s_0) = \{a|$ a is a ground instance of an operator in $\mathcal{A}$ and `precond(a)`$\subseteq s_0\}$
while true **do**
if `states` $= \emptyset$ **then return** `failure` **end if**
choose a state $s \in$ `state`
if $g \subseteq s$ **then return** $\pi(s)$ **end if**
if $E(s) = \emptyset$ **then**
remove s from *states*
else
choose and **remove** an action $a \in E(s)$
$s^{'} \leftarrow$ s/ `Effect`$^{-}(a) \cup$`Effect`$^{+}(a)$
if$s^{'} \notin$ *states* **then**
$\pi(s^{'}) = \pi(s).a$
$E(s^{'}) = \{a|a$ is a ground instance of an operation $\in \mathcal{A}$ and `precond(a)`$\subseteq s^{'}\}$
end if
end if
end

The following example illustrates how this method works in a simple situation of acting in the block world.

Example 5 *Consider a simple situation of acting in the block world with two states: the initial one s_0 and a goal state g defined as depicted in Figure 7*[17].

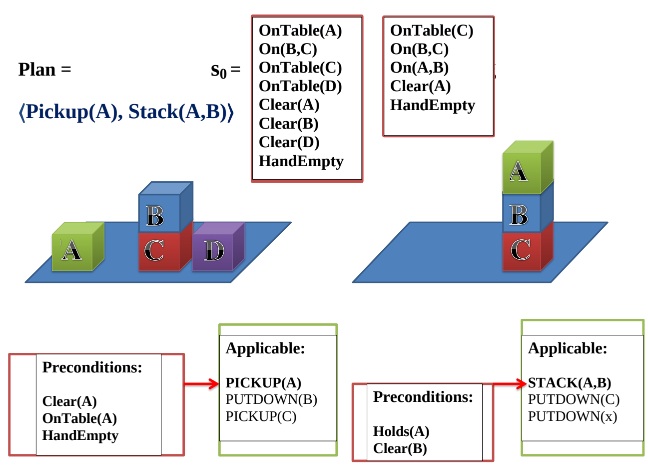


Figure 7: STRIPS-method for a simple situation of acting in a block world. Detected preconditions are only applicable to actions: PICKUP(A) and STACK(A, B), so – due to STRIPS-algorithm – these actions determine the required plan.

Unfortunately, the `forward-search` algorithm may be insufficient to achieve a desired plan and to rich a goal. In fact, it may require a complementation by a back-tracking if it is clear that a chosen path

[17]This example forms a slight modification of example found under: www.cs.bham.ac.uk/rwd./Planning/slides-3.pdf

does not lead to a required goal. Thus, planning sometimes forms a combination of forward-search with backward-search.

0.9.2 Planning as Backward Search

An alternative search-based planning method is a backward search — see: [3]. The main difficulties between forward search and backward search may be put forward as follows.

Forward search:

- we start from the initial state s_0,
- we always move from a state s to the new state $\gamma(s, a)$.

Backward search:

- we start from the goal g,
- we move from a set of states S to a new set of states $\gamma^{-1}(S, a)$.

An idea of backward search is the following one: we start at the goal and apply inverses of the planning operators to determine subgoals. We stop this procedure if a set of subgoals – satisfied by the initial state – is produced. This constitutes a 'core' of the following `Backward search algorithm` (see:[3], p. 73.).

> `Backward-search`$(\mathcal{A}, s_o, g)$
> **begin**
> $\quad\pi \leftarrow$ the empty plan
> $\quad$loop
> $\quad$**if** s_o satisfies g **then return** π
> $\quad$*relevant* $\leftarrow \{a|a$ is an action from $\mathcal{A}$ that is relevant for $g\}$
> $\qquad$**if** relevant $= \emptyset$ **then return** `failure`
> $\qquad$**choose** nondeterministically an action $a \in relevant$
> $\qquad\pi \leftarrow a.\pi$
> $\qquad g \leftarrow \gamma^{-1}(g, a)$.
> **end**

Backward Search: Explanatory Note. In order to better grasp how this algorithm works, assume that a set $\mathcal{A}$ of planning operators (actions)[18], an initial state s_0 and a goal g are given.

One starts with the empty plan. Then, in the next steps, a required plan is searched in the following way.

1 If s_o satisfies the *goal* g than we take s_o and return as a new plan π.

2 Otherwise, we consider a set called 'relevant' of actions that are relevant to the goal g.

- **A** If no action is relevant to g, then the algorithm returns `failure` as it is impossible to generate a plan leading to achieving the goal.
- **B** Otherwise, one nondeterministically chooses an action a (from a new set of them – called 'relevant') and exchanges π for $a.\pi$, i.e a is added(from the beginning) to the initial plan π. Simultaneously, we define a new subgoal by the inverse planning operator $\gamma^{-1}(g, a)$ to choose next actions that satisfies such a subgoal.

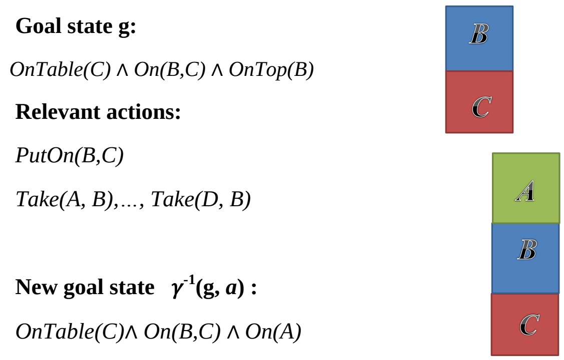


Figure 8: Goal g, applicable actions and a new goal obtained via inverse $\gamma^{-1}(g, a)$ for blocks A, B, C.

This procedure is repeated up to the initial state. Definition 6 and Example 4 (explaining how the inverse operator $\gamma^{-1}(g, a)$ works) allow us to illustrate the backward-search algorithm as well.

As it has already been signalized – forward-search and backward-search methods (with STRIPS as a distinguished one) are widely exploit in *the graph-based planning*. This procedure – similar to iterative deepening – is based on two types of activity:

- exploring a new (graph) search area from the initial one, and
- backtracking from a level P_i for some i[19] including all goal propositions.

The backward search with respect to graph forms some unique specification of the general backward search method and it works as follows. We begin with a level P_i which includes the required goal g, i.e $g \in P_i$, looking for a sequence π of actions – leading to g – which are mutually independent (non-mutex). During this backtrack, we define new subgoals of g: g_1, g_2, g_k, for some k, which allow us to find earlier actions leading to them. If finally level 0 is successfully reached, the corresponding sequence Π of actions forms a desired solution plan.

Since graph-planning-based approach only plays a role of a support in further considerations, a description of more sophisticated methods of this paradigm with examples has been moved to Appendix.

0.10 Planning as Satisfiability

An alternative approach to planning representation relies on satisfiability of formulas expressing different planning components. In fact, it has been mentioned that each planning domain is associated to a planning language. Although logical formalisms, such as McCarty's *Situations Calculus* (see: [45]) , designed for dynamical domains with actions, fluents and situations were elaborated in 60's, the satisfiability-based approach to planning was explicitly formulated relatively late in 1992 in [24]. An importance and a role of planning languages in planning is reflected by the following two facts:

[18] For simplicity, we identify actions with planning operations. Authors of [3] identify actions with ground instances of planning operations.

[19] Each of P_i is a part of a considered graph G admissible at the iterative step i.

Fact1 states $s_i \in \mathcal{S}$ in a planning domain may be naturally viewed as propositions of a planning language $\mathcal{L}$ and $\mathcal{S} \subseteq 2^{\mathcal{L}}$,

Fact2 planning conditions, action preconditions, etc. may be identified with the appropriate formulas representing them in a given planning language $\mathcal{L}$.

This linguistic depiction of planning introduces a new methodological quality and it allows us to built a bridge between planning and computational logic and to incorporate some logical procedures and operations on logical formulas for planning procedures.

One of such procedures are: the so-called *Unit-propagation* and *Davis-Putnam Procedure* – based on unit-propagation – introduced in a purely logical contexts already in 1960 year in [46] by H. Putnam (1926-2016) and M. Davis (1928-), thus many years before an explicit defining of planning as satisfiability of formulas. Both procedures will be described now in details.

0.10.1 Unit-propagation and Davis-Putnam Procedure

Davis-Putnam makes use of the so-called *unit-propagation* – as it has just been mentioned. The main role of this smart subprocedure relies on simplifying the formulas given in a Conjunctive Normal Form (CNF). This procedure may be specified as follows.

Unit-propagation. Assume that a formula Φ of a planning language $\mathcal{L}$ is given in a CNF, that is $\Phi = C_1 \wedge C_2 \wedge \ldots C_k$ for some k (each $C_i \in \Phi$ is called a *clause*). On the input of the `Unit-Propagation` we have Φ and some empty model μ. In output we get a simplified formula Φ and a newly extended model μ. This model is just extended for a cost of the simplified Φ. Namely:

1 a unit clause $\{l\}$ is chosen in Φ (if there exists) ,

2 all clauses $C_i \in \Phi$, where l occurs, are thrown away ,

3 $\neg l$, where $\neg l$ occurs, thrown away from clauses C_i ,

4 the rejected unit clause is added to μ-model.

Thus, we exchange a formula Φ by $\Phi - C$ (for a unit clause $\{l\}$) and by $\Phi - C \cup \{C - \{\neg l\}\}$ for $\{\neg l\}$. Observe, that we reject, in the first case, the whole clause C from Φ. In the second one, we preserve it rejecting a unit clause $\{\neg l\}$ only. This procedure is depicted in a compact form by the following `Unit-Propagate` algorithm.

`Unit-propagate`(Φ, μ)
begin
    while there is a unit clause $\{l\}$ in Φ **do**
        $\mu \leftarrow \mu \cup \{l\}$
        for every clause $C \in \Phi$
            if $l \in C$ then $\Phi \leftarrow \Phi - \{C\}$
            else if $\neg l \in C$ then $\Phi \leftarrow \Phi - C \cup \{C - \{\neg l\}\}$
end

In such a framework, Davis-Putnam procedure may be specified as based on unit-propagation, which is applied to all cases (formulas) excluding two aberrations: when $\emptyset \in \Phi$ and $\phi = \emptyset$.

If unit-propagation is applied to Φ, then algorithm orders to select a variable P such that P or its negation $\neg P$ occurs in Φ and to reject it from Φ. Finally, the algorithm orders to continue the same procedure for the simplified formula, say Φ_1, and for its corresponding (newly extended) models $\mu \cup \{P\}$ or $\mu \cup \{\neg P\}$ (resp.)
In terms of algorithm:

select a variable P such that P or $\neg P$ occurs in Φ

`Davis-Putnam` $(\Phi - \{P\}, \mu \cup \{\neg P\})$
`Davis-Putnam` $(\Phi - \{\neg P\}, \mu \cup \{P\})$

Due to [3] – the whole Davis-Putnam procedure may be algorithmically depicted as follows.

```
Davis-Putnam(Φ, μ)
begin
   if ∅ ∈ Φ then return
   if Φ = ∅ then exit with μ
   otherwise Unit-propagate
   select a variable P such that P or ¬P occurs in Φ
   Davis-Putnam (Φ − {P}, μ ∪ {¬P})
   Davis-Putnam (Φ − {¬P}, μ ∪ {P})
end

Unit-propagate(Φ, μ)
begin
    while there is a unit clause {l} in Φ do
          μ ← μ ∪ {l}
          for every clause C ∈ Φ
                  if l ∈ C then Φ ← {Φ − {C}}
                  else if ¬l ∈ C then Φ ← {Φ − C} ∪ {C − {¬l}}
end
```

Example 6 *([3] pp. 153-154) Consider a formula* $\Phi = D \wedge (\neg D \vee A \vee \neg B) \wedge (\neg D \vee \neg A \vee B) \wedge (\neg D \vee \neg A \vee B)$ *in CNF. Construct a model for this formula via Davis-Putnam procedure.*

Solution: The solution via this method is illustrated in Figure 9. The initial unit propagation allows us to simplify the initial Φ to Φ_1 by rejecting (all occurrences of) D, taking now $\mu = \{D\}$. Suppose that the variable selection rule indicates A. Then P-D calls unit propagation, which eliminates A. As a result, a simplified formula $\Phi = \neg B \wedge B$ is obtained with a new model μ extended by A, i.e. $\mu = \{A, D\}$ now. Nevertheless, the formula Φ generates a contradiction, so no model can exists in this case. The Davis-Putnam procedure backtracks with a new choice $\neg A$. The unit-propagation simplifies Φ once again to $\neg B$ and the final model $\mu = \{D, \neg A, \neg B\}$ is achieved.

Example 7 *Consider a very basic planning situation described by a formula describing a possible situation of an agent's activity. The reservoir of the agent actions is the following :* $Move(P)$ *(the agent can move from a point P),* $Put(A)$ *(the agent can put a block A somewhere),* $Load(B)$ *(the agent can load B somewhere). Find a possible consistent plan of agent's activity.*

Solution: The solution via the same method is illustrated in Figure 10.

0.11 Planning with Temporal Operators

In the earlier chapter, planning domains, planning problems and planning statements were specified. In this chapter, a temporal extension of these definitions will be proposed. Namely, we introduce notions of:

- temporal planning domain,
- temporal planning problem and,

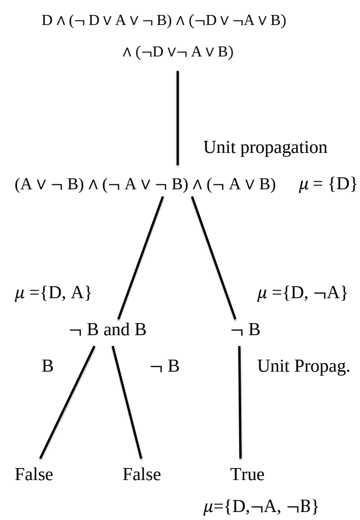


Figure 9: Davis-Putnam procedure for Example 66

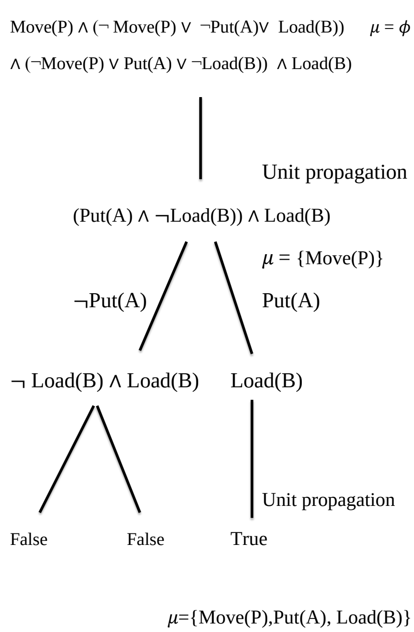


Figure 10: Davis-Putnam procedure for Example 7

- temporal planning statement.

Generally speaking, all of these notions are defined similarly as in a classical planning case. The only difference consists in a fact that (almost) each component of temporal planning domains and problem is *internally equipped* by some *temporal constraints*. Putting aside a formal definition of them (to be presented in details in the next part), we understand them as all possible constraints or restrictions involved in a *time*. They can take a form of time intervals, temporal relations or a point-wise restriction imposed on an action execution.

Some differences between classical planning and temporal planning may be easily observed with respect to the planning domain definition. It has been said that a planning domain $\mathcal{D}$ can be defined in a classical planning case as a tuple:

$$\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \Gamma \rangle \tag{9}$$

where $\mathcal{S}$ forms a set of states, $\mathcal{A}$ is a set of actions and Γ is a set of transition functions between states from $\mathcal{S}$. In temporal planning we simply enrich all of these components by some temporal elements. More precisely, we exchange:

a) a set of states $\mathcal{S}$ for a temporal database $\mathcal{F} = \langle F, C \rangle$, where F constrains states with associated temporal intervals and C is a finite set of temporal and object constraints[20],

b) a set of *actions* $\mathcal{A}$ for a set O of *temporal planning operators*. Each operation $o \in O$ can be seen as an action $a \in \mathcal{A}$ with associated temporal or object constraints imposed on execution of a,

c) a set of *transitions* Γ for a set $\mathcal{X}$ of *domain axioms* defining a 'behavior' of actions, conditions of the plan consistency, restrictions imposed on objects, etc. [21]

This leads to the following definition.

Definition 7 ***(Temporal planning domain.*** *[3], p.320) A temporal planning domain is a tuple $\mathcal{D} = \langle \mathcal{F}, O, \mathcal{X} \rangle$, where:*

a) $\mathcal{F} = \langle F, C \rangle$ *is a temporal database and F – a set of states with associated temporal intervals, C is a finite set of consistent temporal and object constraints,*

b) $O = \{o \mid o = (\text{name(o)}, \text{precond(o)}, \text{effects(o)}, \text{const(o)})^{22}\}$ *is a set of temporal planning operators,*

c) $\mathcal{X}$ *is a set of domain axioms.*

Additionally, two other components of classical planning should be also generalized: an initial state s_0 and a planning goal g. Namely, we also exchange:

d) an initial state s_o for a set – *initial scenario* $\Phi_o \subseteq \mathcal{F}$ that represents the initial scenario with some object- and temporal constrains imposed on initial states and actions,

e) a planning goal for the *final scenario* $\Phi_g = (F_g, C_g)$, where F_g contains goals[23] and C_g is a set of object- and temporal constrains imposed on them.

This leads to the following short definitions of the temporal planning problem and the temporal planning statement.

[20] C is required to be consistent in CSP-sense, that is there exist values for variables that meet all the constraints from C.

[21] Therefore, one can see a temporal planning domain more as semantics models, in which a connection between states and objects and a language for their description is given. A classical planning domains are rather structures that 'exist' outside a language.

[22] const(o) denotes object or temporal constraints imposed on an operation o.

[23] We admit more than one goal of the temporal planning problem.

Definition 8 *(**Temporal planning problem.**) A temporal planning problem is a tuple* $\mathcal{P} = \langle \mathcal{D}, \Phi_0, \Phi_g \rangle$, *where* $\mathcal{D}, \Phi_0, \Phi_g$ *are specified as above.*

Definition 9 *(**Temporal planning statement.**[3], p.320) A temporal planning statement is given by the tuple* $\mathcal{P} = \langle O, \mathcal{X}, \Phi_0, \Phi_g \rangle$, *where* $\mathcal{D}, \Phi_0, \Phi_g$.

Example 8 *Consider two actions performed by a robot R:* `move a block A from a room` R_1 *that should be performed in a time interval* $[t_1, t_2]$ *and an action* `take a block A from` R_1 `to` R_2 *in a time interval [20, 50] (of time units). These activities can be encoded by the so-called 'temporally qualified expressions' describing a temporal database as follows:* `move(A,` R_1`)@`$[t_1, t_2]$ *and* `take(A,` R_1, R_2`)@[20,50]`.

Example 9 *The following formulas of a predicate calculus form a possible exemplification of domain axioms:*

- *An object r cannot be in two distinct places l at the same time:*
  $at(r, l)@[t_s, t_e], at(r', l')@[t'_s, t'_e] \rightarrow (r \neq r') \vee (l \neq l') \vee (t_e \leq t'_s) \vee (t'_e \leq t_s)$,
- *(consistency condition) There is no such a time interval, where both actions* a_i*and*$\neg a_i$ *are performed:*
  $\neg \exists i, j \in I(\bigcup_{i,j \in I}(a_{ij}@[t_i, t_j] \wedge (\neg a_{ij}@[t_i, t_j])$.

Example 10 *Consider a crane* ***Crane*** *in the 3-block worlds (the block are marked:* `A, B, C`*), performing a task of their dislocation. In some initial interval* $[t_1, t_2]$ *all blocks are put on a platform* `Platform` *and a* `Crane` *is free. The initial scenario* Φ_0 *of a temporal planning problem can be specified as follows.*

$\Phi_0 = \{$`at(platform, A)@`$[t_1, t_2]$, `at(platform, B)@`$[t_1, t_2]$, `at(platform, C)@`$[t_1, t_2]$, `free(Crane)@`$[t_1, t_2]\}$

0.12 Planning Languages.

Different planning languages are used for a representation of planning problems and their specification. These languages have at least two common denominators: they are rather propositional, but strongly specified in order to describe different types of objects and activities. Therefore, almost all planning languages should be labeled as *descriptive* languages [24].

In the framework of this work considerations, we restrict a taxonomy of these languages to[25]:

1. *Linear Temporal Logic* (LTL),
2. *Planning Domain Description Language* (PDDL) and
3. *Halpern-Shoham Logic* (HS).

LTL and PDDL will be discussed in this chapter. HS will be described in the next chapter in the context of Allen's algebra of interval relations – as a formalism suitable to describe this algebra in logical terms.

0.12.1 Linear Temporal Logic (LTL)

.

Linear Temporal Logic was invented by Amir Pnueli in [35] in 1977 for the formal programming verifications. LTL may be briefly specified as a modal temporal system referring to time as having a linear nature. LTL may be naturally extended to the so-called *Computational Tree Logic* – as a branching-time logic with a future which is not determined. The syntax and semantics of LTL may be specified in detail as follows.

[24] From another programming perspective, almost all planning languages are declarative languages (PROLOG, ASP).
[25] For example, we avoid syntactic details of *Action Description Language* as we do not intend to make use it later.

Syntax. A bi-modal LTL-language is obtained from standard propositional language (with the Boolean constant $\top$) by adding temporal-modal operators such as: *always in a past* (H), *always in a future* (G), *eventually in the past* (P), *eventually in the future* (F), *next and until* ($\mathcal{U}$) and *since* ($\mathcal{S}$) – co-definable with "until". The set FOR of LTL-formulas is given as follows:

$$\phi := \phi|\neg\phi\,|\phi \vee \psi\,|\phi\mathcal{U}\psi\,|\phi\mathcal{S}\psi\,|H\phi|P\phi\,|F\phi\,|Next(\phi) \tag{10}$$

Some of the above operators of temporal-modal types are together co-definable as follows: $F\phi = \top\mathcal{U}$, $P\phi = \top\mathcal{S}\phi$ and classically: $F\phi = \neg G\phi$ and $P\phi = \neg H\phi$.

Interpretation. LTL is usually interpreted in the point-wise time-flow frames $\mathcal{F}\nabla = \langle T, < \rangle$ and dependently on a set of states S. Thus, the satisfaction relation is defined for pairs (t, s), where t represents a time point and s – a state. Nevertheless – it is sometimes defined for pairs (t, f), where a function $f : T \mapsto S$ associates $t \in T$ to some state $s \in S$. In detail:

1. $(t, f) \models G\phi \iff (\forall t^{'} > t)t^{'} \models \phi$, $(t, f) \models H\phi \iff (\forall t < t^{'})t^{'} \models \phi$.
2. $(t, f) \models F\phi \iff (\exists t^{'} > t)t^{'} \models \phi$, $(t, f) \models P\phi \iff (\exists t < t^{'})t^{'} \models \phi$.
3. $(t_1, f) \models \phi\mathcal{S}\psi \iff$ there is $t_2 < t_1$ such that $t_2, f \models \psi$ and $t, f \models \phi$ for all $t \in (t_1, t_2)$
4. $(t_1, f) \models \phi\mathcal{U}\psi \iff$ there is $t_2 > t_1$ such that $t_2, f \models \psi$ and $t, f \models \phi$ for all $t \in (t_1, t_2)$
5. $(t_k, f) \models Next(\phi) \iff (t_{k+1}, f) \models \phi, k \in \mathcal{N}$.

0.12.2 PDDL and PDDL+

Planning Domain Description Language (PDDL) – invented by McDermott in [36] and developed by Fox's school in [37, 38, 39] – was hailed as being a benchmark language for representation of classical and temporal planning. One can even observe a kind of a cult around it and its extensions – collectively denoted as PDDL+. We give now a short chronology of different versions of PDDL+ beginning with the initial one – commonly known as PDDL1.2.

PDDL1.2 (1998). It based on a distinction between *domain description* and the related *planning description*. The domain description usually consists of a *domain-name definition*, *definition of requirements*, a definition of *object-type hierarchy* and predicates. PDDL1.2 also admits components connected with actions such as: *action parameters, preconditions, effects and action goals*. These action components ensure a correlation with a specification of actions in STRIPS-method.

PDDL2.1 (2002). This version extends PDDL.1.2 towards the *numerically measured fluents* such as: time, distances, weight, energy, etc.. PDDL2.1 is also sensitive for a phenomenon of *continuity*, for example, with respect to *continuous actions*. They are considered in a perspective of different metrics associated to PDDL2.1. All these facts enable of considering this version as a temporally extended PDDL1.2.

PDDL2.2 (2004). This version forms a (slight) further temporal extension of PDDL2.1. by some *timed literals* in order to model exogenous events that occur independently from the plan execution. It also forms a germ of the preferential version of PDDL since it allows to consider mutual relations between events, facts and states (e.g. richeability). In addition, one can impose some features on these relations such as: transitivity, reflexivity, etc. in PDDL2.2. These mutual relations are introduced to PDDL2.2 by *derived predicated*.

PDDL3.0 (2006). This version may be viewed as a full *preferential extension* of PDDL2.1 – as this extension allows us to represent preferences imposed on actions and plan execution. Preferences are considered here as soft constraints, which must not necessary be satisfied. This version allows us to deal with hard-constraints (which should be true for each state of a plan execution), as well.

Independently of a still increasing number of PDDL-extensions, it seems that both PDDL and PDDL+ are burdened by some difficulties.

1. At first, its seems that there is no clear criterion of distinguishing the well-formed PDDL-expressions from the non well-formed expressions.

2. Secondly, there is no clear criterion of distinction between PDDL (PDDL+) and its metasystems. In fact, there is no such a meta-system at our disposal.

3. The next, PDDL (PDDL+) seems to be a half-formal language only – since it is essentially based on natural language expressions. In consequence, PDDL (PDDL+) may share the same antynomies of natural languages.

4. In addition, semantics for PDDL (PDDL+) is very restrictive – as it seems to only refer to the intended models. In this way, this semantics ignores a possible effect of (an existence of) unintended models.

5. Finally, neither PDDL, nor PDDL+ is associated to a solver or planner – by contrast to some formal languages as PROLOG or ASP.

All these facts make this language only partially attractive from the perspective of current considerations. Hence, it will not be exploited very broadly.

0.13 State of the Art and History of Research in this Area

Early works on planning (especially on automated planning) were essentially influenced by works on automated theorem proving, so planning was reflected in them as a theorem proving problem – due to the Green's work [47] from 1969. In this work an axiomatic description of initial states, goals and planning operators were introduced. In the same year, J. McCarty discussed a problem of changeable and unchangeable planning components in terms of the so-called *frame problem* in [45], what allowed to observe that classical planning problem provides a simple solution of the *frame problem*.

Majority of further research on planning referred to planning methodology such as: Nilsson's works [44, 48] – describing a STRIPS-method, partially still in terms of theorem proving. STRIPS was also discussed in [2]. The next planning method – the Davis-Putnam procedure was primary invented by H. Putnam and M. Davis in 1960 in [46] as a purely logical procedure for satisfiability checking for propositional calculus. In 1962, Logemann and Loveland improved in [49] the propositional satisfiability step of the DavisPutnam procedure such that is only requires a linearly input-dependent amount of memory in the worst case. This improved procedure still remains a basis for many SAT-solvers, was recently discussed in [50] and in a seminal book o Harrison [25].

The graph-plan paradigm of planning is relatively new and its was initiated by A. Blum, M. Furst and W. Langford in a series of their works: [4, 5, 6]. Simultaneously, some CSP-techniques for the graph-plan analysis were discussed in [7, 8, 9]. Currently, CSP-techniques are used in much more specified contexts of planning, for example, in mission planning for Unmanned Air Vehicles (UAV), as in [51, 52, 53, 54].

A need of a planning domain description formed a main motivation factor to introduce the *Planning Domain Description Language* (commonly known as PDDL). PDDL –introduced in [36] – was intensively developed by Fox and Long's school and it currently has several extensions (denoted together by PDDL+) suitable for describing *temporal* planning domains – introduced in [37, 38, 39, 40]. Although PDDL as a flexible descriptive language is suitable to be extended and it is commonly considered as a standard language, it cannot constitute a real concurrence for such programming languages as PROLOG or ASP – equipped with effective planning solvers. Moreover, both PDDL and PDDL+ seem to share some difficulties of a natural language such as a lack of a distinction between a language and its meta-language, etc.

The presentation of action in this chapter – adopted from [3] is partially based on a formalism for action representation – called *the Action Description Language* – introduced and developed by Pednault in [21, 22, 23] and naturally interpreted in systems with transitions. This last semantics type will be incorporated in further chapter of the thesis.

Part II

Introduction 2: Temporal Constraints, Preferences and Fuzzy Temporal Constraints

Introduction 2: Temporal Constraints, Preferences and Fuzzy Temporal Constraints

0.14 Temporal Constraints – a Brief Introduction

Last chapter emphasizes a difference between classical planning and temporal planning by means of a notion of temporal planning operators – typical only for temporal planning. It emerges that a difference between these two types of planning is even better graspable in the perspective of the so-called temporal constrains, i.e. constraints (restrictions) of a temporal nature – imposed on planning actions, events or objects. Generally and informally speaking, temporal constraints may be referred to the following temporal planning components [34, 3]:

Comp1 action duration,

Comp2 time of action performing,

Comp3 temporal relations between actions or events[26].

In the perspective – determined by a notion of temporal constraints – a difference between classical and temporal planning may be viewed as follows. Classical planning refers to time (at most) *implicitly*, for example by a sequencing of time-points[27], but *not explicitly*, without a reference to an action duration and without temporal relations between actions and events[28].

Since the meaning and a role of temporal constraints is intuitively understandable and their presentation seems to be the most suggestive in terms of their taxonomy, their exact definitions will be proposed later in terms of the so-called *Simple Temporal Problem* (STP) and its extensions.

At this moment, we only announce that temporal constraints are usually divided into two classes – see, for example, [3]:

- the *qualitative* temporal constraints and
- the *quantitative* ones.

This traditional distinction will be adopted in this chapter as the main taxonomy criterion for temporal constrains.

0.15 Qualitative Temporal Constraints

The nature of qualitative constraints relies on a fact that they 'measure' a temporal *quantity* between entities of different sorts such as: points, objects, intervals, etc. and – in pure form – they are not capable of measuring any quantitative relations between them. The main criterion of a taxonomy of qualitative temporal constraints is a nature of entities that they refer to. Since temporal constraints can be just represented (or even identified) with appropriate temporal relations between these entities, they are suitable to be represented by some algebraic calculus. Classically, the following two base-types of calculi for quantitative temporal constraints are distinguished:

PA point algebra and

IA interval algebra of Allen's temporal relations.

[26]Of course, this short taxonomy does not have any exhaustive character, it only plays an illustrative role.

[27]This sequencing of time points usually refers to an order of actions or states.

[28]Authors of [3] (pp.285-86) explain a difference between classical and temporal planning by a statement that classical planning only answer the questions: *what* changes or *what* conditions are required to make planning procedures inconsistent or reasonable, temporal planning takes also into account the question: *when* something can or cannot occur. This statement, seems to form a kind of a conceptual idealization in the light of the above observation.

0.15.1 Point Algebra

The point algebra (PA) – also called point calculus [55] – is based on reasoning about relations between time points – usually interpreted as rational numbers and denoted by $t_1, t_2, \ldots$ etc. These points form a relation algebra provided that we introduce 3 basic (and pairwise disjoint) relations between time points $(<, >, =)$, all possible unions of them $(\leq, \geq, =)$ and $\top$ as the universal relation and the empty relation $\perp$)[29].

PA as an algebra is closed under *union* of relations ($\cup$), *intersection* ($\cap$), *inverse* ($^{-1}$) and *composition* ($\bullet$). Assuming that R_1, R_2 are some basic relations between time points, one can define these new relations as follows:

$$R_1 \cup R_2 = \{\langle t_1, t_2\rangle : \langle t_1, t_2\rangle \in R_1 \vee \langle t_1, t_2\rangle \in R_2\} \tag{11}$$

$$R_1 \cap R_2 = \{\langle t_1, t_2\rangle : \langle t_1, t_2\rangle \in R_1 \wedge \langle t_1, t_2\rangle \in R_2\} \tag{12}$$

$$R^{-1} = \{\langle t_1, t_2\rangle : \langle t_2, t_1\rangle \in R\} \tag{13}$$

Finally we define the composition of relations R_1 and R_2 in PA as follows:

$$R_1 \bullet R_2 = \{\langle t_1, t_3\rangle | \exists t_2 : \langle t_1, t_2\rangle \in R_1, \langle t_2, t_3\rangle \in R_2\} \tag{14}$$

The composition table for PA is deceptively simple and looks as follows:

$\bullet$	$<$	$=$	$>$
$<$	$<$	$<$	P
$=$	$<$	$=$	$>$
$>$	P	$>$	$>$

Example 11 *Consider a trivial planning situation as depicted in Figure 11. Actions* `taking a block B` *and* `loading a block A` *are simultaneous ($t_1 = t_2$) and action* `putting a block B` *is performed later than* `taking a block B` *(since $t_2 < t_3$). Obviously, action* `loading a block A` *is earlier performed than* `putting a block B`*. Indeed, the composition table for PA allows us to deduce:*

$$(t_1, t_3) = [t_1 = t_2] \bullet [t_2 < t_3], \text{hence } [t_1 < t_3]\,. \tag{15}$$

Thus, action `loading a block A` *is earlier than* `putting a block B`*.*

Although relations of Point Algebra corresponds well with a common sense-based reasoning (in fact, we often think about temporal processes in terms of single time points), Point Algebra is inappropriate from a theoretic point of view. Allen's Interval Algebra founds much broader area of application.

0.15.2 Allen's Interval Algebra

This calculus – invented by J. Allen in [57] in 1983 – constitutes a natural generalization of Point Algebra for a case of intervals which play a role of atoms in this calculus. A construction of this calculus stems from the observation that two (convex) temporal intervals i and j can be qualitatively related (each to each other) in only thirteen possible ways, so the Allen's interval algebra is also called $\mathcal{A}$-13. [30] Six of these basis-relations have their inverses and one relation ('equivalent to') forms its own inverse, what justifies the number of them. Assuming that two temporal intervals A and B are given, all of Allen's relations between intervals may be visualized as depicted in Figure 12.

These 13 basic relations are equipped by two algebraic operations:

[29] The current presentation is based on a work of A. Gerevini [56].

[30] It is not important whether they are open or closed. A description of a concrete type of topology supposed in this calculus was omitted by J. Allen in his work, but he essentially based on topology induced from $\mathbb{R}$. The same approach was implicitly adopted in majority of reconstructions and presentations.

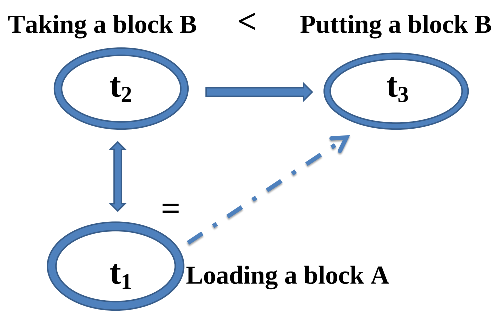


Figure 11: Planning situation with 3 point-actions

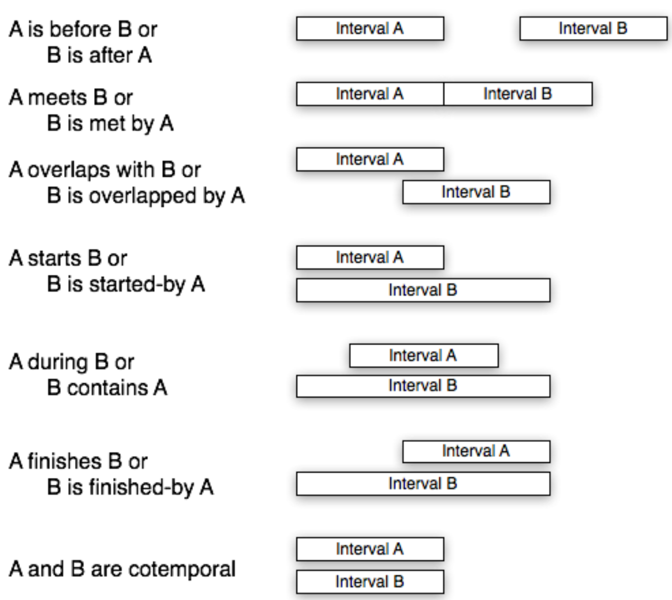


Figure 12: 13 basis Allen relations between intervals.

·	p	m	o	F	D	s	e	S	d	f	O	M	P
p	(p)	(p)	(p)	(p)	(p)	(p)	(p)	(p)	(pmosd)	(pmosd)	(pmosd)	(pmosd)	*full*
m	(p)	(p)	(p)	(p)	(p)	(m)	(m)	(m)	(osd)	(osd)	(osd)	(Fef)	(DSOMP)
o	(p)	(p)	(pmo)	(pmo)	(pmoFD)	(o)	(o)	(oFD)	(osd)	(osd)	*concur*	(DSO)	(DSOMP)
F	(p)	(m)	(o)	(F)	(D)	(o)	(F)	(D)	(osd)	(Fef)	(DSO)	(DSO)	(DSOMP)
D	(pmoFD)	(oFD)	(oFD)	(D)	(D)	(oFD)	(D)	(D)	*concur*	(DSO)	(DSO)	(DSO)	(DSOMP)
s	(p)	(p)	(pmo)	(pmo)	(pmoFD)	(s)	(s)	(seS)	(d)	(d)	(dfO)	(M)	(P)
e	(p)	(m)	(o)	(F)	(D)	(s)	(e)	(S)	(d)	(f)	(O)	(M)	(P)
S	(pmoFD)	(oFD)	(oFD)	(D)	(D)	(seS)	(S)	(S)	(dfO)	(O)	(O)	(M)	(P)
d	(p)	(p)	(pmosd)	(pmosd)	*full*	(d)	(d)	(dfOMP)	(d)	(d)	(dfOMP)	(P)	(P)
f	(p)	(m)	(osd)	(Fef)	(DSOMP)	(d)	(f)	(OMP)	(d)	(f)	(OMP)	(P)	(P)
O	(pmoFD)	(oFD)	*concur*	(DSO)	(DSOMP)	(dfO)	(O)	(OMP)	(dfO)	(O)	(OMP)	(P)	(P)
M	(pmoFD)	(seS)	(dfO)	(M)	(P)	(dfO)	(M)	(P)	(dfO)	(M)	(P)	(P)	(P)
P	*full*	(dfOMP)	(dfOMP)	(P)	(P)	(dfOMP)	(P)	(P)	(dfOMP)	(P)	(P)	(P)	(P)

=

Figure 13: Composition of any two basic relations of Allen-type. The picture may be found under: http://www.ics.uci.edu/alspaugh/cls/shr/allen.

1 an unary *inversion* of a relation (denoted as R^{-1} for a relation R) and

2 a binary *composition* of two relations (denoted as $R_1 \bullet R_2$ for some R_1 and R_2).

For $|I|$ being a universe of all 13 basic Allen's relations, Allen's interval algebra $\mathcal{A}-13$ is a structure $\langle A,^{-1}, \bullet\rangle$, where $^{-1}, \bullet\rangle$ are algebraic operations on $|I|$ – as above.

It appears that $\mathcal{A}$-13 constitutes an enormous and (generally) elusive algebraic object with a number of 8192 possible composed relations. Out of the 8192 relations in the interval algebra, only 27 appear as compositions of basic relations, and each of those comprises either 1, 3, 5, 9 (concur) or all 13 (full) basic relations! Therefore, only some subcases of $\mathcal{A}$-13 are tractable – due to [58].

The composition of any two basic relations are presented in table below.

Example 12 *Consider an except of the Traveling Salesman Schedule and a list of* 3 *cities in France visited by Salesman. An order of the Salesman's visits is as follows: Salesman visits Bordeaux before visiting Marseille and a visiting Marseille meets a visit in Paris. In order to deduce a temporal relation between Paris and Bordeaux, it is enough to compute* `before• meet = before` *– due to the table above.*

0.15.3 Halpern-Shoham Logic

Allen's algebra of interval relations may be naturally represented in terms of a modal logic with operators interpreted by 13 basic Allen's relations. Such a modal system for Allen's algebra is to be known as Halpern-Shoham Logic (HS). This logical representation of Allen's relations forms a non-direct representation of quantitative temporal constraints, what justifies its description in this paragraph. Halpern-Shoham Logic – introduced in [59] – constitutes a concurrent approach to temporal reasoning w. r. t. the Computational Tree Logic (CTL) or the Linear Temporal Logic LTL – the more traditional and point-wise approaches.

More, precisely, HS modally represent the following temporal relations: **a**fter" (or **m**eets"), (**l**ater"), **b**egins" (or **s**tart"), **d**uring", **e**nd"(or **f**inish"),**o**verlap", **e**quivalent" and their inverses, they rendered in HS by corresponding modal operators: $\langle A\rangle$ for **a**fter", $\langle B\rangle$ for **b**egins", $\langle D\rangle$ for **d**uring", etc. The full syntax of HS-entities ϕ is given by a grammar:

$$\phi := p\ |\neg\phi\ |\phi\wedge\phi|\langle R\rangle\phi|\langle\bar{R}\rangle\phi\,, \tag{16}$$

where p is a propositional variable and $\langle\bar{R}\rangle$ denotes a modal operator for the inverse relation with respect to $R \in \{A, B, D, E, O, L\}$.

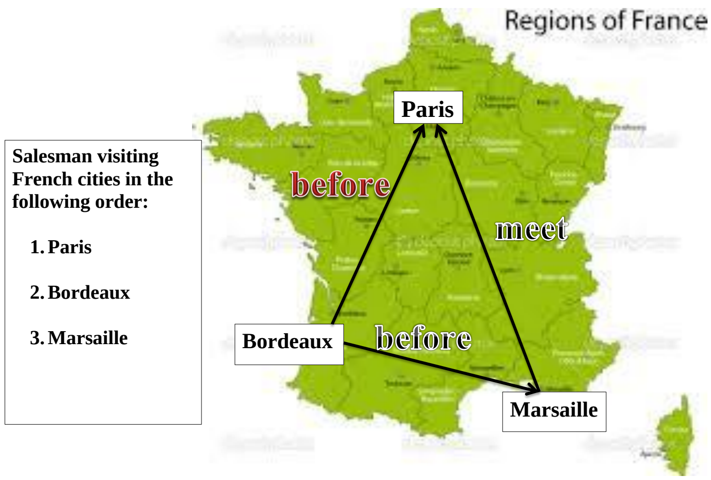


Figure 14: A Salesman Traveling between French cities

Semantics. Generally, HS-semantics of HS is based on models with a domain (universe) containing intervals. Relations in these models are Allen's temporal relations.[31] More precisely – denoting by M a set of all discrete intervals[32] – the HS-model is defined as n-tuple $\mathcal{M} = \langle M, R, V\rangle$, where $R \in \{A, B, D, E, O, L\}$ and $V : \mathcal{P}(Prop) \mapsto M$ is a valuation for propositional letters of $\mathcal{L}(HS)$. If $\phi \in \mathcal{L}(\text{HS})$ and $\mathcal{M}$ is such a model[33], and I is an interval in the $\mathcal{M}$-domain, then the satisfaction for the HS operators looks as follows:

$$\mathcal{M}, I \models \langle R\rangle\, \phi \text{ iff there is such an interval I}' \text{ that } I\,R\,I' \text{ and } \mathcal{M}, I' \models \phi\,. \tag{17}$$

In order to illustrate how HS-logic can be exploited for a use of planning situations with temporal constraints let us consider some except of the Traveling Salesman Schedule.

Example 13 *Consider the same extract of a travel history of the Salesman's Traveling from example 12 (Figure 14). If an atomic formula* $visit^{Bor}_{Sal}$ *denotes a (fact of) visiting Bordeaux by Salesman*[34]*, the above equality* `before• meet = before` *from last example may be rendered by the following HS-formula:*

$$\langle B\rangle\langle M\rangle visit^{Bor}_{Sal} \leftrightarrow \langle B\rangle visit^{Bor}_{Sal}\,. \tag{18}$$

0.15.4 State of the Art

Allen's interval relations were initially introduced and described by J. Allen in 1983 in [57]. A problem of tractable subclasses of Allen's Interval Algebra was described in details in [58]. In [60] the 4-consistency algorithm for the full IA algebra was proposed. Slightly paradoxically, point algebra (PA) was introduced later in [55] in 1986, recently discussed from the algorithmic perspective in [56]. In [61] path consistency problems for PA were discussed and solved with respect to some subset of PA. The renaissance of research on

[31] They play a role similar to accessibility relation in Kripke frame-based models.
[32] Obviously, $M \subseteq \mathcal{P}(N)$, for N.
[33] Of course, this model forms a kind of a Kripke frame-based model
[34] This fact may be also expressed by the appropriate 2-argument predicate. We reject this solution for a use of a simplicity of a presentation in terms of the HS-logic language.

Allen's algebra IA and its representations was strictly connected with a birth of Halpern-Shoham Logic (HS) – introduced in 1991 by J. Halpern and Y. Shoham in [59] – as suitable for a modal representation of IA. This system and its subsystems were recently intensively explored from the perspective of their meta-logical features. Decidability and undecidability of some HS-fragments interpretable over discrete linear order were investigated in [62], over finite linear orders – in [63], over natural numbers – in [64]. Taxonomy of these results was elaborated and presented in the form of a road map of HS-subsystem in [65, 66]. Finally, some epistemic extension of HS was recently proposed by Lomuscio in [67].

0.16 Quantitative Temporal Constraints, or Temporal Constrains Satisfaction Problem and its specifications

Considerations of Section 2 elucidated emphasize qualitative nature of temporal constraints. In this chapter a *quantitative* and computable nature of temporal constraints will be emphasized in terms of the so-called *Temporal Constrains Satisfaction Problem and its specifications* (TCSP).

TCSP constitutes a temporal extension of the so-called *Constraints Satisfaction Problem* (CSP), which forms a general problem-solving paradigm – broadly applicable in planning and scheduling, pattern recognition or consistency checking[35]. TCSP could be briefly specified as follows. Taking a set of variables and a set of constraints on their admissible values[36] a problem is to find a value for each variable such that all variables meet these constraints ([3], p. 167.).

0.16.1 Temporal Constrains Satisfaction Problem – a Detailed Specification

We specify now TCSP in a more formal way. According to [20, 19] – TCSP refers to variables $X_1, X_2 \dots X_n$ – each with a continuous domain and representing time points. Each constraints in TCSP is represented by a set of intervals: $\{I_1, I_2 \dots, I_n\}$ =$\{[a_1, b_1], [a_2, b_2], \dots, [a_n, b_n]\}$. Due to [20, 68] – two following types of TCSP-constrains are traditionally distinguished:

A unary temporal constraints, T_i,

B binary temporal constraints $T_{i,j}$.

The unary (quantitative) temporal constraint, T_i restricts a single time variable X_i and has a form:

$$a_1 \leq X_i \leq b_1 \vee \dots \vee a_n \leq X_i \leq b_n. \tag{19}$$

The binary (quantitative) temporal constraint, $T_{i,j}$ restricts pairs of time variables X_i, X_j and has a form:

$$a_1 \Big(\leq X_j - X_i \Big) \leq b_1 \vee \dots \vee a_n \leq \Big(X_j - X_i \Big) \leq b_n. \tag{20}$$

In such a framework some elementary set-theoretic operations on TCSP's may be naturally introduced. (For simplicity, we consider a simple interval $[a, b]$ instead of their set [37].)

Definition 10 ***(Operations on TCSP's.)*** *Let I and J be intervals corresponding to (time variables) $X_j - X_i$ for some indexes i, j. Then:*

- *the union of I and J, $I \bigcup J$ admits all values that are allowed by either one of them: $I \bigcup J = \{t : t \in I \vee t \in J\}$,*

[35] This area forms a primary and natural application area of CSP and of a variety of its extensions.
[36] That means: 'imposed on values that variables can take'.
[37] Majority of authors considers a broad set of temporal intervals in definitions of TCSP's-operations, see: [20, 19].

- *the intersection of I and J, $I \cap J$, admits only these values that are allowed by both of I and J: $I \cap J = \{t : t \in I \wedge t \in J\}$,*
- *the composition of $I = [a, b]$ and $J = [c, d]$, $I \otimes J = \{[a + c, b + d]$, where $+$ is a normal additive operation between real numbers$\}$.*

Consistency. TCSP problem was introduced as a formalized conceptualization of usual temporal planning situations in order to check a consistency of choices of action sequences. In such considerations, quantitative temporal constrains (of TCSP) are often represented by the appropriate networks or the so-called *distance graphs* $G = \langle X, E \rangle$. Their relationship is simple: A TCSP-network $\langle X, C \rangle$ can be transformed into a distance-graph $G = \langle X, E \rangle$ as follows. Each constraint $a_{ij} \leq X_j - X_i \leq b_{ij}$ of TCSP-network defines a two-labeled arc in G: from X_j to X_i and from X_i to X_j.

In this way, there exist two alternative methods to measure TCSP-consistency: in terms of TCSP-networks or in terms of TCSP-distance graphs. In order to illustrate them let us assume that c_{ij}, c_{ij} and c_{kj} denote some values of (quantitative) temporal constraints and $d(i, j)$ is a minimal distance between vertices i, j in a distance graph G. In such a framework – according to [69, 70] – these criteria may be expressed as follows:

A In terms of TCSP-distance graphs: a network is inconsistent if there is a negative cycle, i.e a vertex such that $d(i, i) < 0$,

B In terms of TCSP-networks: a network is inconsistent if $c_{ij} \cap [c_{ik} \otimes c_{kj}] \neq \emptyset$.

The role of last criterion is reflected by the fact that it forms a 'core' of the following well-known *consistency path algorithm* for TCSP[38] :

```
PC(X, C)
    for each k : 1 ≤ k ≤ n do
        for each pair i, j: 1 ≤ i < j ≤ n, i ≠ k, j ≠ k do
            c_ij ← c_ij ∩ [c_ik ⊗ c_kj]
            c_ij = ∅ then exit (inconsistent)
end
```

The following algorithm illustrates how to exploit the PC-algorithm[39].

Example 14 *Consider a historic situation of the Titanic sinking on 12th April of 1912 year with 5 key time-points of this scenario: A – a time point of a Titanic collision with an Ice Berg, B – a time point, when Titanic begins to sink, C – a time point when the Titanic hull is completely sunk, D – a time point when Carpathia begins to move towards a Titanic position and E – a time point of achieving the Titanic position by Carpathia. Temporal constraints imposed on distances between A, B, C, D are presented on a Figure 15. The question is the following: does Carpathia have a chance to reach the Titanic position if it is known that Titanic will completely sink between 5-6 h from its collision with the Ice Berg?*

[38] The path-consistency algorithm was introduced in a pioneering Allen's paper [57].

[39] Nevertheless – in generality – a consistency problem for TCSP belongs to a class of NP-hard problems. That means that it is hard in a class of NP-problems. To make the matter worse, a reduction of the considered intervals does not change the complexity radically. Namely, it holds the following theorem(The proof of this theorem follows from a reduction of this problem to 3-coloring problem.).

Theorem 1 *(Dechter, 1991, [20])*

A *Deciding consistency for TCSP is NP-hard,*

B *Deciding consistency for TCSP with no more than two intervals per edge is NP-hard.*

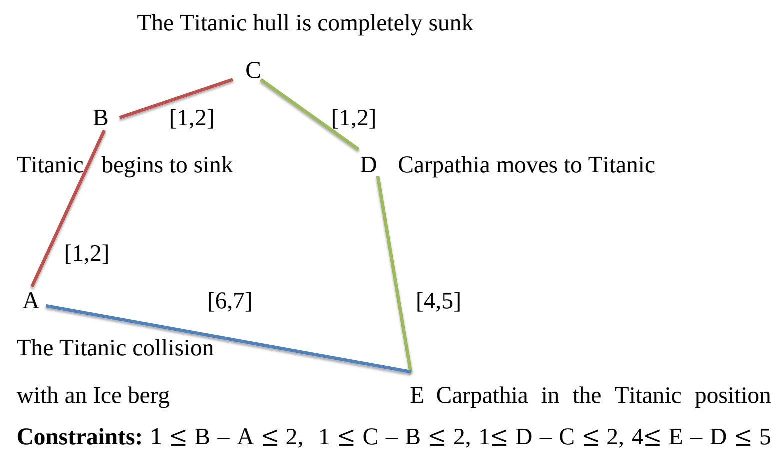


Figure 15: Temporal constraint network for the Titanic catastrophe

Solution 1 *It is known that* $5 \leq (T - A) \leq 6$, *where* T *is a time point of a complete sinking of Titanic. Thus, the above question could be reformulated to a question whether a composition interval* $[A, B] \otimes [B, C] \otimes [C, D] \otimes [D, E]$ *is contained in the time interval [0, 5] or [0,6]. (The red and green lines represent the considered time intervals. A blue line represent a time interval between points A and E that should be computed.)* $[A, B] \otimes [B, C] \otimes [C, D] \otimes [D, E] = [1, 2] \otimes [1, 2] \otimes [1, 2] \otimes [4, 5] = [1 + 1 + 1, 2 + 2 + 2 + 5] = [7, 11]$ *– according to TCSP'-operation* $\otimes$*, given in Definition 10. Hence*

$$[A, E] \cap ([A, B] \otimes [B, C] \otimes [C, D] \otimes [D, E]) = [5, 6] \cap [7, 11] = \emptyset\,, \tag{21}$$

thus the presented PC-network is inconsistent and Carpathia will not reach the Titanic's position before its sinking.

These facts justify relatively high intractability of temporal problems in terms of TCSP. Fortunately, this situation becomes operationally comfortable if we move from a general framework of TCSP to its special subproblem, called *Simple Temporal Problem.* We define it in the next paragraph.

0.16.2 Simple Temporal Problem

Simple Temporal Problem is such a simple subproblem of TCSP that all constraints are specified by a single interval [40]. More formally, *the Simple Temporal Problems*(STPs) is the kind of the Constraints Satisfaction Problem, where a constraint between time-points X_i and X_j is represented in the constraint graph as an edge $X_i \to X_j$, labeled by a single interval $[a, b]$ that represents the constraint:

$$a \leq (X_j - X_i) \leq b. \tag{22}$$

[40] The conceptual basis of this chapter presentation makes use the following works [20, 17, 18, 19].

Alternatively, this constraint may be represented by a pair of inequalities:

$$(X_j - X_i) \leq b, \quad (X_j - X_i) \leq -a. \tag{23}$$

Solving an STP means to find an assignment of values to variables such that all temporal constraints are satisfied – according to [19]. It appears that STPs can be solved in polynomial time, whereas the complexity of a general TCSP belongs to a class of NP-problems.

Generally, STP is sometimes associated to the so-called distance edge-weighted graph $G = \langle V, E \rangle$ where V is a set of vertices and E – a set of edges between them. Each edge $i \to j$ (between vertices i, j) is labeled by a weight a, representing the linear constrain $(X_j - X_i) \leq a$. If there is more than one path between i and j, one can consider an intersection of all these paths in order to obtain a new constraint: $(X_j - X_i) \leq d$, where d is the shortest path from i to j (see:[20], p.68).

Since further extensions of STP are not used in the proper body of the thesis, their description will be omitted. Thus, let us interrupt this presentation in this point. (More advanced considerations may be found in Appendix.) Instead of it, let us focus our attention on the notion of preference.

0.16.3 Preferences

Preferences – a general specification.

Preferences deserve a special attention. In fact, they introduce some portion of rationality to temporal reasoning as they reflect our wishes or intentions with respects to different objects or activities (for example: with respect to a choice of actions in planning etc.)[41]. This notion of preference allows us to better grasp a dynamic nature of decisions, games or planning and it can refer to various sorts of entities, such as: worlds, propositions or actions [74].

Preference is sometimes identified[42] with a binary comparative relation "at least as good as", denoted by $\preceq$[43]. Thus, the following four properties are usually considered to be part of the meaning of such a concept of a (weak) preference:

Axiom1 $A \prec B \to \neg(B \prec A)$ *asymmetry of preference*,

Axiom2 $A \prec B \wedge B \prec C \to A \prec C$ *transitivity of preferences*,

Axiom3 $A \preceq A$ *reflexivity*,

Unfortunately, there is a little common consensus among logicians about relational semantic representation of preferential features. For example, transitivity forms such a controversial property and many examples – such as *Sorites Paradox* – have been offered to show that it should not hold in general[44]. By contrast, Davidson underlines some arguments for a normative appropriateness of preference transitivity and he suggests that transitivity is constitutive for the meaning of preference[45]. A variety of semantic contexts, that preferences can be involved in, can be elucidated by the following 3 situations of preferred choices.

[41] A notion of preferences plays a central role in a variety of multi-contextual research branch, including a moral philosophy, psychology, decision and game theory, logic and AI. The philosophical and psychological provenance of the earlier research on preferences and their nature can be detected in works of Armstrong in 30th and 40th such as [71, 72]. Preferences in the contexts of economic analysis was discussed also relatively early by F. Ramsey in 1928 in [73].

[42] Although they form a significant component of many reasoning types in AI, there is no one consensus between AI-researcher with respect to a unified method of their representation and defining.

[43] This relation is usually called *weak preference* in a contrast to *a strong preference*, denoted by $\prec$.

[44] The *Sorites Paradox* employs a series of objects that are so arranged that we cannot distinguish between two adjacent members of the series, whereas we can distinguish between members at greater distance ([71, 72].). Consider 1000 cups of coffee, numbered $C_0, C_1, C_2 \ldots$ up to, for example, C_{99}. Cup C_0 contains no sugar, cup C_1 one grain of sugar, cup C_2 two grains etc. Since one cannot taste the difference between C_{99} and C_{98}, they can be considered as equally good (of equal value), $C_{99} \sim C_{98}$. For the same reason, we have $C_{98} \sim C_{97}$, etc. up to $C_1 \sim C_0$, but clearly $C_0 > C_{99}$.

[45] Constructing an analogy to length measurement, Davidson in [75](p.73) asks: If length is not transitive, what does it mean to use a number to measure length at all?"

Example 15 *(Point-wise preference.) We can prefer to go to the cinema at 12:00 with a preference $\frac{3}{10}$ and tomorrow at 17:00 a preference $\frac{7}{10}$. And only in our birthday we are strongly motivated with a preference = 1 to go to the cinema for each seans.*

Example 16 *(Interval comparative preference.) We can prefer to go to the cinema today afternoon more than tomorrow evening.*

Example 17 *(Interval non-comparative preference.) We strongly prefer to go to the cinema than something other.*

Each of these situations (with point-wise preferences) should be modeled in a different way. The first situation (point-wise preferences) rather presupposes a point-wise semantics, the second and the third situation – the interval-based semantics. Whereas a presented point-wise modeling of preferences has a long tradition and is reflected in a subject literature, the interval-based modeling still waits for a broader investigation. Thus, the presented interval-based approach describes ideas of the author of the thesis from [76] – based on a formalism from [77, 78] and [79]. There will be *Fuzzy Modal Preferential Logic* (FMPL), based on a Quantitative Possibillistic Fuzzy Logic from [79, 77].

Preferences - in a more detailed depiction.

A majority of point-wise approaches such as [80, 81, 82] stems from the observation that preferences may be rendered by the appropriate modal formulas of a dynamic or epistemic logic. Thus, preferences are often rendered in a *modal preferential language*, usually based on a grammar:

$$p\,|\neg\phi|\phi\wedge\psi|\langle\mathrm{Pref}\rangle_i\phi\,|[\mathrm{Pref}]_i\phi^{46}. \tag{24}$$

Modalities: $\langle\mathrm{Pref}\rangle_i\phi\,,[\mathrm{Pref}]_i\phi$ may be read: ϕ is (weakly) preferable and ϕ is (strongly) preferable (respectively)[47].

Despite some gap between the formal sense of these operators and a common usage of preferences in a natural discourse, this modal description of preferences has some important advantage: it naturally and (almost) immediately justifies an appropriateness a pointwise Kripke frame-based semantics for preferences. Intuitively, the syntactic description means that some states t (considered by an agent i as good as the initial state u) have a property ϕ (satisfy ϕ). For a 'box'- type operator a word 'some' means 'for all', for a 'diamond'-type operator it means 'for at least one'. (see: Fig.8).

The point-wise semantics. Obviously, this translation into semantics requires a kind of accessibility relation between states u and t. This relation – dependently on the preferred convention – is just a *preference relation* and it can be a kind of order[48]. It naturally leads to a defining of Kripke models for preferences as a tuple of the form:

$$\mathcal{M}=\langle U,\preceq_i,V\rangle. \tag{25}$$

where U is a non-empty set of states (possible worlds) $\preceq_i$ is the appropriate binary *preference relation* and V is a usual valuation. We read: $s\preceq_i t$ as 't is at least as good as s' or 't is preferred as s'. This pointwise Kripke semantics preserves a comparative nature of preferences – independently of a fact that preferences are rendered by unary modal operators in preferential languages.

[46]Sometimes, some additional operators such as $K_i\phi$ (an agent i knows that ϕ) are considered.

[47]Alternatively, one may state that $[\mathrm{Pref}]_i\phi$ says that all worlds – which the agent considers as least as good as the current one – satisfy ϕ and similarly for $\langle\mathrm{Pref}\rangle_i\phi$. We only exchange: *all* for *at least one*.

[48]They can form partial orders as in [83], total orders or linear orders as in [80, 81].

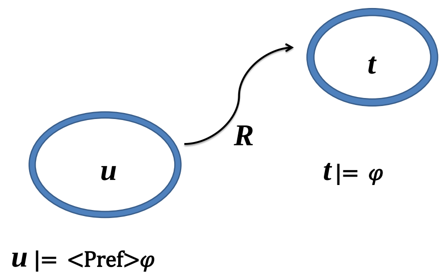


Figure 16: A poinwise Kripke-based semantics for preferences. In a world t (R-accessible from u) is satisfied all (ϕ), what is possible in u.

Preferences versus (temporal) constraints?

The above considerations through the light on mutual relations between preferences and constraints. Putting aside a historical of their different conceptual provenance (preferences stem for the area of decision theory, economy and psychology, meanwhile constraints – from logic and AI) – one could detect some further and fundamental differences between these entities:

Diff.1 (temporal) constraints *should* be satisfied in order to ensure a consistency of plans, systems etc., but preferences *should not*,

Diff.2 (temporal) constraints are divided into two disjoint groups: the quantitative and the qualitative ones. Meanwhile, preferences refer to some *quantities and degrees* (expressed by adverbs: much, weak, strongly, etc.), which can be *naturally* measured in a quantitative way (for example by associating some values $\alpha \in [0, 1]$ to them)[49].

Nevertheless, both types of AI-entities show some similarity. They can be equally imposed on the same types of components of temporal reasoning, i.e. for actions, events, objects, plans, schedules, decisions etc. Moreover, temporal constraints also form a gradual hierarchy – as they may be divided into *weak* and *strong* constraints. By such a distinction, only a satisfaction of strong constraints is necessary from the point of view of a preserving of consistence of systems, plans, schedules, etc. This makes weak constraints more similar to preferences[50]. Some additional similarities and differences will be discussed in next chapter of the thesis.

0.17 Two Depictions of Fuzzy Temporal Constraints of Allen's Sort

In this section, two different depictions of fuzzy temporal constraints of Allen's sort will be described. The first one stems from research of De Cock-Shockert's school. This approach adopts the appropriate portion of a calculus of fuzzy relations and it is theoretically well-founded. Inventors of this approach propose an idea of fuzzification of Allen's relations between fuzzy intervals. They begin with a single non-fuzzy relation

[49]It does not mean that such a solution is absolutely impossible in a case of temporal constraints, but it seems to be much less natural and it seems to redefine a primary meaning of temporal constraints.

[50]Authors of [84] give venture to their radicalism in this area thinking that soft temporal constraints can be measured by preference functions associated to semi-rings. Thus, they are willing to (at least partially) identify these categories.

'before' in order to introduce its fuzzified versions: 'approximately at the same time as before' and 'long before'. These initial fuzzy relations make form a basis for further fuzzification.

An alternative approach to fuzzification of Allen's relations has been recently proposed by H-J. Ohlbach in terms of integrals and convolutions. Hence, this approach may be viewed as a more 'analytic' one. The integral component is introduced here – due to [85, 86] – thanks the Integral Mean Theorem. This theorem allows us to define fuzzy Allen's relations as the averaged and normalized values of functions representing Allen's relations between fuzzy intervals. The complete taxonomy of fuzzy Allen's relations was introduced by Ohlbach in the conceptual framework of his approach.

The presentation of these two depictions will be prefaced by a terminological framework used by inventors of these approaches.

0.17.1 Terminological Framework

Allen's relations once again

As earlier described in subsections 1.2, 1.3 and 1.4 of 'Introduction', time points and time intervals form elementary conceptual atoms of temporal reasoning with temporal constrains. These 'atoms' may be combined together The easiest case is determined by a situation when a single time point remains in some relation with respect to a (normal)[51] interval in $\mathbb{R}$ or[52] with respect to a crisp interval in $\mathbb{R}^2$.

These point-interval relations can be further naturally extended to the interval-interval relations. In this way we obtain the Allen's interval-interval relations – introduced in [57] as the relations between two intervals without metric – as depicted on Fig. 1

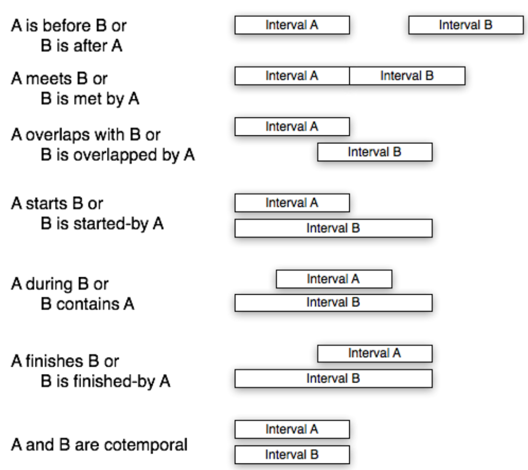


Figure 17: 13 basis Allen relations between intervals.

Consider a case of $\mathbb{R}^2$. The simplest possible situations is determined by crisp intervals. Obviously, the situation gets much more complicated if we take into account fuzzy intervals instead of crisp temporal intervals. A need to consider such intervals is justified by a plurality of imprecise practical contexts denoted by such phrases as "almost in the night', 'last year', 'around midnight', etc. In order to extrapolate earlier considerations for a case of fuzzy intervals, two new formal definitions of fuzzy sets and Lebesgue integrals will be introduced in the next paragraph.

[51]That means: compact and continuous).

[52]We consider $\mathbb{R}$ more generally – rather than a topological space than a metric one. Note that each metrics defines a topology, but not in the inverse direction. This approach follows from a fact that we intend to see Allen's relations as qualitative temporal constrains.

Fuzzy intervals and basic operations on them

Normal, compact intervals – considered in Allen's interval algebra – may be naturally defied by their start- and end points. By contrast, fuzzy intervals – as objects in $\mathbb{R}^2$ – cannot be defined in this simple way. The commonly used convention to define them relies on defining the appropriate function which somehow determines these intervals. As usual, this characteristic function is to defined as Riemann's integrable.

Definition 11 ***(Fuzzy interval)***. *Assume that $f : \mathbb{R} \mapsto [0.1]$ is a Riemann's integrable function on $\mathbb{R}$*[53]. *Then the fuzzy interval i (corresponding to a function $f^{i(x)}$) is defined as follows:*

$$i(x) = \{(x, y) \in \mathbb{R} \times [0, 1] : y \leq f^{i(x)}(x)\}. \tag{26}$$

It means that each fuzzy interval is a 2-dimensional 'object'. The simplest ones consist of a core $C(i)$ and a support $S(i)$. The core is the part of the interval i where fuzzy values are equal to 1 and the support – where values are greather than 0 – as depicted in Figure 18. The first x-coordinate is a line parallel to y-axis, where y-values moves to a constants value 1 and the last x-coordinate – is a line, when an established value 1 moves to values from a set $[0, 1]$. Obviously, both coordinates single out the core of each fuzzy interval. They are usually denoted by i^{fK} and i^{lK}. The size of $|i| = \int_a^b f^{i(x)}(x)\mathrm{d}x$, for $f^{i(x)}(x)$ defined on $[a, b]$. Unfortunately, most of typical set-theoretic Boolean operations cannot be defined for fuzzy intervals

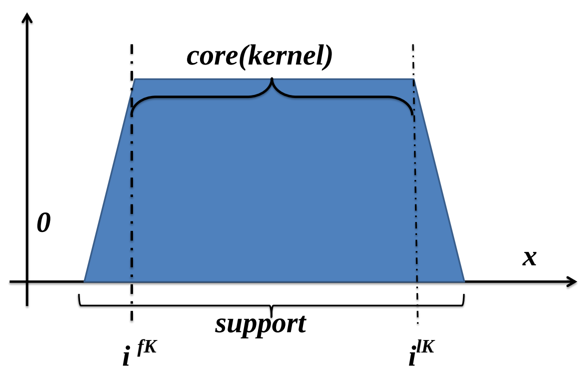


Figure 18: Core, support and x-coordinates of a fuzzy interval

in the same way as for ordinary intervals. Instead of it, there is a lot of axiomatically characterized unary and binary transformations – called t-norms – that enable to introduce these operations. Each t-norm $\mathcal{T}$ is an increasing mapping from $[0, 1]^2 \rightarrow [0, 1]$ satisfying the following conditions: $\mathcal{T}(1, a) = a, \mathcal{T}(a, b) = \mathcal{T}(b, a), \mathcal{T}(\mathcal{T}((a, b), c) = \mathcal{T}(a, \mathcal{T}(b, c))$. Finally, the condition $\mathcal{T}(a, c) \leq \mathcal{T}(a, b) + \mathcal{T}(b, d)$ is also required.

Operations of an intersection and a union of two fuzzy intervals are classically defined in terms of t-norms as follows:

$$(i \cap j)(x) =^{\text{def}} \min\{i(x), j(x)\} \tag{27}$$

$$(i \cup j)(x) =^{\text{def}} \max\{i(x), j(x)\}, \tag{28}$$

[53]It is a well-known fact of mathematical calculus that there exist some Riemann's integrable functions that are not continuous. This relatively weak condition allows us to preserve a skeleton of a fuzzy interval and it is introduced for a use of further analysis.

but a class of admissible norms is wider and it contains (for example) the so-called Hamacher family norms[54]. One of the most useful operations on fuzzy intervals is a cut-relation, which allows us to consider only a chosen part of (usually infinite) intervals between a chosen point x_1 and x_2. Formally:

$$cut_{x_1,x_2}(i)(x) =^{def} 0 \text{ if } x < x_1 \text{ or } x \geq x_2 \text{ or } i(x) - \text{otherwise}. \tag{29}$$

This situation may be generalized exchanging Riemann integrability for Lebesgue integrability. It allows us to admit a situation that a function – determining a fuzzy interval – can have some points of discontinuity. This solution seems to be a reasonable one in the light of our schedule-planning problems[55].

Riemann integrals and Lebesgue integrals. We assume that concepts of a measure and of Riemann's integral are known (see: Appendix). The main difference between Riemann's integrals and Lebesgue integrals consists in a different way of approximation of considered functions. Namely, taking a function $f : [a, b] \mapsto \mathbb{R}$, we can approximate a Riemann's integral $\int_a^b f(x)dx$ by vertical polygonal posts – as depicted on Figure 19 – by sums $\sum_i f(x_i)\triangle(x_i - x_{i-1})$ for $i \mapsto \infty$. The same function may be approximated by *horizontal* posts representing $\mu\{x : f(x) > t\}$ and – arithmetically – by sums $\sum_i a_k\mu\{x : f(x) > t\}$ for some real[56] a_k and a given measure μ. The Lebesgue integrals is just based on this way of approximation. Its formal definition is as follows.

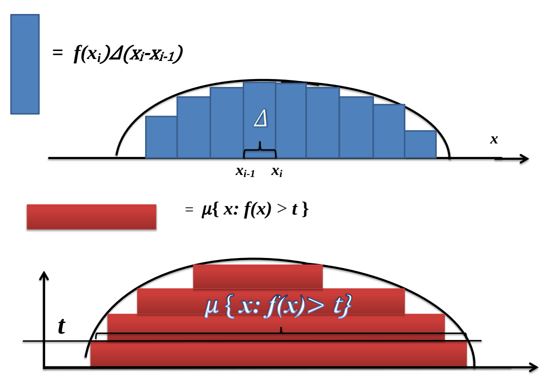


Figure 19: Difference between Riemann and Lebesgue integrals.

Definition 12 ***(Lebesgue integrals.)*** *Assume that a function $f : \mathbb{R} \mapsto \mathbb{R}_+$ is Riemann integrable in its domain. Let also assume that a denumerable additive measure μ is given, i.e. for all pairwise disjoints sets $A_1, A_2 \ldots \in A$ such that $\bigcup_{i=1}^{\infty} A_i \in A$ it holds: $\mu(\bigcup_{i=1}^{\infty} A_i) = \Sigma_{i=1}^{\infty}\mu(A_i)$. Let also define $f^*(t) = \mu\{x|f(x) > t\}$ for some $t \in \mathbb{R}_+$. The Lebesgue integral of f is then defined as:*

$$\int f d\mu = \int_0^{\infty} f^* dt\,, \tag{30}$$

where the right integral forms an ordinary improper Riemann integral.

Definition 13 *Each real function f for which the integral (1) exists is to be called* `Lebesgue integrable` *or* `integrable in Lebesgue sense`*.*

[54] Since we will essentially use the standard definitions, we omit these alternative definitions for brevity of the presentation.

[55] Further profits of this generalization will be emphasized later in further considerations

[56] This way of defining seems to, somehow, justify why discontinuity points are admissible as they are do not violate the structure of horizontal posts; their existence makes no important difference.

Lebesgue integrability may be exploited in specification of characteristic functions for fuzzy intervals. This allows us to generalized the earlier definition of them. The new definition is as follows.

Definition 14 ***(Fuzzy interval)****. Assume that $f : \mathbb{R} \mapsto [0.1]$ is a Lebesgue's integrable function (so not neccessary continuous) on $\mathbb{R}$*[57]*. Then the fuzzy interval i (corresponding to a function $f^{i(x)}$) is defined as follows:*

$$i(x) = \{(x, y) \in \mathbb{R} \times [0, 1] : y \leq f^{i(x)}(x)\}. \tag{31}$$

0.17.2 De Cock-Schockaert's School Approach to Fuzzified Allen's Relations

The approach of De-Cock-Schockaert's school deserves to be called: 'an algebraic' one. In fact, this approach – developed in such papers as: [87, 88, 89] make use of an algebraic apparatus elaborated in the framework of relational calculus and in the algebraic semantics of t-norm-based fuzzy logic. As earlier mentioned, the class of t-norms allows to define operations on fuzzy relations.

Namely, if R, S are fuzzy relations (i.e. R is a fuzzy set in $U \times V$ and S – a fuzzy set in $V \times W$) and $\mathcal{T}$ is a t-norm, then one can define *sup$\mathcal{T}$-composition* (see:[90], pp. 298-99):

$$R \bullet S(u, w) = \sup_{v \in V} \mathcal{T}(R(u, v), S(v, w)). \tag{32}$$

Introducing also a new function by the condition:

$$I(a, b) = \sup\{\lambda | \lambda \in [0, 1] \text{ and } T(a, \lambda) \leq b\}, \tag{33}$$

one can introduce the so-called: *subproduct* $R \lhd_I S(u, w)$ and *superproduct* $R \rhd_I S(u, w)$ defined (*resp.*) as follows:

$$R \lhd_I S(u, w) = \inf_{v \in V} \mathcal{T}(R(u, v), S(v, w)). \tag{34}$$

$$R \rhd_I S(u, w) = \inf_{v \in V} \mathcal{T}(S(u, w), R(w, v)). \tag{35}$$

This definitions are suitable to be extended to the more sophisticated definitions such as:

$$A \lhd_I (R \rhd_I B) = \inf_{u \in U} \mathcal{I}(A(u), \inf_{v \in V} \mathcal{I}(B(v), R(u, v)). \tag{36}$$

Fuzzified Allen's relations. This basic algebraic framework forms a convenient bridgehead for inventors of this approach to the fuzzification of Allen's temporal relations. Obviously, Allen's relations may be fuzzified in a different way. For example, one could begin with some motivating examples of fuzzy Allen's relations such as: a occurs long after b' or 'a occurs after or approximately at the same time as b' and 'a occurs just after b'. For a given pair of points (u, v) De Cock, Schockaert decided to take the similar fuzzified relations: 'v is long before u ' and 'v before or approximately at the same time'– denoted by $L^{\leq}(u, v)$ and $L^{\preceq}(u, v)$ (*resp.*) – as as basis for further fuzzification.

Observe now that $L^{\leq}(u, v)$ renders the extent to which u is much smaller than v. Similarly, $L^{\preceq}(u, v)$, expresses the extent to which u is smaller or approximately equal to v (see:[90], p. 302, [88], p. 111.). Assume now that A, B are fuzzy interval and $v \in A$ and $u \in B$. Which fuzzified Allen's relations may be rendered by $A \lhd_I L^{\leq} \rhd_I B$? The case of $L^{\leq}(u, v)$ suggests that this relation defines fuzzified 'before'-relation. This conjecture may be justified as follows.

In fact, note that:

$$A \lhd_I (L^{\leq} \rhd_I B) = \inf_{u \in U} \mathcal{I}(A(u), \inf_{v \in V} \mathcal{I}(B(v), L^{\leq}(u, v)). \tag{37}$$

[57]It is a well-known fact of mathematical calculus that there exists some Riemann's integrable functions that are not continuous. This relatively weak condition allows us to preserve a skeleton of a fuzzy interval and it is introduced for a use of further analysis.

Interpret this equality. At first, we should take inf from all $v \in B$, such that A is *long before* them (the internal parentheses). Secondly, we look at the external $\inf_{u \in U}$ from $\mathcal{I}$-values. The definition (48) of I demands to take sup from these $u \in A$ to be smaller than $\inf_{v \in V} \mathcal{I}(B(v), L^{\leq}(u, v)$. Hence, we take the end part of A. Finally, the whole relations expresses the degree to which the end of A is much smaller than the beginning of B. It allows us to write:

$$\text{before}(A, B) = A \lhd_I L^{\leq} \rhd_I B. \tag{38}$$

Consider now two relations: $A \lhd_I L^{\preceq} \rhd_I B$ and $B \bullet_I L^{\preceq} \bullet_I A$. The first one expresses the degree to which the end of A is smaller than or approximately equal to the beginning of B. Similarly – the second one expresses the degree to which the end of B is smaller than or approximately equal to the beginning of A. (See: [90], p. 302).

It is not difficult to see that if we take a minimum from both relations, i.e. $\min\{A \lhd_I L^{\preceq} \rhd_I B, B \bullet_I L^{\preceq} \bullet_I A\}$, we are able to formally define 'meet'-relation between them. Thus,

$$\text{meets}(A, B) = \min\{A \lhd_I L^{\preceq} \rhd_I B, B \bullet_I L^{\preceq} \bullet_I A\}. \tag{39}$$

Similar reasoning can be applied to the other qualitative Allen's relations, what leads to their following definitions, for as parameter $\beta \in [0, +\infty]$[58](see: [88], pp. 111-12).

Name	**Notation**	**Definition**
before	$b(A, B)$	$A \lhd L^{\leq}_{\beta} \rhd B$
overlaps	$o(A, B)$	$\min\{A \bullet (L^{\preceq}_{\beta} \rhd_I B), B \bullet L^{\leq}_{\beta} \bullet A, (A \lhd L^{\leq}_{\beta}) \bullet B)\}$
during	$d(A, B)$	$\min\{B \bullet (L^{\leq}_{\beta} \rhd A), (A \lhd L^{\leq}_{\beta}) \bullet B\}$
meets	$m(A, B)$	$\min\{A \lhd L^{\preceq}_{\beta}) \rhd B, B \bullet L^{\preceq}_{\beta} \bullet A\}$
starts	$s(A, B)$	$\min\{(A \bullet L^{\preceq}_{\beta}) \rhd B, (B \bullet L^{\preceq}_{\beta}) \rhd A, (A \lhd L^{\leq}_{\beta}) \bullet B)\}$
finishes	$f(A, B)$	$\min\{(A \bullet L^{\preceq}_{\beta}) \rhd B, (B \lhd L^{\preceq}_{\beta}) \bullet A, (B \bullet L^{\leq}_{\beta}) \rhd A)\}$
equals	$e(A, B)$	$\min\{A \bullet (L^{\preceq}_{\beta}) \rhd B), (B \bullet L^{\preceq}_{\beta}) \rhd A, A \lhd (L^{\preceq}_{\beta}) \bullet B), B \lhd (L^{\preceq}_{\beta}) \bullet A)\}$

Obviously, functions $L^{\leq}$ and $L^{\preceq}$ should be precisely defined – in particular – for computational reasons. It may be done in the following fashion. Since $L^{\leq}_{(\alpha,\beta)}(a, b)$ should represent the fact that a is *long before* b (in terms of parameters α, β, for $a \in \mathbb{R}$ and $b \in \mathbb{R}^{+}$), one can assume that $L^{\leq}_{(\alpha,\beta)}(a, b)$ occurs with a degree 1, if a distance between a and b is greater than $\alpha + \beta$. If this distance is even smaller than α, we can assume that this relations does not occur (it takes a degree 0). Between these two cases, we take a gradual degree of occurrence given by $\frac{b-a-\alpha}{\beta}$. It is reflected by the following definition of $L^{\leq}_{(\alpha,\beta)}(a, b)$ (see: [91], pp. 518-19):

$$L^{\leq}_{(\alpha,\beta)}(a, b) = \begin{cases} 1, \text{ if } b - a > \alpha + \beta\,, \\ 0, \text{ if } b - a \leq \alpha\,, \\ \frac{b-a-\alpha}{\beta}\text{ , otherwise} \end{cases}$$

In contrast, $L^{\preceq}_{(\alpha,\beta)}(a, b)$ measures how near of b is a in terms of parameters of α and β. Namely, it should take 1, if a distance between $b - a$ is possibly minimal. Formally, if $b - a \leq \alpha$. $L^{\preceq}_{(\alpha,\beta)}(a, b)$ it does not occur if a distance is large, i.e. if $b - a \geq \alpha + \beta$. Otherwise, we take a gradual degree. It leads to the following definition.

$$L^{\preceq}_{(\alpha,\beta)}(a, b) = \begin{cases} 1, \text{ if } b - a \leq \alpha\,, \\ 0, \text{ if } b - a \geq \alpha + \beta\,, \\ \frac{b-a+\alpha+\beta}{\beta}\text{ , otherwise} \end{cases}$$

[58]This parameter defines how the phrase: 'much smaller than' should be interpreted. (See:[90], p. 110

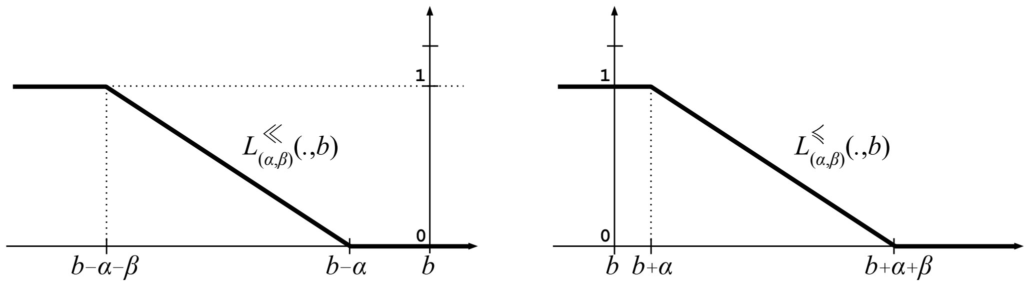


Figure 20: A visual presentation of relations $L^{\leq}_{(\alpha,\beta)}(.,b)$ and $L^{\preceq}_{(\alpha,\beta)}(.,b)$.

It appears that both relations are co-definable in the following sense:

$$L^{\preceq}_{(\alpha,\beta)}(a,b) = 1 - L^{\leq}_{(\alpha,\beta)}(a,b). \tag{40}$$

In addition, they show some computationally useful properties such as[59]:

$$L^{\preceq}_{(\alpha_1,\beta_1)} \bullet L^{\preceq}_{(\alpha_2,\beta_2)} = L^{\leq}_{\alpha_1+\alpha_2+\min \beta_1,\beta_2,\max(\beta_1,\beta_2)}. \tag{41}$$

$$L^{\preceq}_{(\alpha_1,\beta_1)} \bullet L^{\preceq}_{(\alpha_2,\beta_2)} = L^{\preceq}_{(\alpha_1+\alpha_2+\min\{\beta_1,\beta_2\},\max\{\beta_1,\beta_2\})}. \tag{42}$$

Unfortunately, not all of them are intuitive. Fortunately, each of them is detailed proved by inventors of this approach, what essentially supports a mathematical foundation of this approach.

Example 18 . *Consider two numbers: 20 and 25 and parameters $\alpha = 1$ and $\beta = 8$. In order to decide, how is a degree of occurrence of 20 long before 25, let us consider $L^{\leq}_{(2,8)}(20,25)$. Since $25 - 20 = 5 \leq 7$, thus $L^{\leq}_{(2,8)}(20,25) = 0$ according to the definition of $L^{\leq}_{(\alpha,\beta)}(a,b)$. Similarly, we can check that $L^{\preceq}_{(1,8)}(25,20) = \frac{20-25+1+8}{8} = \frac{1}{2}$. It exactly means that 25 cannot be considered as a number that occurs long before. Meanwhile, 25 may be considered to be before or at approximately the same time as 20 with a degree $\frac{1}{2}$.*

Since an objective of this presentation was to put forward a basis of a conceptual apparatus of the De Cock-Schockaert's position, we can interrupt it in this point.

0.17.3 Fuzzy Allen's Relations in Ohlbach's Integral-based Depiction

Let us move to the alternative approach to fuzzy Allen's relations in Ohlbach's depiction. It was elaborated in such works as: [85, 86]. Ohlbach begins with the simplest case of Allen's relations of the type: 'point-interval'. The next, he generalized them to Allen's relations for two fuzzy intervals. Finally, his efforts is oriented to a fuzzification of these relations. Although we preserve this conceptual chronology, we propose a slight modification of Ohlbach's proposal. Namely, we slightly extend Ohlbach's justification for the convolution representation of fuzzy Allen's relation.

Allen's relations of the type: points-fuzzy intervals

Assume that some point p, a fuzzy interval j and one of 13 Allen's relations, say R, are given. Observe that one can put the point-interval relation, $R_p j(x)$, which asserts that p is located in a position defined by R with respect to the interval j, etc.

[59]We omit an argument (a,b)

Example 19 *Taking a point p and an interval j, the relation $B_p(j)$ will asserts that p lies 'before' the interval j and $D_p(j)$ asserts that p is 'during' j.*

Let us preface further considerations by some useful observation that each point-interval relation (of Allen's sort) determines its corresponding function.

Colollary 1 *([85, 86])*[60] *A fuzzy point-interval relation R(t, i) is a function that maps a time point t and the interval i to a fuzzy value. Conversely, if i is a fuzzy interval and $R^{'}$ is a function, then R defined as follows:*

$$R(t,i) =^{def} R^{'}(i)(t) \tag{43}$$

is the corresponding fuzzy point-interval relation.

Allen's relations for two fuzzy intervals

Observe now that each such a point-interval Allen's relation may be extended to its corresponding interval-interval relation over a new fuzzy interval – as depicted in Fig. 22. For example, taking a time point t, an interval (not necessary a fuzzy one) j and a point-interval *R(t, j)* we can put

$$R(i,j) = i(t)R(t,\ j). \tag{44}$$

It allows us to write: $R(i,j)(t) = f^{i(x)}(t)R(j)(t)$, where $f^{i(x)}(t)$ is a function characterizing an interval i.

Example 20 *In this way one can specify after(i,j) = i(t)after(t, j), before(i,j) = i(t)before(t, j) and all interval-interval Allen's relations.*

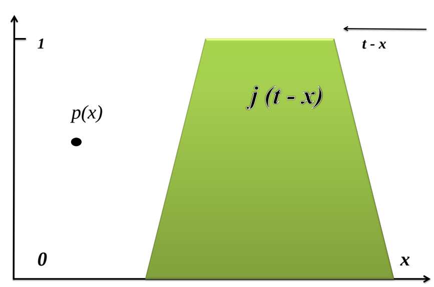


Figure 21: A point 'p' before a fuzzy interval. A visualization of a point-interval 'before' relation. (Interval j is considered here as dependent on the argument $t - x$ – according to the arrow direction on the diagram. The sense of this depiction will be explained in detail in chapter 2 of 'Contrubutions').

Fuzzy Allen's Relations for Fuzzy Intervals

In order to define fuzzy Allen's relations – due to Ohlbach's ideas – let us return to Allen's point-interval relations and consider, say 'before'-relation $B_p(j)(x)$, for a fuzzy interval j and i such that $p \in i$. Due to – [86], this relation may be rendered in terms of the so-called *extend* function E^+ and the complementation operator $N(E^+)$. This function 'behaves' as the functions depicted in Fig. 20 for $L^{\leq}$-relation. Namely, $N(E^+)$ decreases in the right neighborhood of a given point (b-α in Fig.20a) and it increases for arguments being far from it. Let us try to think about $B_p(j)(x)$ determined by $N(E^+)$ in terms of probability theory

[60] This corollary was introduced as a definition by Ohlbach in [85].

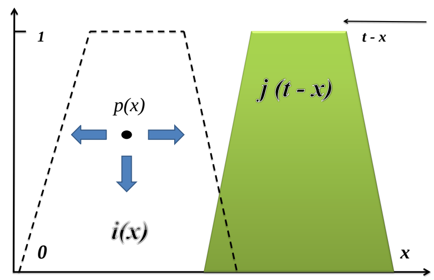


Figure 22: A point 'p' is blown up to the new fuzzy interval $i(x)$.

now.

Allen's point-interval relations in terms of probability theory. Therefore, assume that a probability space Ω of elementary events with a probability measure $P : \Omega \to [0, 1]$ are given. For a given fuzzy interval i we define points of its $\mathbb{R}$-support (See: Figure 18) as elements of Ω. Define also a *random variable* $X : \Omega \to \mathbb{R}$ such that $X(\omega) = X(p) = x \in \mathbb{R}$. In other words, we associate each point p of i-support to a single variable x of a real line. It enables to view $N(E^+)$ as a *distribution* function for i. In fact, $N(E^+)$ in Fig. 20 a) 'represents' a probability the event: $-\infty \leq X = x < b-\alpha$. Formally, $N(E^+)(x) = P(-\infty \leq X = x < b-\alpha))$.

Observe also that such a $N(E^+)$ is a continuous function and $N(E^+) < \infty$. Thus, there exists a function f_X to be called *probability density* – co-definable with $N(E^+)$ as follows:

$$N(E^+)(x) = \int_{-\infty}^{x} f_X(x)\mathrm{d}x. \tag{45}$$

Summing up, the (fuzzy) point-interval relation $B_p(j)$ in terms of $N(E^+)$ may be interpreted as a distribution for the second fuzzy interval i, that contains p's points. It remains to decide, what might represent the fuzzy interva-interval relation $B(i, j)$.

H-J. Ohlbach postulates to consider a unique expected value for X for in this role, although he did not render this postulate explicitely. In a general case, having a function $\phi : \mathbb{R} \to \mathbb{R}$, the expected value $E(\phi(X))$ is defined as:

$$E(\phi(X)) = \int_{-\infty}^{\infty} \phi(x)\mathrm{d}(F(x)), \text{ where } F(x) \text{ is } a\text{distribution}. \tag{46}$$

Fuzzy Allen's relations of the interval-interval type. It remains to specify this expected value of (38) in our case, or taking $N(E^+)(x) = F(x)$ (as a distribution) in (38). Therefore,

$$E(\phi(X)) = \int_{-\infty}^{\infty} \phi(x)\mathrm{d}(N(E^+)(x)). \tag{47}$$

Because of (37), we have

$$\int_{-\infty}^{\infty} \phi(x)\mathrm{d}N(E^+)(x) = \int_{-\infty}^{\infty} \phi(x)\mathrm{d}(\int_{-\infty}^{x} f_X(x)\mathrm{d}x) = \int_{-\infty}^{\infty} \phi(x) f_X(x)\mathrm{d}x, \tag{48}$$

thus

$$E(\phi(X)) = \int_{-\infty}^{\infty} \phi(x) f_X(x)\mathrm{d}x. \tag{49}$$

If put $\phi(x) = i(x)$ as a function characterizing the interval i, we can obtain the required form in our case:

$$E(i(x)) = \int_{-\infty}^{\infty} i(x)\widehat{B_p}(x)\mathrm{d}x, \tag{50}$$

where $\widehat{B_p(x)}$ denotes the fuzzy point-interval relation 'before' as a density function.

Example 21 . *If assume that*

$$F(x) = N(E^+) = \begin{cases} 0, & \text{for } x \le a, \\ \frac{x-a}{b-a}, & \text{for } a < x \le b, \\ 1, & \text{for} x > b, \end{cases} \quad \textit{then } f(x) = \begin{cases} 0, & \text{for } x < a, \\ \frac{1}{b-a}, & \text{for } a \le x \le b, \\ 1, & \text{for } x > b, \end{cases}$$

and $B(i,j) = E^j_{before}(i(x)) = \frac{1}{b-a}\int_{-\infty}^{\infty} i(x)\mathrm{d}x.$

In order to define fuzzy Allen's relations, let us begin with the point-interval relations $R_p(j)$. Even in a case of fuzzy intervals, $R_p(j)$ may 'behave' in two ways. These possibilities are depicted in Fig. 23 a), b) and c) for 'before$_p$(t)'-relation. In fact, $R_p(j)$ may be represented by a pure step function (a point c)) or by its fuzzified version – point a) and b).

Taxonomy of Fuzzy Allen's Relations in Ohlbach's depiction

We have just emphasized how fuzzy Allen's relations may be rendered in terms of integrals. In addition, a general form of them was elaborated for 'before'-relation. In this moment, a complete taxonomy of fuzzy Allen's relations in Ohlbach's depiction will be introduced – due to [85, 86].

Aberrative cases. However, its presentation requires some explanatory remarks. Note that the integral form (42) for 'before'-relation in terms of the expectation was elaborated for a fuzzy interval i with a left-infinite support $(-\infty, a]$. It was implicitly assumed that j has a compact support in a compact interval $[a, b] \subset \mathbb{R}$. Nevertheless, j may be aberrative. In fact, it may be empty or to have a right-infinite support $[a, \infty)$. If j is empty, then one cannot state that i is 'before' j, so the relation must yield 0 as its value in this case. If j has a right-infinite support $[a, \infty)$ the integral (42) may be infinite. For that reason, an intersection $|i \bigcap j|$ is considered instead of the whole i and j and $N(E^+)$ as restricted to it.

For brevity of presentation, the following convention is adopted. The fact that a fuzzy interval i has a right-infinite will be briefly announced by: $'i$ is $[a, \infty)$-type' and it will be written: $i = [a, \infty)$. Similarly, we will talk about intervals of $(-\infty, a]$-type.

Normalization. Obviously, all the integrals above, in particular (37) and (42), are finite. In particular, (37) holds, if and only if the integral on the right side of (37) is finite. It is warranted by the fact that $N(E^+)$ is assumed to be finite. Whereas finiteness of the integrals constitutes a sufficient condition from a purely mathematical point of view, it is unsatisfactory to consider (42) and similar conditions as the adequate representations for fuzzy Allen's relations. In fact, we expect that these integrals will take fuzzy values from $[0, 1]$, so they should be normalized.

Different methods of normalization is known and used. For 'before'-relation, the factors $|i|$ and $|i|^b_a$ defined as follows:

$$|i| = \int_{-\infty}^{\infty} i(x)\mathrm{d}x, \quad |i|^b_a = \int_a^b i(x)\mathrm{d}x. \tag{51}$$

To justify a choice of them, note that if we even move i to the area, where $\widehat{B(j)(x)} = 1$, then we get $\int_{-\infty}^{\infty} i(x)\widehat{B(j)}\mathrm{d}x = \int_{-\infty}^{\infty} i(x)1\mathrm{d}x = |i|$ and still $before(i, j) = 1$ (see: [85], p. 26.)

By contrast, $meet(i,j), start, finishes$ require the normalization factors of the type $N(i,j)$, so as dependent on both i and j. Ohlbach argues in [85], p. 26) for a choice of the following two factors of this type, as they admit also 1 as a possible value:

$$N(i,j) = \max_a \int_{-\infty}^{\infty} i(x-a)j(x)\mathrm{d}x, \quad N(i,j) = \min(|i|,|j|). \tag{52}$$

Taxonomy. We begin it with *before(i,j)*-relation. In order to underline a fuzzy character of each relation, they are equipped by the index *'Fuzzy'*.

1. Before^{Fuzzy} (see: [85, 86]): Assume that the point-interval relation 'before' *B(j)* is given. We can consider the following cases (dependently on finiteness of infiniteness of supports of fuzzy intervals to be considered):

- If i is $[a,\infty)$-type or $i=\emptyset$ or $j=\emptyset$, than one cannot state that i is 'before something', thus fuzzy *before(i,j)* must yield 0.
- If i is $(-\infty,a]$-type or the intersection of i and j is empty, then i always remains 'before' j, what justifies the value 1. (Figure 23 c)
- When i is $(-\infty,a]$-type, but j is bounded or of $[a,\infty)$-type, then – because $\int_{-\infty}^{\infty} i\widehat{B(j)}dx$ would be infinite – we take an intersection $i \cap_{min} j$ instead of the whole infinite i. (Figure 23 b)
- Otherwise, we begin from $B_p(j)$ to extend it over i , as described in (42) above (see: Figure 23 a).

It allows to define:

$$\textit{before(i,j)}^{Fuzzy} = \begin{cases} 0 & i=\emptyset \text{ or } i=[a,\infty) \text{ or } j=\emptyset, \\ 1 & i=(-\infty,a] \text{ or } i\cap_{\min} j=\emptyset, \\ \int_{-\infty}^{\infty} (i\cap_{\min} j\widehat{B(j)}dx/|i\cap_{\min} j| & i=(-\infty,a], j \text{ is bounded or } j=[a,\infty), \\ \int_{-\infty}^{\infty} i\widehat{B(j)}dx/|i| & \text{otherwise.} \end{cases}$$

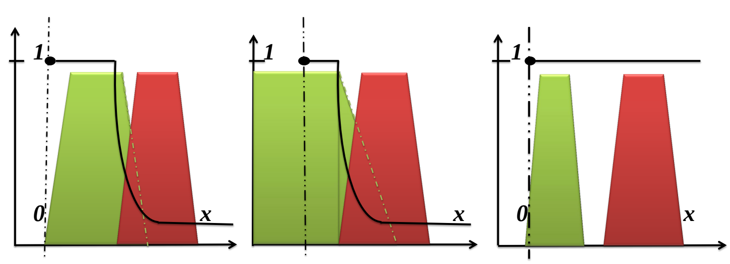


Figure 23: Fuzzy 'before'-relation of the interval-interval type dependently on 'size' of intervals. This relation may be represented by the dark lines.

2. Meet^{Fuzzy} (see: [85, 86]): Assume that some functions *Fin(i)* and *St(j)* that cut, somehow, final points of i-interval and start-points of j-interval (resp.) are given. Note:

- If i or j are empty, they cannot meet, so the relation yields 0.
- Similarly – if i is $[a, \infty)$-type and j is $(-\infty, a)$ and conversely, for the same fixed a,
- Otherwise, one can define this relation as a statement that the 'end' part *Fin(i)* of the interval i touches the initial part *St(i)* of the interval *j*. The factor *N(Fin(i), S(j))* normalizes this integral to be smaller than 1.

It leads to the following depiction.

$$
\boxed{\textit{meet(i,j)}^{Fuzzy} = \begin{cases} 0 & \text{if } i = \emptyset \text{ or } j = \emptyset \text{ or } i = [a, \infty) \text{ and } j = (-\infty, a), \\ & \text{or } j = [a, \infty) \text{ and } i = (-\infty, a) \\ \displaystyle\int_{-\infty}^{\infty} \frac{Fin(i)St(j)dx}{N(Fin(i), St(j))} & \text{otherwise.} \end{cases}}
$$

3. During^{Fuzzy} (see: [85, 86]): Assume that the point-interval 'during'-relation *D(j)* is given. Let us observe that

- If i is empty we put 1 as the relation holds for all possible points (i is always during j),
- If i is infinite and$j$ is empty, this relation does not hold at all – as above, so we put 0,
- If i is infinite, we should 'cut' this interval in the appropriate points: min of the first x-coordinates of i and j and max of the last x-coordinates of (kernel) of i and *j*.
- Otherwise, this relation "collects" all points of an i-interval such that each p of them satisfies a point-interval *during(p, j)*-relation *D(j)*, what may be rendered by the used integral.

In a consequence, we obtain the following definition of 'during'-relation. ($\widehat{D(j)}$ denotes a density-based representation of $D(j)$ – by analogy to $\widehat{B}(j)$ and $B(j)$.)

$$
\boxed{\textit{during(i,j)}^{Fuzzy} = \begin{cases} 1 & \text{if } i = \emptyset, \\ 0 & \text{if} j = \emptyset \text{ or } i \text{ is infinite,} \\ \displaystyle\int_{-\infty}^{\infty} i^{'}(x)\widehat{D(j)}\mathrm{d}x/|i^{'}| & \text{if } i \text{ is infinite,} \\ \displaystyle\int_{-\infty}^{\infty} i^{'}(x)\widehat{D(j)}\mathrm{d}x/|i^{'}| & \text{otherwise.} \end{cases}}
$$

where $i^{'} = cut_{min(i^{fK}, j^{fK}), max(i^{lK}, j^{lK})} i(x)$ (see: Figure 18, section 4.)

4. Equals^{Fuzzy} (see: [85, 86]): This relation holds iff an interval i is a subset of j and *vice versa*, hence:

$$
equals(i, j)^{Fuzzy} = during(i, j)^{Fuzzy} during(j, i)^{Fuzzy}. \tag{53}
$$

5. Start^{Fuzzy} (see: [85, 86]): Assume that the interval-interval during-relation *D(i,j)* and two functions $St_1(i)$ and $St_2(j)$, which cut start-sections of intervals i and j are given. Let us note that:

- If one of the interval is empty or one of them has a form $(-\infty, a]$, this relation does not hold, so we write 0.
- If one of them has a form $(-\infty, a]$, this relation does not hold as above, so we write 0.
- If both of them are infinite and of the form $(-\infty, a]$, than this relation can be represented by relation *D(i,j)* alone as the 'shorter' interval, say *i*, starts the 'longer' one, say *j*, so it also holds *during(i,j).*

- Otherwise, we will have two starting sections $St_1(i)$ and $St_2(j)$ of both intervals, thus *start(i,j)*$^{Fuzzy} = \int_{-\infty}^{\infty} \frac{St_1(i)St_2(i)dx}{N(St_1(i)St_2(j)}D(j,i)$ [61].

  Hence we get:

$$starts(i,j)^{Fuzzy} = \begin{cases} 0 & \text{if } i = \emptyset \vee j = \emptyset \ \vee i = (-\infty, a] \\ & \vee j = (-\infty, a], \\ D(i,j) & \text{if both } i,j \text{ are } (-\infty, a], \\ \int_{-\infty}^{\infty} \frac{St_1(i)St_2(j)dx}{N(St_1(i), St_2(j))} D(i,j) & \text{otherwise.} \end{cases}$$

5. Finish^{Fuzzy} (see [85, 86]): Let us begin with the following observations.

1. The reasoning in this relation case is similar like wrt the 'start'-relation.
2. The only difference consists in a considering final parts $(Fin_1(i), Fin_2(j))$ of both intervals i and j. For the same reason we should also exchange the intervals of $(-\infty, a]$ for $[a, \infty)$.

It leads to the following definition.

$$finishes(i,j)^{Fuzzy} = \begin{cases} 0 & \text{if} i = \emptyset \vee j = \emptyset \ or \ i \vee j = [a, \infty), \\ D(i,j) & \text{if both } i,j \text{ are } [a, \infty) - \text{type}, \\ \int_{-\infty}^{\infty} \frac{Fin_1(i)Fin_2(j)dx}{N(St_1(i), Fin_2(j))} D(i,j) & \text{otherwise.} \end{cases}$$

Since ideas of defining the 'overlap' are similar and, somehow, definable in terms of 'during'-relation – but a slightly more sophisticated formally – we omit its presentation. It may be found in [85, 86].

0.17.4 State of the Art and History of Research in this Area

Simple Temporal Problem was introduced by R. Dechter in [20] as a time-polynomially tractable restriction of the framework of Temporal Constraint Satisfaction Problems (TCSP). Due to – [20] – *The Simple Temporal Problems*(STPs) is a kind of the Constraints Satisfaction Problem, where a constraint between time-points X_i and X_j is represented in the constraint graph as an edge $X_j \to X_i$, labeled by a single interval $[a_{ij}, b_{ij}]$ that represents the constraint $a_{ij} \leq X_j - X_i \leq b_{ij}$. Meanwhile, a TCSP-consistency criterion in terms of a graph negative cyclic was proposed and broadly discussed in [69, 92, 70]. Some alternative approaches to represent temporal knowledge was proposed in terms of linear inequalities of Malik and Binford in [93] and Valdes-Perez [68] or in terms of Dean and McDermott's time map in [1].

In order to address the lack of expressiveness in standard STPs, some extended version of STP – the so-called *Simple Temporal Problem with Preferences* (STPP) was proposed by L. Khatib in [17]. A lack of flexibility in execution of standard STPs was a motivation factor to introduce the so-called *Simple Temporal Problem under Uncertainty* (STPU) in [18]. In order to capture both the possible situations of acting with preferences and under uncertainty, a new STP-specification in a form of the *Simple Temporal Problem with Preferences under Uncertainty* (STPPU) was proposed in [19].

Chronologically, the first (rather philosophically motivated) approach to the logical representation of preferences was proposed by von Wright in [82] by means of an interval logic with a kind of a chop-star operator. The slightly younger approach to their logical modeling was expressed in terms of probability theory by Jeffrey in [94]. Independently of a lack of a consensus between researchers concerning the modeling

[61] Note that in a crisp case this relation may be given by a product $\{x = a\}\times$ *D(j)*, where $x = a$ is a line containing the initial points of both crisps intervals and *D(j)* is some point-interval relation during *D(j)*.

of preferences, one can indicate some typical trends in this area. Namely, preferences are often considered in a point-wise manner in game-theoretic terms like in [80, 81] and represented by a total or linear order relation or, in a weaker sense, as a reflexive and transitive relation as in [83].

Contributions

Chapter 1

CONTRIBUTIONS: Temporal Planning Problems Approaches: Critique and Taxonomy

Inhaltsangabe

1.1 Temporal Planning with Fuzzy Constraints and Prefereces: Analysis of Selected Problems . . . 70

1.2 The Temporal Travelling Salesman Problem . . . 74

1.2.1 Informal Defining of TTSP . . . 74

1.2.2 Formal Definition of TTSP. . . . 75

1.3 Multi-Agent Schedule-Planning Problem . . . 79

1.3.1 A Multi-Agent Schedule-Planning Problem – a More Practical Depiction . . . 79

1.3.2 General Formulation of a Multi-Agent Schedule-Planning Problem . . . 81

1.3.3 Types of Temporal Constraints of PM-AS-PP . . . 82

1.4 Multi-Agent Schedule-Planning Problem *versus* Temporal Traveling Salesman Problem – Their Modeling and Representations . . . 85

1.1 Temporal Planning with Fuzzy Constraints and Prefereces: Analysis of Selected Problems

The last section 'Introduction' presented a broad *spectrum* of different approaches to temporal reasoning with constraints and preferences. In particular, a machinery of classical planning and its temporal extension was discussed in section 1. The perspective of two different planning paradigms:

PA1 *planning as graph searching*, and

PA2 *planning as satisfiability.*

was considered. In second part of 'Introduction', a special focus was paid to the notion of temporal constraints in in a context of *Simple Temporal Problem* and its admissible extensions such as: *Simple Temporal Planning under Uncertainty* and *Simple Temporal Planning under Uncertainty and Preferences.* In addition, the concept of preferences as such were analyzed in detail. This part was complemented by a description of two approaches to fuzzy Allen's relations in a more *algebraic depiction* of De Cock-Schockaert's school and a more *analytic, integral depiction* of H-J. Ohlbach.

It appears that temporal planning with fuzzy constraints and preferences is involved in some (both inherent and 'external) difficulties. Technically speaking, a conjunction of the following shortcomings is observed:

D1 The area of research on both temporal planning and temporal constraints is properly not subject-specified.

D2 There exists an unwanted dualism: quantitative-qualitative w. r. t. the temporal constraints,

D3 Schockaert's approach is algebraically well-founded, but technically inconvenient and partially contra-intuitive as based on a calculus of combined infima and suprema, minima and maxima. In contrast, Ohlbach's depiction is intuitive and based on a stronger 'analytic' machinery, but it still requires a stronger mathematical grounding.

D4 Finally, a knowledge about a real 'area of use' of planning methods, such as: STRIPS or Davis-Putnam procedure, is restricted and unclear.

Let us briefly discuss each of the difficulties. Since teh difficulty D1 forms a special focus area of this chapter, we avoid it now for a cost of a brief describing of D2, D3 and D4.

D2 and D3, or the unwanted dualism: 'quantitative-qualitative' and shortcomings of DeCock-Schockaert's approach

Let us recall that temporal constraints are usually divided into two subclasses:

- qualitative constraints and,
- quantitative ones.

The first subclass covers (at least): Point Algebra [56] and Allen's algebra of intervals [57, 59]. The subclass of quantitative temporal constraints covers Temporal Constraint Problem (TCP) [20] and its specifications in terms of *Simple Temporal Problem* (STP) and its extensions STPU, STPPU [20, 17, 18, 19]. The main discrepancies between these two types of constraints may be briefly listed as depicted here.

Qualitative temporal constraints	Quantitative temporal constraints
Allen's Algebra + Point Algebra	TCP and its specifications: STP, STPU, STPPU
Intervals *without arithmetic specification.* **Example**: $I_1, I_2, \ldots$	Intervals *with arithmetically specified* start and end-points. **Example**: $[a_1, b_1]$, $[1, 3]$, ...
Operations on intervals give intervals (*without arithmetic specification*). **Example**: I *before*$\bullet$*before* $J = I$ *before* J	Operations on intervals give intervals with arithmetically specified start- and end-points **Example**: $[1,3]\bigcap[2,4] = [2,3]$.
Results of operations on temporal constraints may be ambiguous. **Example:** $finish \bullet overlap$ may give *overlap*, *start*, but equally *during* as a result.	Results of operations are unambiguous, but techniques of quantitative reasoning are not easily adaptable. **Example:** $[1,3]finish \bullet overlap[2,3]$ gives $[1,3]overlap[2,3]$ as the only admissible answer. However, compositions of Allen's relations do not coincide well with the STP-based conceptualization.

A remedy for this dualism might be a proposal of De Cock-Schockaert's school – as a combination of both types of temporal constraints is considered. However, this approach is partially computable inconvenient and contra-intuitive as based on a sophisticated combination of infima, suprema, minima and maxima. The next, norms of max and min form a very unique type of admissible norms, so their use immediately restricts a field of reasoning. In addition, it is not clear how this proposal works in the conceptual framework determined by STP's, STPU's and STPPU's.

A new remedy in terms of Ohlbach's integral-based, theoretically and practically supported, will be proposed in chapter 1 of 'Contribution'.

D4, or a restricted knowledge about a real 'area of use' of planning methods

Although a portion of knowledge about limits of utility of (classical) planning methods may be found in different places (for example, see: [48], pp. 299-300 for STRIPS), our knowledge about fuzzy temporal and preferential extensions of them is restricted. Let us briefly discuss this issue with respect to the STRIPS-algorithm. One one hand, one could argue that a lack of references to temporal constraints does not constitute any internal shortcoming of STRIPS-algorithm. On the other hand, this lack might be seen as a real difficulty in the perspective of the question: 'How this method is useful in temporal reasoning?'.

In fact, it is not clear *how deep* the STRIPS-algorithm is suitable to be temporally extended[1]. Each attempt of an answer to that question meets some ambiguity. In order to indicate it, let us return to the following fragment of STRIPS-algorithm, which describes a choice of actions from a given set $\mathcal{A}$ of them (As earlier, s_0 is a distinguished initial state of a given planning domain S, $g \in S$ is a goal and π denotes a plan):

Fragment of STRIPS:

$s \leftarrow s_0$
$\pi \leftarrow$ the empty plan
if s satisfies g then return π

applicable $\leftarrow$ $\{a|a$ is a ground instance of an operator in $\mathcal{A}$ and `precond(a)` is satisfied in $s\}$.

[1]That means: 'extended to a version that contains a new component with temporal constraints'.

Assume now that C denotes a set of temporal constraints imposed on $\mathcal{A}$. The ambiguity consists now in a possibility to extend STRIPS-algorithm by temporal constraints from C (at least) in the following ways:

Method1 : only one action a from $\mathcal{A}$ should be chosen in order to check whether a satisfies constraints from C,

Method2 : a pair of actions, say (a_1, a_2), should chosen in order to comparatively check which action (possible both) respects constraints from C,

Method3 : n-tuples of actions from $\mathcal{A}$ should chosen in order to comparatively check which action (possible all) respects constraints from C.

To make the matter worse, each method might be naturally associated to the following open questions:

1. Should these actions be chosen deterministically or non-deterministically?

2. If we admit a hierarchy of actions satisfying constraints from C, how to reconcile/combine this hierarchy with hierarchy of actions determined by preferences?

A fuzzy temporal and preferential extension sof STRIPS will be proposed in Chapter 2 of 'Contribution' as a unique remedy for this difficulty, just signalized. The similar proposal will be put forward for Davis-Putnam procedure.

D1, or a lack of a subject-specification of temporal planning

Let us, slightly non-chronologically, return to the first of the difficulties, D1. It consists in a lack of a subject-specification of temporal planning domains; in a lack of the appropriate problems. This difficulty is also reflected in a discrepancy between a representation of planning paradigms and a representation of temporal constraints.

Whereas the approaches to representation of temporal constraints are rendered in terms of some temporal problems to be solved (such as: STP, STPU or STPPU), temporal and classical planning are usually introduced in a more methodological way. Indeed, all planning paradigms (planning as graph-search, planning as satisfiability, etc.) should be more seen as different methodologies equipped with some unique tools (such as STRIPS-algorithm, Davis-Putnam procedure, etc.) than a class of domain-specified planning problems.

This discrepancy must be overcome and introduces a need of a subject-specification of temporal planning. This subject-specification is also worth exploring for (at least) three reasons:

Reason1 : Temporal constraints and preferences (especially) are essentially dependent on the situation contexts and usually cannot be considered *in abstracto*, but they should be rather relativised to the appropriate contexts[2] and to practical problems to solve,

Reason2 : Planning requires the same reference to practical situations, contexts and problems,

Reason3 : Finally, majority of potential planning problems may be divided into the appropriate classes of problems.

A taxonomy for the subject-specification of temporal planning will be elaborated in the current chapter. We preface its presentation by introducing criteria that must be satisfied by such a taxonomy.

[2]Author of this work is aware of the whole pragmatic sense of this notion, which cannot be, however, defined in details as it can be considered in a common sense for a use of this analysis.

Towards an exhaustive taxonomy of temporal reasoning problems.

It is clear that the appropriate subject-specification (taxonomy) of temporal planning problems should respect (at least) the following criteria.

Criterion 1 : This taxonomy should be possibly exhaustive. Exhaustiveness of such a taxonomy means that the distinguished classes of temporal planning problems exhaust (or almost exhaust) the whole *spectrum* of planning problems.

Criterion 2 : It should be possibly realistic. This criterion suggests to consider planning such as it is usually considered in realistic problems as, for example, together integrated with scheduling.

Criterion 3 : This taxonomy should cover mutually complementary and disjoint classes of temporal problems.

Taking into account these guidelines, the following two paradigmatic and complementary classes of problems will be proposed in the framework of the introduced taxonomy:

Class1 Temporally extended Traveling Salesman Problem,

Class2 Multi-Agent Schedule-Planning Problem.

A complementarity of these problems is illustrated by the following juxtaposition.

Temporal Traveling Salesman Problem	**Multi-Agent Schedule-Planning Problem**
temporal planning	planning + scheduling
a single agent (salesman) case	multi-agent case
temporal constraints (rather) in a quantitative form	temporal constraints (rather) in a quantitative form

Nevertheless, both paradigmatic problems will be conceived in such a way to reconcile the quantitative and the qualitative approach to modeling temporal constraints. These problems will be described in detail in next chapter.

Motivation for introducing this taxonomy.

At the end of this section, let us preface the question of a role of this taxonomy, just introduced. In fact, it seems that its exhaustiveness – reflected by mutual complementary of these class of problems – does not constitute any sufficient argument to introduce them. In addition, both *Traveling Salesman Problem* (TSP) and *Nurse Job Scheduling Problem* (NJSP) – as a conceptual basis problem for *Multi-Agent Schedule-Planning Problem* are well-known optimization and scheduling problems. To make the matter worse, these problems and their subproblems may be formulated in different variants and solved in different ways.

Although an existence of such a variety of different variants of TSP and NJSP might be, somehow, problematic for researchers aimed at finding some new smarter, quicker or more universal solutions of these problems, this fact does not constitute any real difficulty for further analyses. In fact, a *role* of these problems (more precisely: their temporal and preferential extensions) is different in a conceptual framework of the thesis and may be specified as follows:

Role1 They constitute a basis for their further temporal and preferential extensions.

Role2 They constitute a (potentially enormous) *reservoir* of temporal constraints and preferences – modeled and represented in different forms.

Role3 They allow us to elucidate computational and programming-wise aspects of modeling these entities of temporal planning.

Role4 They, finally, deliver some situation contexts for exemplification of problems discussed in 'Contributions'.

It seems that this multi-dimensional role justifies the proposed taxonomy as based on these two problems.

1.2 The Temporal Travelling Salesman Problem

In this section a basic, generic temporal planning problem to be referred to as *The Temporal Traveling Salesman Problem* (or TTSP, for short) will be introduced and described in detail. The TTSP is an extension of the standard Traveling Salesman Problem (TSP) – a classical optimization problem[3] – over time component. It is in fact composed as a specific mixture of *travel planning* and *task performing* subject to temporal constraints. The formal definition of TTSP and its solution will be prefaced by some informal definition of this problem [4].

1.2.1 Informal Defining of TTSP

The *Traveling Salesman* is an agent, capable of traveling (e.g. from town to town) and performing tasks (e.g. deliver goods). In the basic statement there is exactly one such agent and many delivery points located at different nodes of a graph. Each delivery task is assumed to have some temporal extent – i.e. has some assigned length in time. For example, to unload a truck takes a certain interval of time. Moreover, there is given a network of nodes (locations, towns) represented by a graph. Each vertex is assigned a delivery task to be accomplished at the node. Each such task (e.g. a course) is identified by a unique name, and a start time and an end time. Both the start time and the end time are referring to the calendar/clock time and are rigid. For example, they can refer to the working hours of a delivery point. In order to accomplish a task, the agent must:

* arrive at the node *at or after* the start time, and
* stay at the node for a period necessary to complete the delivery task,
* leave it *before or at* the end time.

In a generalized version, there may be other requirements, both of hard and soft nature (e.g. Allen's relations, Fuzzy Temporal Relations).

Further, any travel from a node to another, directly connected node takes some amount of time. In the basic formulation, this is just an interval (a *floating* one). This means that the travel can be started at any time (e.g. the agent uses its own car). Now, a semi-informal depiction of the *The Temporal Traveling*

[3] Traveling Salesman Problem has a long chronology. It is commonly assumed that it was primary formulated c.a. 1800 year by the British mathematician W.R. Hamilton in terms of Hamiltonian cycles. Anyhow, in a mathematical depiction TSP was formulated in 1932 by K. Menger in [11] and developed by I. Heller in [12] and solved by M.M. Flood in [10] and in many other places. Today TSP is alternatively defined in terms of Hamiltonian cycles – due to ideas of this mathematician as follows: *Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?.*

[4] *Traveling Salesman Problem* (TSP) – is known to be a paradigmatic example of NP-complete problem of the optimization theory (in its basis formulation). It was shown by C. Papadimitriou in [95] that even if a distances between cities in a basis formulation of TSP are determined by euclidean metrics, the TSP remains NP-complete. In addition, even though this problem is computationally difficult, a large number of heuristics and exact methods to solve it are known. They allows us to solve this problem completely and to approximate its solution in a reasonable way even though this problem is formulated for such enormous numbers of cities as millions.

Salesman Problem is as follows:

Consider an agent – called later the *Salesman* – which moves over a given graph G such the following conditions are satisfied.

1. The agent starts from some predefined initial node.
2. He has to travel through all specified nodes, finishing at some arbitrary node (this may be the initial node, as in the classical version, or any other arbitrarily defined one).
3. At any node, the agent must accomplish the delivery task assigned to that node.
4. He must act according to the temporal constraints - the task must be accomplished within the operation interval.

The *solution* of the problem is given by a sequence of nodes to be visited such that:

* the sequence start with the predefined initial node,
* it covers all the required nodes (tasks),
* it ends at the required node (e.g. the same as the start node),
* at each node the temporal constraints are satisfied (Node Temporal Constraints),
* there is enough time to travel between any successive nodes (Travel Temporal Constraints).

Usually, it is assumed that the solution of TSP must be optimal (time or costs should be possibly minimal). It follows from the fact that TSP originally forms an optimization problem, as earlier mentioned. For the same reason, the solution of TTSP as a temporal extension of TSP should be also optimal in the same sense if consider TTSP as an optimization problem. However, we avoid this restriction as we are not interested in the optimization aspects of TTSP, but in a modeling and formal representation of this problem.

are not interested in such a depiction as we

1.2.2 Formal Definition of TTSP.

In order to formally define the TTSP one needs the following components:

* a graph representing the delivery points (nodes) and links among them,
* definition of temporal constraints at each node (Node Temporal Constraints or Task Temporal Constraints),
* definition of global constraints w.r.t. the travel,
* definition of global temporal constraints or performance index (e.g. the total time to cover or delivery requirements).

The definition of TTSP is presented below.

Definition 15 ***The Temporal Traveling Salesman Problem*** *(TTSP) is defined as n-tuple*

$$TTSP = (G, \gamma, s, e, \delta) \tag{1.1}$$

where:

- *G is the graph representing the problem, $G = (V, E)$, V is the set of vertices and E is the set of edges,*
- *γ is the function assigning time to edges, $\gamma\colon E \to \Delta$, where Δ is the set of admissible (floating) intervals of time (for intuition, any edge is assigned an interval of time necessary to go along it),*
- *s and e are the functions defining start and end of operation of any node $v \in V$ (e.g. $s(v)$, $e(v)$), or the agent a at the node (e.g. $s(a, v)$, $e(a, v)$),*
- *δ is such a mapping defining admissible or necessary time for operation at any node $v \in V$ that: $\delta : v \mapsto \Delta t \subset [s(v), e(v)]$*

The last condition for δ expresses the fact that a task is performed in v in some internal time Δt *within* the interval $[s(v), e(v)]$ associated to v.

Example 22 *Consider a salesman K and delivering packages between between n cities (with temporally measured distances between them) – denoted by $C_1, C_2, \ldots, C_n$ and $\gamma(C_i \to C_j)$ respectively, for $i, j \in 1, 2 \ldots, n$. To all cities $C_1, C_2 \ldots, C_n$ a temporal restriction for K's operation in each of them is associated and given by intervals $I_1, I_2, \ldots, I_2$ (resp.). In addition, to each city C_i (for i as above) the start and end points s_i, e_i for K's operation are associated.*

Assume, as usual, that K is visiting all cities (from a given list) in such a way to find the shortest possible route that visits each city exactly once and leads to the origin city. Assume that K begins with a city C_1 and a temporal distance between C_1 and C_2 amounts 3 hours and temporal distances between each pair of other cities is bounded by M hours. Assume in addition that a package, say A, must be delivered in C_2 in some temporal interval I_2. It is known that K will perform the required task in C_2 in the time interval $(1:00, 2:00) \subset I_2 = 5h$ or between $(3:00, 4:00) \subset I_2 = 5h$.

Due to above definition – TTSP for K (symbolically $TTSP^K$) may be formally given as follows:

$$TTSP^K = (G^K, \gamma^K, s^K, e^K, \delta^K), \tag{1.2}$$

where:

- $G^K = \Big(C_1, C_2, \ldots, C_n, \{C_i \to C_j\} : i \neq j,\, i, j \in \{1, 2 \ldots, n\}\Big)$,
- $\gamma^K(C_1 \to C_2) = 3h, \gamma^K(C_i \to C_j) \leq M, i \neq j,\, i, j \in \{1, 2 \ldots, n\}$,
- $\delta^K(C_2) = I_2 = 5h$,
- $(1:00, 2:00) \subset I_2 \vee (3:00, 4:00) \subset I_2$, $(s_i^K, e_i^K) \subset I_i$, $i \in \{1, 2 \ldots, n\}$.

Example 23 *Consider a professor X involved in some temporally restricted activities in German cities. Taking into account his weekly activity timetable, we can distinguish the following activities:*

1. *One day Prof. X carries out 3 lectures at LMU in Munich between 8:00 and 13:15.*
2. *The next day he participates in the scientific conference (between 10:00 and 15:00) at the Humbold's University of Berlin with a 30-min talk (14:15 - 14:45).*

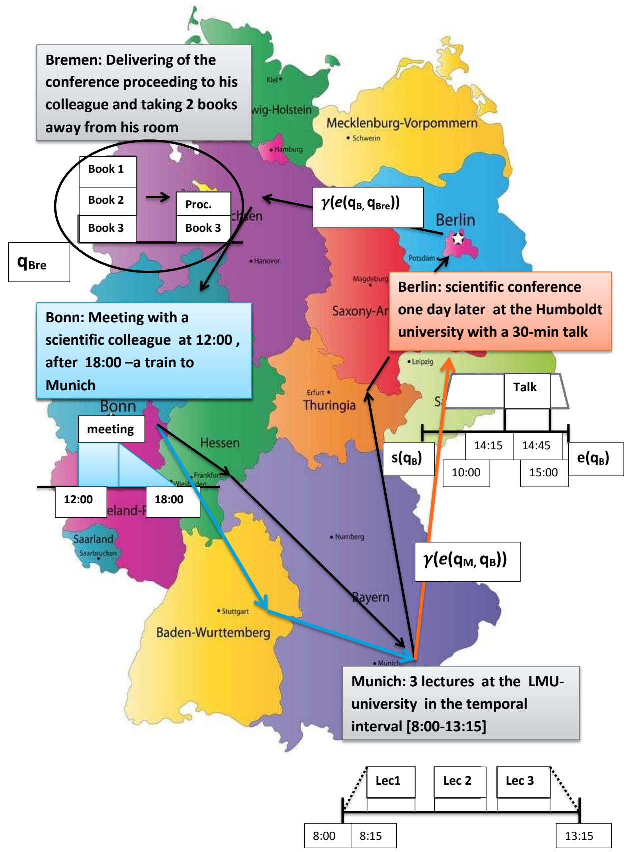


Figure 1.1: The visualization of the TTSP for Visiting Professor in terms of the proposed conceptualization.

3. *During the third day of his activity, he meets his colleague in Bremen, delivering him a conference proceeding from Berlin and he visits his office at the university taking 2 books away from his room,*

4. *In the 4th day, he has a scientific appointment at 12:00 in order to manage to take a night train to Munich.*

This activity is also involved in some preferences with respect to the choice of the appropriate communication way between cities – as depicted on in Fig.1.1.

Assume that D denotes a day $D, D+1$ – a one day later, $D+2$ – two day later. Adopt also the convention that: X:00D denotes X:00h on D-day, etc. Thus, TTSP may be formally given as follows:

$$TTSP^{Prof} = (G^{Prof}, \gamma^{Prof}, s^{Prof}, e^{Prof}, \delta^{Prof}), \tag{1.3}$$

where:

- $G^{Prof} = \Big(V^{Prof} = \text{Munich}, \text{Berlin}, \text{Bremen}, \text{Bonn},\ E = \{\text{Munich} \to \text{Berlin}, \text{Berlin} \to \text{Bremen}, \text{Bremen} \to \text{Bonn}, \text{Bonn} \to \text{Munich}\}\Big),$

- γ^{Prof}(Munich $\rightarrow$ Berlin), γ^{Prof}(Berlin $\rightarrow$ Bremen), γ^{Prof}(Bremen $\rightarrow$ Bonn), γ^{Prof}(Bonn $\rightarrow$ Munich),
- δ^{Prof}(Munich) $= (8:15D, 13:00D)$, δ^{Prof}(Berlin) $= (14:15(D{+}1), 14:45(D{+}1))$, δ^{Prof}(Bremen) $=$ $(14:15(D+2), 14:45(D+2))$,
  δ^{Prof}(Bonn) $= (12:00(D+3), 12:00+\epsilon)(D+3))$,

- (s^{Prof}(Munich), e^{Prof}(Munich)) $= (8:00, 13:15)$,
  (s^{Prof}(Berlin), e^{Prof}(Berlin)) $= (10:00, 15:00)$,
  (s^{Prof}(Bremen), e^{Prof}(Bremen)) $= (00:00, 24:00)$,
  (s^{Prof}(Bonn), e^{Prof}(Bonn)) $= (00:00, 18:00+\epsilon)$.

In order to solve the given TTSP's one has to find the appropriate routes over the graph satisfying all the constraints (operational, temporal). Such a rout defines an *admissible solution*.

Definition 16 ***(Solution of TTSP)*** *Given a TTSP*

$$TP = (G, \gamma, s, e, \delta)$$

defined as by def.15, its solution is defined as a sequence of nodes $Q = (q_1, q_2, \ldots, q_n)$ such that:

> *each $q_i \in V$, $q_i \neq q_j$ for $i \neq j$, and any $v \in V$ appears in Q,*
>
> *the following Node Temporal Constraints are satisfied:*
>
> $$s(a, q_i) \geq s(q_i)$$
>
> *(the agent can start delivery task at node q_i only after beginning of the operation of that node),*
>
> $$e(a, q_i) \leq e(q_i)$$
>
> *(the agent must end the delivery task at node q_i at or before the end of operation time of that node),*

where the agent operation time are defined recursively, as follows:

- *the starting time is defined as the start of operation time of node q_1:*

$$s(a, q_1) = s(q_1)$$

- *the end of agent operation time at node q_i is the sum of its start time and the necessary operation time:*

$$e(a, q_i) = s(a, q_i) + \delta(a, q_i)$$

- *the start of agent operation time at node q_i for $i > 1$ takes into account the end operation time at the former node and the travel time:*

$$s(a, q_i) = e(a, q_{i-1}) + \gamma(e(q_{i-1}, q_i))$$

Obviously, the following condition must be satisfied to enable an admissible solution:

$$s(a, q_i) + \delta(a, q_i) \leq e(q_i)$$

i.e. the agent must be able to complete the task at any node before the closing time for that node. In case of looking for *optimal solution*, an admissible solution assuring minimal total operation time should be searched for.

1.3 Multi-Agent Schedule-Planning Problem

In recent chapter, the *Traveling Salesman Problem* was introduced as a paradigmatic problem for the whole class of a one-agent planning problem. This problem was introduced not only informally, but also formally. This chapter will be devoted to the class of Multi-Agent Schedule-Planning Problem (MASPP). At first, this class will be described in a more general way. Secondly, it will be introduced in a (promisingly) applicable way. This way of MASPP presentation seems to better correspond to a nature of this problem and it is partially determined by earlier literature-based depictions of it.

The problem of this class can be briefly specified as problems with:

* an inventing the action sequence in order to perform a goal (a planning component),

* association actions to agents that could be performed by them due to their skills (a scheduling component).

In the next step, this class will be specified in details by considering its paradigmatic exemplification – the *Multi-Agent Schedule-Planning Problem* in order to indicate a couple of temporal constraints – typical for problems of this class.

1.3.1 A Multi-Agent Schedule-Planning Problem – a More Practical Depiction

The proposed Multi-Agent Schedule-Planning Problem may be seen as a (relatively far) generalization of problems – earlier considered in the specialist literature such as: the so-called Nurse Job Scheduling Problem (NJSP)[5]. These works consider this problem as optimization problem of scheduling without planning components and preferences.

It appears that NJSP – in formulations known in a subject literature – formed a stimulating problem for operational research, which also supported a broad development of constraints logic programming methodology – especially exploring by Nottingham's school. All these fact are, somehow, reflected in such works as: [13, 14, 15, 16]. NJSP was depicted from different perspectives and different constraints were associated to situations described by NJSP. In a consequence, this problem as it should be rather seen a basis *reservoir* of possible more specified formulations. Unfortunately, no of earlier depictions seems to be completely satisfying in the light of current analyses leading to elaborating of new model and and ways of representation of NJSP and TSP. For example, Ernst's approach in [15] is mathematically general, but refer to simplified situations. By contrast, Lepegue's approach from [97] is pretty detailed, but a formalization of possible temporal constraints for NJSP seems to be too excessive and it is not intuitive.

Taking into account these known formulations of NJSP – we elaborate a new compromise depiction of

A the Multi-Agent Schedulo-Planning Problem,

B the Preferential Multi-Agent Schedule-Planning Problem.

Our formulation will be an effect of a compromise between diversity of possible aspects of the problem and a simplicity in a presentation of them.

[5]This problem is to be also known as *Nurse Rostering Problem* – see for example: [13, 16, 96]

Multi-Agent Schedule-Planning Problem (M-AS-PP). Consider a factory with n-agents working in a rhythm of the day-night shifts: D–the day shift and N–the night shift. Generally – each day at least one person must work at the day shift and at least one – at the night one. Each agent has "working shifts" and "free shifts". These general rules of scheduling is constraints in the following way.

HC1 The charm of the shift organization should be fair: each agent must to have equally: 2 day-shifts and 2 night-shifts.

HC2 Each agent can be associated to at most one shift,

HC3 Some shifts are prohibited for agents,

HC4 Length of the shifts sequences associated to each agent is restricted,

HC5 Quantity of the shifts in a scheduling period is restricted,

HC6 Quantity of the shifts per a day is restricted.

The M-AS-PP consists in a construction of a scheduling diagram, which respects all these constraints[a]..

[a]In [97] some other hard constraints are considered – such as: 'The assignment of compulsory administrative tasks must be respected' or 'Employees must finish a task before starting another one'. They will be omitted as they seem to be a kind of meta-rules for a scheduling process in author's opinion.

As one can easily see, a couple of the so-called hard constraints in the above depiction of M-AS-PP was indicated, HC1-HC6.

Generally, *Hard constraints* are specified as these constraints that *should be violated* in a scheduling task. They ensure a feasibility of the scheduling task. *The soft constraints* may not be satisfied, but a degree of their satisfaction is a measure how good is a scheduling plan. To make requirements with respect to the hard constraints more liberal, we use the so-called relaxation, i.e. a weakening of the strong constraints. We often use this solution, when satisfaction of all hard constraints leads to an inconsistency.

A further relaxaction of requirements or expectations allows us to consider the next category of preferences. The main nature of preferences as relations was discussed in 'Intoduction 2'. In this section, we are interested in another sense of this concept. The preferences are wishes or expectations of an agent, for example, with respect to the action execution ora their sequencing.

Both the soft constraints and preferences are admitted in the *Preferential Multi-Agent Schedule-Planning Problem (PM-AS-PP)*. Generally, all hard constraints of *Multi-Agent Schedule-Planning Problem* are preserved in this new problem. In fact, it forms a kind of an extension of teh initial M-AS-PP towards soft constraints and preferences. They are also defined in the same context as HC's.

Preferential Multi-Agent Schedule-Planning Problem (PM-AS-PP) Consider a factory with n-agents working in a rhythm of the day-night shifts: D–the day shift and N–the night shift. Generally – each day at least one person must work at the day shift and at least one – at the night one. Each agent has "working shifts" and "free shifts". These general rules of scheduling is constraints in the following way.

HC1 The charm of the shift organization should be fair: each agent must to have equally: 2 day-shifts and 2 night-shifts.

HC2 Each agent can be associated to at most one shift,

HC3 Some shifts are prohibited for agents,

HC4 Length of the shifts sequences associated to each agent is restricted,

HC5 Quantity of the shifts in a scheduling period is restricted,

HC6 Quantity of the shifts per a day is restricted.

Assuming also an agent $n_k \in N$ and the chosen (real) parameters m, M and α Different soft constraints and preferences of a general form are also considered in the scheduling procedure.

SC7 A preferential quantity of shifts in a scheduling period is established,

SC8 A preferential scheduling charm's covering by shifts in a scheduling period is established,

SC9 A preferential lenght of the shifts sequence associated to an agent is fixed,

Pref1 A number of actions (preferred by an agent n_k) to be associated to its schedule is greater than m and smaller than M,

Pref2 An agent n_k prefers to perform an action a with a degree α[a].

The M-AS-PP consists in a construction of a scheduling diagram, which respects all these constraints.

[a]The parameters may be chosen arbitrarily, but they are fixed in the whole M-AS-PP problem. In some particular cases, the choice of them may be restricted according to the appropriate criteria or other restrictions.

It easy to observe that SC's and preferences have a common denominator: something is preferred. However, SC's express a global preference of the whole problem, but preferences render particular preferences of a single agent. Because of this distinction, we are willing to consider the global external preferences 'more seriously' as soft constraints.

1.3.2 General Formulation of a Multi-Agent Schedule-Planning Problem

Let us consider a generic temporal multi-agent task scheduling problem. Roughly speaking, a set of agents, each of them possessing specific skills, is to be assigned some temporal tasks to be completed. Each agent can accept only tasks consistent with its skills. Execution of tasks should be performed according to predefined partial order relation. Further auxiliary constraints (e.g. Allen's type constraints for execution periods of certain actions) or extensions (e.g. parallel execution of actions by a single agent) are possible. Below, a generic, simple formalization of this problem is put forward.

Consider a set $\mathbb{A} = \{A_1, A_2, \ldots A_n\}$ of n agents. Each agent can possess one or more skills. Let $\mathbb{S} = \{S_1, S_2, \ldots S_k\}$ denote the set of predefined skills. Assume σ is the function defining a two-valued measure for all the skills of any agent; so σ is defined as:

$$\sigma\colon (\mathbb{A}, 2^{\mathbb{S}}) \mapsto \{0, 1\}.$$

For practical reasons, it is convenient to represent this function in a tabular (matrix) form as follows:

	S_1	S_2	$\ldots$	S_k
A_1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\ldots$	$\sigma_{1,k}$
A_2	$\sigma_{2,1}$	$\sigma_{2,2}$	$\ldots$	$\sigma_{2,k}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_n	$\sigma_{n,1}$	$\sigma_{n,2}$	$\ldots$	$\sigma_{n,k}$

where $\sigma_{i,j} = \begin{cases} 1, & \text{if } S_j \in \sigma(A_i) \text{ (the } i\text{−th agent possess the } j\text{−th skill)} \\ 0, & \text{otherwise.} \end{cases}$

Similarly, consider a set $\mathbb{T} = \{T_1, T_2, \ldots, T_m\}$ of tasks to be executed. Each tasks, in order to be executable by an agent, requires some specific skills. Assume θ is the function defining all the skills required to execute a specific task; so θ is defined as:

$$\theta\colon (\mathbb{T}, 2^{\mathbb{S}}) \mapsto \{0, 1\}.$$

Again, for practical reasons it is convenient to represent this function in a tabular (matrix) form as follows:

	S_1	S_2	$\ldots$	S_k
T_1	$\theta_{1,1}$	$\theta_{1,2}$	$\ldots$	$\theta_{1,k}$
T_2	$\theta_{2,1}$	$\theta_{2,2}$	$\ldots$	$\theta_{2,k}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
T_m	$\theta_{m,1}$	$\theta_{m,2}$	$\ldots$	$\theta_{m,k}$

where $\theta_{i,j} = \begin{cases} 1, & \text{if } S_j \in \theta(T_i) \text{ (the } i\text{−th task requires the } j\text{−th skill)} \\ 0, & \text{otherwise.} \end{cases}$

For simplicity, it is assumed that a single task can be executed by a single agent, one task at a time. Task T_j can be executed by agent A_i if and only if the agent possesses all the required skills. Formally, skills associated to tasks (obtained by the projection on $2^{\mathbb{S}}$ in a domain of θ) should be contained in skills (obtained by the projection on $2^{\mathbb{S}}$ in a domain of σ) associated to agents from $\mathbb{A}$. Symbolically:

$$\pi_{2^{\mathbb{S}}}\Big(dom\{\theta(T_l, S_j)\}\Big) \subseteq \pi_{2^{\mathbb{S}}}\Big(dom\{\sigma(A_i, S_j)\}\Big),$$

and the execution can start whenever the agent is free; this holds for all $i \in \{1, 2, \ldots n\}$, $j \in \{1, 2, \ldots k\}$ and $l \in \{1, 2 \ldots, m\}$.

Now, roughly speaking, the problem consists in efficient assignment of all the tasks to given agents, so that the tasks can be executed, all the constraints are satisfied, and the total execution time will perhaps be minimal. More precisely, the assignment problem is defined as follows.

1.3.3 Types of Temporal Constraints of PM-AS-PP

As mantioned, temporal constraints in both M-AS-PP and and PM-AS-PP might be divided into two groups:

- *Hard constraints* and
- *Soft constraints.*

Hard constraints are specified as these constraints that *should be violated* in a scheduling task. They ensure a feasibility of the scheduling task. *The soft constraints* may not be satisfied, but a degree of their satisfaction is a measure how good is a scheduling plan. To make requirements with respect to the hard constraints more liberal, we use the so-called relaxation, i.e. a weakening of the strong constraints. We often use this solution, when satisfaction of all hard constraints leads to an inconsistency.

For a mathematical representation of temporal constraints imposed on PNJSP we introduce the following set of parameters. Instead of agent skills we will consider agent roles (contracts)[6]:

- $N = \{n_1, n_2 \ldots, n_k\}$ as set of agents (agents),
- $R = \{r_1, r_2, \ldots, r_k\}$ as a set of roles (contracts)
- $D = \{d_1, d_2, \ldots, d_k\}$ as a set of days in a week,
- $Z = \{z_1, z_2\}$ as a set of admissible shifts during days from D,
- $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ as a set of actions.

It enables representing now M-ASP-P by its formal instances in the form of the triple

$$(N, D, Z, A, HC), \tag{1.4}$$

where N, D, Z are given as above and HC denotes a set of hard constraints imposed on actions from A and their performing. Similarly, PM-ASP-P may be given by the n-tuple of the form:

$$(N, D, Z, A, HC, SC, P), \tag{1.5}$$

where where N, D, Z and HC are given as above and SC and P denote a set of soft constraints and preferences (*resp.*) Introducing SC to the n-tuple 1.7 follows from the adopted hierarchy of constraints. The hard constraints cannot be violated, the soft ones may be violated, but they should be satisfied before preferences.

This notation allows us to elaborate the following representation of hard and soft constraints. Since their list is not exhaustive[7], it might be relatively naturally extended.

HC 1: The charm of the shift organization should be fair: each agent must to have equally 2–day shifts and 2–night shifts

Assume that Z_{day} denotes a set of day-shifts and Z_{night} denotes a set of night-shifts. Then this strong constraint may be shortly mathematically rendered as follows:

$$\sum_{z \in Z_{day}} X_{n,d,z} = 2 \wedge \sum_{z \in Z_{night}} X_{n,d,z} = 2. \tag{1.6}$$

HC 2: Each agent can be associated to at most one shift

This strong constraint can be shortly mathematically expressed as follows:

$$\sum_{z \in Z} X_{n,d,z} = 1. \tag{1.7}$$

[6] All of these constraints are typical for scheduling problems of this type to be known as (usually) NP-hard – see: [13].

[7] This fact plays no important role as the main objective of this juxtaposition consists in the *quantitative representation* alone, which will be later combined with qualitative temporal constraints (of Allen's sort) for a use of further investigations.

HC 3: Some shifts are prohibited for an agent n

This strong constraint renders the following prohibition: some shifts are prohibited for an agent n and a relaxation usually cannot be referred to it. If Z_n denotes a set of shifts prohibited for a agent $n \in \{N_1, N_2 \ldots, N_k\}$, then this constraint can be mathematically depicted as follows:

$$\sum_{z \in Z_n} X_{n,d,z} = 0. \tag{1.8}$$

HC4: Length of the shifts sequence associated to an agent

It is a strong constraint that defines a restriction for the sequence of shifts associated to a agent. If we denote by a minimal and a maximal number of shifts associated to a agent n by m_z^{min} and by m_z^{max} (resp.), then this constraint can be rendered as follows:

$$m \leq \sum_{d}^{m_z+d} X_{n,d,z} \leq M\,. \tag{1.9}$$

HC 5: Quantity of shifts in a scheduling period

It defines the minimal and the maximal quantity of shifts during a given scheduling period (day, months, etc. − associated to a single agent.) If we denote − the minimal and the maximal quantity of shifts that can be obtained by agents in a given scheduling period by s and by S (resp.), then this constraint can be rendered mathematically as follows:

$$s \leq \sum X_{n,d,z}^{d \in D} \leq S\,, \tag{1.10}$$

where $X_{n,d,z}$ is defined as above.

HC6: The Quantity of the shifts per day is restricted

It defines the minimal and the maximal quantity of temporal shifts during a day − associated to a single agent. If we denote the minimal and the maximal number of agents in a role r − which should obtain a shift z during a day d − by r and by $R(resp.)$, then this constraint may be mathematically rendered as follows:

$$r \leq \sum^{n \in N} X_{n,d,z} \leq R\,, \tag{1.11}$$

where $X_{n,d,z}$ is defined as the following characteristic function[8]:

$$X_{n,d,z} = \begin{cases} 1 & \text{if an agent } n \text{ works a shift } z \text{ in a day } d, \\ 0 & \text{otherwise.} \end{cases}$$

SC 7: Preferential quantity of shifts in a scheduling period

This constraint forms an effect of a relaxation of the constraint HC5 referring to the quantity of shifts in a scheduling period. In the framework of this constraint, some preferential value R^{best}, defining a preferred number of agents in a role r that should be associated to a shift z during a day d. It is easy to observe that a materialization of this preference should be mathematically expressed by the fact that the real value of $\sum_n X_{n,d,z}$ is always smaller than an arbitrary chosen $\epsilon > 0$. Formally:

$$|\sum_{n \in N} X_{n,d,z} - R^{best}| = 0\,. \tag{1.12}$$

SC 8: Preferential scheduling charm's covering by shifts during a working day

This constraint forms an effect of a relaxation of the constraint HC6 referring to the daily covering of the scheduling charm's by shifts. In the framework of this constraint, some preferential value S^{best} defines a

[8]This binary representation can be also exchanged by a classical one: $X_{n,d} = z$ as presented in [13].

preferred number of shift, that should be associated to an agent n during a day d. It is easy to observe that this preference will be materialized, when the value $\sum_{d\in D_h} X_{n,d,z}$ is possibly close to the preferred S^{best} one. If $|d|$ denotes a lenght of a working day and $\sum_{d\in D_h} X_{n,d,z}$ (normally equal to a natural number) is normalized by a normalization factor, SC8 may be rendered by means of ϵ-neighborhood of a desired value as follows.

$$\forall \epsilon > 0 \quad \forall x, y\Big(|x-y| < |d| \Rightarrow |\sum_{z\in Z} X_{n,d,z}(x) - S^{best}| < \epsilon\Big)^{9}. \tag{1.13}$$

SC 9: Preferential length of the shifts sequence associated to an agent during a day

This soft constraint forms a result of a relaxation of the constraint HC4 concerning the length of the shift sequence associated to an agent n. Assumed that the preferred value m^{best} is given and a day d is fixed. Since both the sum $\sum_{z\in Z} X_{n,d}$ and m^{best} (without a normalization) must be natural numbers, this preference will be materialized if these values are equal. Formally:

$$|\sum_{z\in Z} X_{n,d} - m^{best}| = 0\,. \tag{1.14}$$

Finally, we can also introduce some exemplary preferences[10] imposed on actions.

Preference 1: Number of actions associated to an agent (generally) should be no greater than l and no smaller than k

$$\forall n (k \leq \sum_{a\in\mathcal{A}} X_{n,d,z} \leq l)\,. \tag{1.15}$$

Preference 2: Number of actions associated to each agent during a day cannot be greater than M and smaller than K

$$\forall n \forall d (K \leq \sum_{a\in\mathcal{A}} X_{n,d,z} \leq M)\,. \tag{1.16}$$

Preference 3: An agent n_k prefers to perform an action a with a degree $> \alpha$ in a scheduling period.

$$\alpha \leq \sum_{d\in D_h}\sum_{z\in Z} X_{n_k,a,z,d}\,. \tag{1.17}$$

Obviously, the list of constraints is not exhaustive and it may be naturally enlarged. However, we interrupt their presentation in this point to solve the problem of the appropriate representation of the problems that were indicated.

1.4 Multi-Agent Schedule-Planning Problem *versus* Temporal Traveling Salesman Problem – Their Modeling and Representations

We have just introduced a subject-specification of temporal planning area. In fact, we distinguished two classes of problems:

- problems of the type of *Temporal Traveling Salesman Problem* (TTSP) and
- problems of the type of *Multi-Agent Schedule-Planning Problem*(MA-S-PP).

[9] Note that $X_{n,d,z}(x)$ may be slightly reformulated to be a function of an argument of x.

[10] These preferences will be called *internal preferences* as they form preferences of an operating agent, inside of the system. We will distinguish them from *global preferences* that will be introduced in chapter 1 of 'Contributions'.

Both TTSP and MA-SP-P play a role of the paradigmatic problems in their classes. Both problems will be formally represented and modeled in 'Contributions' (MA-SP-P constitutes a subject-basis for analyses in Chapter 1 and Chapter 2. TTSP is used in this role in Chapter 3 and Chapter 4.)

It seems to be clear that these two problems – as a matter of their different nature – should be differently represented and modeled. To give some hints how to do this, let us return to Definition 19 of TTSP. Two types of functions were singled out in it:

- γ as the function assigning time to edges, $\gamma\colon E \to \Delta$, where Δ is the set of admissible (floating) intervals of time,
- δ as a unique mapping defining time necessary for operation at any node $v \in V$.

Meanwhile, MA-SP-P is defined in terms of a finite set A of agents, their skills or ability to work and associated actions. The ability is measured by the appropriate values with normalized values.

Summarizing these differences one can state that:

1 M-AS-PP requires some computable portion of knowledge about skills or ability of agent to work. In addition, some flexibility and continuity of working cycles and shifts play a crucial role.

2 In TTSP – representable rather by graphs than summary diagrams – temporal agent operations at nodes are combined with (temporally restricted) agent moves along graph edges. In addition, a combination of temporal and spatial components itself plays even more significant role than arithmetically depicted temporal restrictions imposed on agent activity.

Technically speaking, the difference may be rendered as follows. The appropriate representation of M-AS-PP should be based – at first – on intervals of the form $(d,z)^n_a$ (representing time in terms of a day d and a shift z for an agent n performing an action a). Secondly – this representation should base on normalized sums of a general form $\sum X_{d,z,a,n}$. They represent an ability of n with respect to a in an arithmetic way in terms of the function $X(d,z)$.

In contrast, the appropriate representation of TTSP should be rather rendered in terms of mixed modal formulas, such as the exemplary one:

$$[\text{K prefers}_{C_1}]\langle\text{Later}\rangle\text{Deliver}^{A}_{C_k}\,{}^{11}. \tag{1.18}$$

The detected differences seem to suggest a direction of modeling and representation of both problems. We intend to show that:

1 M-AS-PP is adequately representable in the computational 'analytic' framework of the convolution-based approach,

2 TTSP is adequately representable in the logical framework, in terms of some Preferential Halpern-Shoham logic.

At the end of this chapter, we put forward a brief road map informing about problems, associated to difficulties D1, D2, D3, D4, with the methods of their overcoming.

[11] In this formula the outer operator [K prefers]ϕ plays a role of a box-type operator for representation of preference of K and $\phi = \langle$ Later $\rangle\psi$ plays a role of (an additionally specified) $\langle L\rangle$-operator of HS logic. Finally, $\psi = \text{Deliver}^{A}_{C_2}$ is already a unique atomic formula. We return to formulas of this type in chapter 3 of 'Contribution' devoted to Preferential Halpern-Shoham Logic.

Type of Difficulty	**Chapters/section**	**Problem**	**Overcoming methods**
D1 (subject-specification of temporal planning)	Chapter 1,2 (for M-A-S-PP)	To formally represent and model fuzzy temporal constraints and preferences with M-A-S-P-P as a subject-basis	Combining fuzzy temporal constraints for M-A-S-P-P with Ohlbach's fuzzy Allen's relations
D1 (subject-specification of temporal planning)	chapter 3,4 (for TTSP)	To formally represent and model fuzzy temporal constraints and preferences with TTSP as a subject-basis	Combining Multi-valued Logic for Preferences and Halpern-Shoham Logic interpretable in fibred semantics
D2 (dualism: qualitative-quantitative)	(Chapter1,2)	To formally represent and model fuzzy temporal constraints and preferences with M-A-S-P-P as a subject-basis	Combining fuzzy temporal constraints for M-A-S-P-P with Ohlbach's fuzzy Allen's relations
D2 (dualism: qualitative-quantitative)	(Chapter 5)	To reconcile the analysis-based approach with the logical one in temporal planning	Construction of the hybrid plan controller. The logical approach delivers a description of the robot motion and plan; the robot trajectories represented by integrable functions.
D3 (Difficulty of Ohlbach's approach)	Chapter 1	To find a deeper theoretic grounding for Ohlbach's proposal	Introducing a portion of real analysis-based mathematical theory for fuzzy Allen's relations representable by convolutions in $\mathbf{L}^2(\mathbb{R})$ -space of Lebesgue integrable functions.
D4 (unclear knowledge about a real utility of planning methods)	(Chapter 1/section2)	To propose the fuzzy temporal and preferential extensions of STRIPS and of Davis-Putnam procedure	Adaptation of the convolution-based representation of fuzzy Allen's relations to the original methods

Chapter 2

Temporal Planning with Fuzzy Temporal Constraints and Preferences. The Real Analysis-Based Depiction

Inhaltsangabe

2.1 Introductory Remarks **89**
2.1.1 Motivation of Current Analyses 89
2.2 Mathematical Foundation of the Convolution-based Approach to Fuzzy Allen's Relations **90**
2.2.1 Motivation of the Convolution-based Approach 90
2.2.2 Terminological Framework of the Discussion 91
2.2.3 Mathematical Foundation in Terms of Real and Abstract Analysis . . . 93
2.2.4 Algebraic Aspects of the Convolution-based Fuzzy Allen's Relations . . . 99
2.3 Modeling of Fuzzy Temporal Constraints and Preferences **101**
2.3.1 Fuzzy Temporal Constraints in a New Representation 101
2.3.2 Preferences in a New Representation. 107
2.4 Fuzzy Temporal Constraints and Preferences in STRIPS and Davis-Putnam Procedure **109**
2.4.1 Fuzzy Temporal Constraints and Preferences in STRIPS 109
2.4.2 Towards Temporal-Preferential STRIPS 110
2.4.3 Fuzzy Temporal Constraints and Preferences in Davis-Putnam Procedure . . . 113
2.5 Concluding Remarks **116**

Abstract. This chapter[1] is devoted to mathematical foundations for the convolution-based depiction of fuzzy Allen's interval relations. This convolution-based depiction is later used to define qualitative-quantitative fuzzy temporal constraints and preferences in a new way. Next, it is shown how these new definitions may be adopted to such planning procedures as STRIPS and Davis-Putnam procedure and how these procedures may be temporally and preferentially extended.

2.1 Introductory Remarks

It is perhaps surprising that a majority of approaches to the representation of fuzzy temporal constraints and preferences does not exploit more advanced mathematical machinery – offered by real analysis, functional analysis or control theory. The pioneering papers of H-J. Ohlbach [86, 85] seem to be ground-breaking, since they make available many computational results and techniques of real analysis and measure theory with respect to these issues. In fact, fuzzy Allen's relations in terms of convolutions, as 'Introduction' presents, immediately obtain a couple of new operational (analytical and algebraical) features. Thanks to this solution, their definitional elusiveness becomes smaller.

2.1.1 Motivation of Current Analyses

A majority of works – as earlier mentioned – such as: [18, 19, 75, 3, 20, 80, 81, 82] avoids the following two problems:

P1 how to elucidate and describe computational side of these entinites and

P2 how to reconcile the qualitative and quantitative nature of them.

It seems that a proposal of H-J. Ohlbach from [86, 85] constitutes a promising 'bridgehead' to overcome these difficulties. Unfortunatelly, this interesting proposal of H-J. Ohlbach suffers from the following (fortunately: rather not-severe problems and lacks):

Lack1 At first, it still waits for a detailed mathematical grounding.

Lack2 Secondly, this proposal in itself is not suitable to be reconciled with temporal constraints different from Allen's relations. As an aftermath – the Ohlbach's integrals are not adaptable to temporal constrains describing our Multi-Agent Problem.

Lack3 Next, Ohlbach's constructions are partially conditional. For example, his integrals must be finite, etc.

Lack4 Finally, one could argue that integrals are not completely capable of reflecting the proper nature of all fuzzy Allen's relations. For example, fuzzy Allen's relations may be seen as obtained by a combinations of two functions (of different arguments) characterizing different fuzzy intervals. Meanwhile, a usual integral is built up from integrands of the same argument.

All these facts and difficulties of Ohlbach's approach and a pressing need to counter them determine the main motivation factor of current considerations. These lacks motivate us to put forward a slightly more advanced proposal to represent fuzzy Allen's relations in terms of convolutions as norms of them. It opens new possibilities to carry out a deep-analysis of the concept of fuzzy Allen's relations as immersed in a unique Banach space of Lebesgue integrable functions. It also constitutes a convenient bridgehead to elaborate a real analysis-based theory of fuzzy Allen's relation. This theory may be easily complemented by a purely algebraic contribution.

[1]The current depiction of the issues of this chapter was improved after remarks and comments of anonymous reviewers of the conferences: IJCAI2017 and KR2016. In addition, some issues of this chapter, such as approximation in terms of Hardy-Littlewood Theorem and the Convolution Theorem are included in the paper 'Fuzzy Temporal Reasoning–a Perspective of the Logical and Computational Complementation', submitted to: *WIRE Data Mining and Knowledge Discovery.*

Objectives and novelty of investigations.

This chapter has the following three objectives.

Obj1 The first purpose is to propose a new convolution-based representation of fuzzy Allen's relations and to elaborate an algebraic and a real analysis-based foundation for this representation.

Obj2 The second purpose is to exploit a new approach for defining *(general) fuzzy temporal constraints* and (global) preferences on a base of the convolution-based representation of fuzzy Allen's relations – introduced in [86, 85].

Obj3 Finally, the third purpose is to adopt this new conceptual apparatus to extend two planning methods: STRIPS-algorithm and Davis-Putnam procedure.

For these reasons, we intentionally put aside our two leading problems of temporal planning in order to develop our conceptualization framework to investigate these two problems once again in a new way.
A novelty of the current approach – in a comparison to the original Ohlbach's contributions – consists in the following facts:

Nov1 An alternative convolution-based approach to fuzzy Allen's relations, inspired by Ohlbach's integral-based approach, is proposed.

Nov2 A new general type of temporal constraints will be proposed (on a base of some synthesis of the quantitative ones with Allen's relations).

Nov3 Ohlbach's convolution-based approach is extended for a case of new general fuzzy temporal constraints and preferences and based on a new broader foundation of Lebesgue integrable functions.

Nov4 finally, a complete fuzzy logic system for Ohlbach's convolutions is introduced and explored (in Appendix).

2.2 Mathematical Foundation of the Convolution-based Approach to Fuzzy Allen's Relations

2.2.1 Motivation of the Convolution-based Approach

In Section 4.2 of 'Introduction' Ohlbach's interval approach to representation of fuzzy Allen's relations was described in detail. It emerged that this approach may be rendered in terms of probability theory and it is multi-stages. Recall briefly the stages of his construction. It begins with:

- *Fuzzy Allen's relations of the point- interval* type. They assert that a point, say p, remains in R-relation to a fuzzy interval j. Symbolically: $R_p(j)$, where R is a chosen Allen's relation.

- *Fuzzy Allen's relations of the fuzzy interval- fuzzy interval* type, if blow points p's to a new fuzzy interval, say i. Since each $R_p(j)$ may be interpreted as a distribution function, it also has a density-based representation. They are interpreted as expected values of the general form:

$$R(i,j)^{Fuzzy} = \int_{-\infty}^{\infty} i(x)\widehat{R_p}(j)(x)\mathrm{d}x, \tag{2.1}$$

where $\widehat{R_p}(j)(x)$ forms a density-based representation of $R_p(j)(x)$. The right side of (2.1) is also normalizable to ensure that it will takes values from $[0,1]$.

To cut the long story short, Ohlbach proposes to see fuzzy Allen's relations as *normalized integrals of a single variable*. Meanwhile, one has an impression that this interpretation is not sufficient. It follows from the following reasons:

A Fuzzy Allen's relations are viewed here as fuzzy values of integrals, what seems to be an excessive simplification.

B This approach 'escapes' towards reasonings based on probability theory and statistics instead of real-analysis and algebra-based reasonings.

C It seems to not (completely) properly emphasize a sense of some definitions of fuzzy Allen's relations.

In order to elucidate **C**, let us return to this fragment of $meet^{Fuzzy}(i,j)$-definition, given in Section 4.3:

$$meet(i,j)^{Fuzzy} = \int_{-\infty}^{\infty} \frac{Fin(i)St(j)dx}{N(Fin(i), St(j))} \tag{2.2}$$

As earlier explained, this definition is based on two functions: $Fin(i)$ and $St(j)$ that 'cut' the initial part of i and the final part of j (resp.). Obviously, both intervals should be integrated together. It could be materialized as follows. One one hand, one can take a function, say $Fin(i)$ 'running' towards the interval j. On the other hand, one can take a similar function $St(j)$ 'running' in the inverse direction – towards the interval i. It means that $St(j)$ should be rather consider as a function of a new argument, say t, and $Fin(i)$ as $Fin(i)(x-t)$[2]. Meanwhile, (2.2) interprets both $Fin(i)$ and $St(j)$ functions of the same argument $x \in \mathbb{R}$.

It remains what is a nature of such a combination of $Fin(i)$ and $St(j)$. The natural candidate to represent such a 'combination' of mutually independent functions is a *mathematical convolution* of functions. In order to confirm this conjecture, one needs to enlarge a terminological background of the discussion.

2.2.2 Terminological Framework of the Discussion

The notion of convolution, that we need, requires a new conceptual framework. Its determined by a class of Lebesgue integrable functions on $\mathbb{R}$, denoted by $L(\mathbb{R})$. This class forms a unique example of the so-called *Banach spaces*.

In order to describe both types of spaces, assume that X is a given vector (linear) space[3]. Each vector space is defined over a scalar field, say K. This fact is denoted by $X(K)$. The usual scalar fields are: the field R of real numbers $\mathbb{C}$, or over the field $\mathbb{C}$ of complex numbers. We write then: $X(\mathbb{R})$ and $X(\mathbb{C})$ (*resp.*) to render the fact that X is defined over $\mathbb{R}$ or $\mathbb{C}$.

Assume also that $X(\mathbb{R})$ is given. Let us introduce now a new function $\|\bullet\| : X(\mathbb{R}) \mapsto [0,\infty)$ that respects the following conditions:

1 $\|x\| = 0 \iff x = 0$.

2 $\|\alpha x\| = |\alpha|\|x\|$, for $\alpha \in \mathbb{R}$.

3 $\|x + y\| \leq \|x\| + \|y\|$.

This function is to be called *a norm* and the whole space $(X(\mathbb{R}), |\bullet|)$ forms a *normed space*.

Example. Different types of the normed spaces are used in functional analysis and its mathematical applications. The simplest examples are:

[2]The argument $x - t$ ensures that i and j meet together. Thus, $Fin(i)$ should be seen as a function of an argument x in t-translation.

[3]See: Appendix.

$(\mathbb{R}, |\bullet|)$ with $\|x\| = |x|$, for each $x \in \mathbb{R}$.

Another kind of examples one often sees in mathematics are presented in the table below.

A *Banach space* is such a *normed vector space* X, which is complete with respect to that norm, that is to say, each Cauchy sequence $\{x_n\}$ in X converges to an element x in X.

This last condition may be written as:

$$\lim_{n\to\infty} x_n = x, \tag{2.3}$$

or – alternatively – as follows:

$$\|x_n - x\|_X \to 0. \tag{2.4}$$

Type of spaces	**Notation**	**Elements of the space**	**Norms**
The space of real or compact numbers	$\mathbb{R}$, $\mathbb{C}$	Real or compact numbers	$\|x\| = \|x\|$
The euclidean space of real or compact numbers	$\mathbb{R}^n$, $\mathbb{C}^n$	Sequences of the form: $(x_1, x_2, \ldots, x_n)$	$\|x\| = \left(\sum_1^\infty \|x\|^2\right)^{\frac{1}{2}}$
The space of infinite sequences	l_∞	Infinite sequences $(x_1, x_2, \ldots)$	$\|x\| = \sup_i \|x\|$
The space of continuous functions on $\mathbb{R}$	$C(\mathbb{R})$	Functions $f, g, h \ldots$– continuous on $\mathbb{R}$	$\|x\| = \sup_{x\in\mathbb{R}} \|f(x)\|$
The space of Lebesgue integrable functions on $\mathbb{R}$ 'in square'	$L^2(\mathbb{R})$	Functions $f, g, h, \ldots$ - Lebesgue integrable on $\mathbb{R}$	$\|f\| = \left(\int \|f\|^2 \mathrm{d}x\right)^{\frac{1}{2}}$
The space of Lebesgue integrable functions on $\mathbb{R}$ 'in p'	$L^p(\mathbb{R})$	Functions $f, g, h, \ldots$ - Lebesgue integrable on $\mathbb{R}$	$\|f\| = \left(\int \|f\|^p \mathrm{d}x\right)^{\frac{1}{p}}$, $1 < p < \infty$

This consideration allows us to define convolutions for Lebesgue integrable functions.

Definition 17 ***(Convolution.)*** *Let us assume that functions f and g are Lebesgue integrable in $\mathbb{R}$, i.e they belong to $(L^1(\mathbb{R}, \|\bullet\|)$. Then the convolution of f and g – denoted by $f * g$ – is usually defined as follows:*

$$(f * g)(x) = \int_{-\infty}^{\infty} f(x-t)g(t)dt\,, \tag{2.5}$$

where the right side is an improper (Riemann) integral.

Independently of algebraic features (for details – see: [98], pp. 159-161), convolutions show some computationally useful property. This property is expressible in terms of the *Fourier transform* $\mathcal{F}$ of a function $f \in L(\mathbb{R})$, defined for all $x \in \mathbb{R}$ as follows:

$$\mathcal{F}(x) = (2\pi)^{-\frac{1}{2}} \int_{\mathbb{R}} f(t)e^{-ixt}\mathrm{d}t. \tag{2.6}$$

The so-called (Borel's) Convolution Theorem asserts that the Fourier transform of a convolution of two functions is equal to a multiplication of Fourier transforms or each of these functions [4]. This theorem (in a

[4]The operational utility of this theorem also consists in some 'economy' of numerical computation if implement such computation for Allen's relation on computer. In fact, the standard convolution algorithm has quadratic complexity. The Convolution Theorem reduces it from $O(n^2)$ to $O(n \log n)$.

version for $L(\mathbb{R})$) may be rendered as follows.

Convolution Theorem. If $f, g \in L(\mathbb{R})$ and $\mathcal{F}$ is a Fourier transform, then

$$\mathcal{F}(f * g) = \mathcal{F}(f)\mathcal{F}(g). \tag{2.7}$$

The important formal machinery is also delivered by Fubini's Theorem. This theorem asserts that a double integration is equal (under some conditions) to a single integration on the appropriate product space. This fact will be exploited in further considerations.

The double integration is defined with respect to the so-called *product measure*. If two measurable spaces $(X, \sigma_1, \mu_1), (Y, \sigma_2, \mu_2)$ are given, then a *product measure* $\mu_1 \times \mu_2$ is defined to be such a measure on the measurable product space (where $X \times Y, \sigma_1 \otimes \sigma_2$) ($\sigma_1 \otimes \sigma_2$ is ρ-algebra on the Cartesian product $X \times Y$) that the following holds:

$$\mu_1 \times \mu_2(A \times B) = \mu_1(A)\mu_2(B),$$

where $A \in \sigma_1, B \in \sigma_2$.

Theorem 2 ***(Fubini)****([99], p. 386) Suppose that X and Y are ρ_1- and ρ_2-measure space (resp.) with measures μ and v (resp.) and suppose that the product space $X \times Y$, $f(x, y)$ is $\mu \times v$-measurable on $X \times Y$ and at least one of the three integrals*

$$\int_{X \times Y} |f(x, y)| \mathrm{d}\mu \times v(x, y) < \infty \tag{2.8}$$

$$\int_Y \int_X |f(x, y)| \mathrm{d}\mu(x) \mathrm{d}v(y) < \infty \tag{2.9}$$

$$\int_X \int_Y |f(x, y)| \mathrm{d}v(y) \mathrm{d}\mu(x) < \infty \tag{2.10}$$

is finite. Then:

A *the function $x \to f(x, y)$ is in $L^1(X, \rho_1, \mu)$ (Lebesgue integrable in this space), for almost all $y \in Y$,*

B *the function $y \to f(x, y)$ is in $L^1(X, \rho_2, v)$, for almost all $x \in X$,*

C *the function $y \to \int_X f(x, y) \mathrm{d}\mu(x)$ is in $L^1(Y, \rho_2, v)$,*

D *the function $x \to \int_Y f(x, y) \mathrm{d}v(x)$ is in $L^1(Y, \rho_1, \mu)$*

and the following holds:

$$\int_X (\int_Y f(x, y) \mathrm{d}y) \mathrm{d}x = \int_Y (\int_X f(x, y) \mathrm{d}x) \mathrm{d}y = \int_{X \times Y} f(x, y) \mathrm{d}(x, y). \tag{2.11}$$

2.2.3 Mathematical Foundation in Terms of Real and Abstract Analysis

Let us return to the initial problem of the convolution-based representation of fuzzy Allen's relations. It has already been said that convolutions are suitable to represent mathematically a 'combination' of two independent functions of different variables. Physics adds a new interpretation of a role of convolution. It represents two independent signals running in inverse directions. Obviously, both functions $Fin(j)(t)$ and $St(i)(x - t)$ from $meet^{Fuzzy}$-definition behave in the same way. Finally, there is another argument for the convolution-based representation of fuzzy Allen's relations. Namely, it is a known fact of real analysis that convolutions take finite values. It means that their use would make a mathematical discussion on fuzzy Allen's relations less conditional than the earlier one in terms of Ohlbach's integral approach. In fact, a success of Ohlbach's approach is possible, provided that the appropriate integrals are finite. In the convolution-based approach this problem disappears thanks to this elementary property of convolutions.

In this convention, fuzzy Allen's relations should be rather rendered as follows:

$$\text{meet(i, j)}^{Fuzzy}(x) = \int_{-\infty}^{\infty} \frac{Fin(i)(x-t)St(j)(t)dt}{N(Fin(i)(x-t), St(j)(t))} \tag{2.12}$$

(Note that $\text{meet(i, j)}^{Fuzzy}$ is a function of x-argument as we integrate with respect to the second argument t.) In the similar way, one could modify other definitions of fuzzy Allen's relations. For example[5]:

$$\text{before(i, j)}^{Fuzzy} = \int_{-\infty}^{\infty} i(x-t)\widehat{B}(j)(t)\mathrm{d}t/|i|, \quad \text{during(i, j)}^{Fuzzy} = \int_{-\infty}^{\infty} i^{'}(x-t)\widehat{D}(j)(t)\mathrm{d}t/|i^{'}|, \tag{2.13}$$

where $i^{'}$ is defined from i as explained in Section 4.3.

Fuzzy Allen's relations as norms of convolutions. It seems that an idea to represent fuzzy Allen's relations in terms of convolution is just the appropriate one. As illustrated, convolutions better 'encode' an idea of a combination of two functions, as in 2.10. In addition, convolutions are computationally convenient, as they are finite. Of course, they might be also normalized by a normalization factor to ensure that they take values from $[0, 1]$.

Nevertheless, it seems that convolutions still are not ideal in this role. In order to illustrate it, let us consider the following paradoxical dichotomy.

A On one hand, assume that we have two different fuzzy Allen's relations, say $R_1^{Fuzzy}(i, j)$ and R_2^{Fuzzy} given by two normalized convolutions C_1 and C_2 (resp.). Assume that they take the same values $\alpha \in [0, 1]$ in some area. (They diagrams are identical, at least in some area).

B On the other hand, assume that we have a single fuzzy Allen's relation, say $R^{Fuzzy}(i, j)$, depicted by a normalized convolution C. Anyhow, C may be multiplied by a scalar $\alpha \in \mathbb{R}$. This new convolution αC would take another values than C alone, although it essentially represent the same initial $R^{Fuzzy}(i, j)$.

Obviously, we are rather willing to consider C and αC as mutually linked (they represent $R^{Fuzzy}(i, j)$ and $\alpha R^{Fuzz}(i, j)$[6]). Simultaneously, we rather want to see C_1 and C_2 as the mutually independent, even though their diagrams are (at least partially) identical. In fact, they represent different fuzzy Allen's relations $R_1^{Fuzzy}(i, j)$ and $R_2^{Fuzzy}(i, j)$.

Therefore, as already suggested, convolutions do not constitute the appropriate representation of fuzzy Allen's relations. Fortunately, it appears that *norms from convolutions* may avoid the difficulties. To confirm this hypothesis, assume that $R^{Fuzzy}(i, j)$ is represented now by two norms, say $\|\|_1$ and $\|\|_2$. Formally, we postulate

$$R^{Fuzzy}(i, j) = \|C\|_1, \quad R^{Fuzzy}(i, j) = \|C\|_2. \tag{2.14}$$

Let us state that $\|C\|_1$ and $\|C\|_2$ may be viewed as *mutually equal* provided that there are such real constants $a, b < \infty$ that:

$$a\|C\|_1 \leq \|C\|_2 \leq b\|C\|_1{}^7. \tag{2.15}$$

Note that in our case this condition is satisfied. In fact, it is enough to put $a = b = \alpha$ (see: point **B** above). In fact, (2.13) renders the following fact: 'even if two norm-values are associated to a given fuzzy Allen's

[5] For brevity of the presentation we avoid conditions for intervals i and j)

[6] Note that this scalar multiplication is not defined in the initial Allen's algebra. It is, somehow, thinkable in the convolution-based representation of fuzzy Allen's relations. However, we are not interested in a sense of such an operation from a non-mathematical point of view.

[7] This property is a known property of norms.

relation, they remain mutually equivalent. In the whole current analysis, a sense of the statement: 'two norms are mutually equal' is just given by (2.13).[8]

This mutual equivalence of norms just delivers an argument to represent fuzzy Allen's relations by *norms of convolutions*.

Formally, if $R(i,j)$ denotes a convolution 'basis' of $R^{Fuzzy}(i,j)$, then $R^{Fuzzy}(i,j)$ can be written:

$$R^{fuzzy}(i,j) = \|R(i,j)\|_{L(\mathbb{R}^1)}. \tag{2.16}$$

Example 24 . *Assume that Allen's relation 'before' in a convolutive representation $B(i,j) = f^i \star B_p(j)$ is given, for some functions f^i and $B_p(j)$. Then fuzzy 'before' B^{fuzzy}:*

$$B^{fuzzy}(i,j) = \|B(i,j)\|_{L(\mathbb{R}^1)}. \tag{2.17}$$

But $B(i,j)$ is a convolution, so $B(i,j)(x) = \int_{-\infty}^{\infty} f^i(x-t)B_p(j)(t)\mathrm{d}t$. Assuming that $B(i,j)(x) \in L(\mathbb{R})$[9] *with the norm $\|f\| = \int |f(x)|\mathrm{d}u$, for each $f \in \Big(L(\mathbb{R}^1), \|\ \|\Big)$, we have*

$$B^{fuzzy}(i,j) = \| \int_{-\infty}^{\infty} f^i(x-t)B_p(j)(t)\mathrm{d}t\|_{L(\mathbb{R}^1)} = \int | \int_{-\infty}^{\infty} f^i(x-t)B_p(j)(t)\mathrm{d}t|\mathrm{d}\mu. \tag{2.18}$$

Remark 1 *Note that the 'external' Lebesgue integral in (2.16) may be also exchanged for the appropriate Riemann integral. However, we omit this step.*

Summing up, our initial proposal with respect to the method of representation of fuzzy Allen's relations may be rendered briefly as depicted in the table.

Type of relations:	**Representation:**	**Examples:**
Allen's relations of the type: 'interval-interval'	expected values	If $i(x)$ and $R_p(j)(t)$ are given and defined as usual, then $E(i(x)) = R(i,j)^{Fuzzy}(x) = \int_{-\infty}^{\infty} i(x)\widehat{R_p(j)(x)}\mathrm{d}x$, ($\widehat{R_p(j)(x)}$ – a density function for $R_p(j)(t)$)
Fuzzy Allen's relations of the 'interval-interval' type	L($\mathbb{R}$)-norms of convolutions	$R^{fuzzy}(i,j) = \|R(i,j)(x)\|_{L(\mathbb{R})} = \int (\int_{-\infty}^{\infty} f^i(x-t)R_p(j)(t)\mathrm{d}t)\mathrm{d}\mu$

However, it still arises a pressing need to check whether Ohlbach's convolutions (representing Allen's relations, such as $R(i,j)$ in the table above) still belong to $\Big(L(\mathbb{R}^1), \|\ \|\Big)$. The following lemma positively answers to this question.

Lemma 1 *Suppose that f^i and $R_p(j)$ are respectively: a function characterizing a fuzzy interval i and a point-interval relation for a fuzzy interval j. Assume also that they are both Lebesgue integrable. Thus $\int f^{i(x)}(x-t)R(j)(t)\mathrm{d}\mu$ is also Lebesgue integrable and $\|f^i * R(j)\| \leq \|f^i\|\|R(j)\|$.*

[8] In essence, the mutual equivalence of norms often means that they generate the same metrics and topologies. We will not make use of this interpretation as we are not interested in metric aspects of L($\mathbb{R}$)-spaces. Indeed, the fact that two norms may be linked as given in (2.13) plays a more important role.

[9] This fact will be proved later.

Proof: Suppose that $(x,t) \mapsto f^{i(x)}(x-t)R_p(t)$ is assumed to be $\mu(x) \times v(t)$-measurable on $\mathbb{R}^2$. In order to show that $f^{i(x)} * R_p(j) \in L^1(\mathbb{R}^1)$ it is enough to make use of Fubini's theorem.

For that reason, one needs to verify that the hypothesis of Fubini's theorem is satisfied. In order to make it, consider the double integral:

$$\int_{\mathbb{R}} | \int_{\mathbb{R}} |f^{i(x)}(x-t)R_p(j)(t)| \mathrm{d}x\mathrm{d}t = \int_{\mathbb{R}} |R_p(j)(t)| \bullet \int_{\mathbb{R}} |f^{i(x)}(x-t)| \mathrm{d}x\mathrm{d}t = \tag{2.19}$$

$$= \int_{\mathbb{R}} |R_p(j)(t)| \bullet \|f^{i(x)}\| \mathrm{d}t = \|f^{i(x)}\| \bullet \|R_p(j)\| < \infty. \tag{2.20}$$

Therefore, the hypothesis of Fubini's theorem is satisfied, so $f^i * R_p(j) \in L^1(\mathbb{R})$ from the thesis of Fubini's theorem. In addition, we obtain thanks to this theorem:

$$\|f^{i(x)} * R_p(j)\|_{L^1(\mathbb{R})} = \int | \int |f^{i(x)}(x-t)R_p(j)(t) \mathrm{d}v(t)| \mathrm{d}\mu(x) \tag{2.21}$$

$$\leq \int |f^{i(x)}(x-t)R_p(j)(t)| \mathrm{d}v(t)\mathrm{d}\mu(x) = \int |f^{i(x)}(x-t)R_p(j)(t)| \mathrm{d}\mu(x)\mathrm{d}v(t) \tag{2.22}$$

$$= \|f^{i(x)}\|_{L^1(\mathbb{R})} \|R_p(j)(t)\|_{L^1(\mathbb{R})} < \infty. \tag{2.23}$$

The equality (2.22) holds thanks Fubini's theorem, but (2.23) follows from the fact that both f^i and $R_p(j)$ are Lebesgue integrable, so their integral norms in $L^1(\mathbb{R})$ are finite. □

Up to know we exploited a possibility to normalize some convolutions. Now we state and prove that each convolution as a construction basis for a fuzzy Allen's relation is normalizable. This fact is reflected by the following theorem.

Theorem 3 *Convolution-basis of fuzzy Allen's relations are normalizable.*

Proof: It has already been show that $\int_{-\infty}^{\infty} f^{i(x)} R(t,j) dt < \infty$. Obviously, both $f^{i(x)}$ and $R(t,j)$ belong to $L^1(\mathbb{R})$ from definition. Thus also $\int_{-\infty}^{\infty} f^{i(x)} R(t,j) dt \in L^1(\mathbb{R})$ – thanks to Fubini's theorem and $\int_{-\infty}^{\infty} f^{i(x)} R(t,j) dt < \infty$ (see above). It means that there is an upper bound for it, say M. It is enough now to put a factor N to ensure that $0 \leq \frac{M}{N} \leq 1$. □

It also appears that fuzzy Allen's relations are representable by *uniformly continuous* functions.

Theorem 4 *Let $1 \leq p < \infty$, $f_i, R(j) \in L_p(\mathbb{R})$, $\|R(j)\| \leq M$, for some M where i_f is a characteristic function for a fuzzy interval i and $R(j)$ is a functional representation of a point-interval relation (of Allen type) with respect to a fuzzy interval j. Assume also that f_i is uniformly continuous on $\mathbb{R}$. Then $f_i \star R(j)$ is* `uniformly continuous` *on $\mathbb{R}$, too.*

Proof: Since f_i depends on $x - t$ and $R(j)$ dependent on t, let us establish $x - t = z$. It easy to see now that there is a $\rho > 0$ such that $z_1, z_2 \in \mathbb{R}$ and $\|z_1 - z_2\| < \rho$ implies $\|f_i(z_1) - f_i(z_2)\| < \epsilon$. However, f_i is assumed to be uniformly continuous in $\mathbb{R}$, i.e. there is a $\rho > 0$ such that for all $z_1, z_2 \in \mathbb{R}$, $|z_1 - z_2| < \rho$ implies $\|f_i(z_1 - f(z_2)\| < \epsilon$.

Then $\|z_1 - z_2\| < \rho$ also implies

$$\|i_f \star R(j)(z_1) - f_i \star R(j)(z_2)\|$$
$$= \Big(\int (|i_f(z_1) - f_i(z_2)| \bullet |R(j)(t)|)^p dt \Big)^{\frac{1}{p}}$$
$$= \|(f_i(z_1) - f_i(z_2)) \bullet R(j)\|_p \leq \|f_i(z_1) - f_i(z_2\| \bullet \|R(j)\|_p < \epsilon_1,$$

where $\epsilon_1 = M\epsilon$. Obviously, $\epsilon_1 \to 0$. The last inequality follows from the fact that $\|R(j)\|_p \leq M$ and from Schwartz's inequality $\|xy\| \leq \|x\|\|y\|$, for each $x, y \in L^p(\mathbb{R})$, $1 < p < \infty$. $\square$

The next theorem illustrates a computational power of the convolution-based approach. In fact, it forms a unique version of *Borel's Convolution Theorem* for convolutions used for defining fuzzy Allen's relations. It will be briefly called: 'Convolution Theorem for fuzzy Allen's relations'.

Theorem 5 ***(Convolution Theorem for fuzzy Allen's relations)****. Let i, j be fuzzy intervals. Let also $f^i, R(t,j) \in L^1[-\infty, \infty]$ be a function characterizing a fuzzy interval i and a point-interval Allen relation (resp.). Then their convolution $h = f^i * R$ has the following property.*

$$h(x) = \int_{-\infty}^{\infty} |f^{i(x)}(x-t) * R(j)(t)| \mathrm{d}t = \int_{-\infty}^{\infty} |f^{i(x)}(x-t)| \mathrm{d}x \int_{-\infty}^{\infty} |R(j)(t)| \mathrm{d}t. \tag{2.24}$$

Simply,

$$\mathcal{F}(f^i * R(j)) = \mathcal{F}(f^i)\mathcal{F}(R(j)). \tag{2.25}$$

Proof: Assume that $f^i, R \in L^1(\mathbb{R}^1)$ – as in the formulation of this theorem – are given. In order to compute their convolution

$$h(x) = \int_{-\infty}^{\infty} |f^{i(x)}(x-t) * R(j)(t)| \mathrm{d}t \tag{2.26}$$

we exploit Fourier transforms of f^i and R, i.e. $\mathcal{F}(f)$ and $\mathcal{F}(g)$ (respectively) – defined as follows:

$$\mathcal{F}(f^i) = \int_{\mathbb{R}} f^{i(x)}(x-t) e^{-2\pi ixv} \mathrm{d}x \quad \text{and} \tag{2.27}$$

$$\mathcal{F}(R) = \int_{\mathbb{R}} R(j)(t) e^{-2\pi ixv} \mathrm{d}t. \tag{2.28}$$

Therefore, for all $y \in \mathbb{R}$, Fourier transform $F(f * g)$ is as follows.

$$\mathcal{F}(f^i * R(j)) = \mathcal{F}(h)(x) = \mathcal{F}(\int_{-\infty}^{\infty} |f^{i(x)}(x-t) R(j)(t)| dt) = \tag{2.29}$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |f^{i(x)}(x-t) R(j)(t)| \mathrm{d}t e^{-2\pi ixy} \mathrm{d}x = \int_{-\infty}^{\infty} R(j)(t) \int_{-\infty}^{\infty} f^{i(x)}(x-t) e^{-2\pi ixy} \mathrm{d}x \mathrm{d}t. \tag{2.30}$$

By substitution $x - t = u$, or $x = t + u$ (and $\mathrm{d}x = \mathrm{d}u$) we obtain

$$\int_{-\infty}^{\infty} R(j)(t) \int_{-\infty}^{\infty} f^{i(x)}(x-t) e^{-2\pi ixy} \mathrm{d}x \mathrm{d}t = \int_{-\infty}^{\infty} R(j)(t) \Big(\int_{-\infty}^{\infty} f^{i(x)}(x-t) e^{-2\pi i(t+u)y} \mathrm{d}u \Big) \mathrm{d}t. \tag{2.31}$$

Applying Fubini theorem in order to interchange the order of limitation we can write:

$$\int_{-\infty}^{\infty} R(j)(t) \Big(\int_{-\infty}^{\infty} f^{i(x)}(x-t) e^{-2\pi i(t+u)y} \mathrm{d}u \Big) \mathrm{d}t = \tag{2.32}$$

$$= \int_{-\infty}^{\infty} R(j)(t) e^{-2\pi ity} \mathrm{d}t \int_{-\infty}^{\infty} f^{i(x)}(x-t) e^{-2\pi iuy} \mathrm{d}x = \mathcal{F}(f^i)\mathcal{F}(R(j)). \tag{2.33}$$

Therefore,

$$\mathcal{F}(f^i * R(j)) = \mathcal{F}(f^i)\mathcal{F}(R(j)). \tag{2.34}$$

It exactly means that

$$h(x)=\int_{-\infty}^{\infty}|f^{i(x)}(x-t)*R(j)(t)|\mathrm{d}t=\int_{-\infty}^{\infty}|f^{i(x)}(x-t)|\mathrm{d}x\int_{-\infty}^{\infty}|R(j)(t)|\mathrm{d}t=(2.21.).$$

Finally, let us undertake the question of majorization of fuzzy Allen's relations. It seems that an attempt of majorization might be carried out in terms of the approximations based on *Hardy-Littlewood Maximal Theorem.* This key idea of an approximation in terms of this theorem is as follows. At first, we take a locally integrable function, say f, and return for it another function, say f^*. This new function, at each point x of its domain gives the maximum average value, which f can have on balls centered at that point. More formally[10]:

$$f^*(x)=\sup_{r>0}\big\{\frac{1}{|B(x,r)|}\int_x^u fd\mu : u\in[x,\infty)\big\}. \tag{2.35}$$

It may be also defined:

$$f^{**}(x)=\sup_{r>0}\big\{\frac{1}{|B(x,r)|}\int_u^x fd\mu : u\in(-\infty,x]\big\}. \tag{2.36}$$

If f^* is defined as above for such a non-negative, Lebesgue measurable function on $\mathbb{R}$ that $\int_E f\mathrm{d}\mu$ on each compact set E(see: [99], pp. 422, 426), then Hardy-Littlewood Maximal Theorem asserts, for example, the following approximation ([99] p. 424):

$$\int_E f^*d\mu\le\frac{1}{k}\mu(E)+\frac{1}{1-k}\int f(x)\log^+ f(x)dx^{11}. \tag{2.37}$$

It appears that this theorem may be also formulated and proved(under some conditions) for fuzzy Allen's relations in their convolution-based depiction.

Theorem 6 ***Hardy-Littlewood Minimal Theorem for fuzzy Allen's relations***. *Let $f^iR(j)$ be an arbitrary convolution function, such that $\int_E i_fR(j)\mathrm{d}\mu<\infty$ for each compact E of $\mathbb{R}$ representing an Allen's relation for given fuzzy intervals i,j. Then for each k such that $0<k<1$ and for each convolution function, we have*

$$\int_E\Big(i_fR(j)\Big)^*d\mu\le\frac{1}{k}\mu(E)+\frac{1}{1-k}\int i_fR(j)\log^+ i_fR(j)dx, \tag{2.38}$$

$$\max\{\int_E\Big(i_fR(j)\Big)^*d\mu,\int_E\Big(i_fR(j)\Big)^{**}d\mu\}\le\frac{1}{k}\mu(E)+\frac{1}{2-k}\int i_fR(j)\log^+ i_fR(j)dx, \tag{2.39}$$

where $\big(i_fR(j)\big)^*$ and $\big(i_fR(j)\big)^{**}$ are defined due to 2.31 and 2.31.

Outline of the proof. Note that the right side of the inequality 2.34 'approximates', somehow, fuzzy Allen's relations from the left side of (2.34), so the thesis of the theorem asserts a majorization of it by the sum on the right side of the inequality 2.36. To prove this, it is enough to verify that $f^iR(j)$ respects the assumptions of Hardy-Littlewood Minimal Theorem to make use of the thesis of this theorem. (A detailed proof of the original version of Hardy-Littlewood Theorem may be found in [99], pp. 426-28.)

By assumptions, it holds: $\int_E i_fR(j)\mathrm{d}\mu<\infty$ for each compact E of $\mathbb{R}$. In addition, $f^iR(j)$ is a convolution, so it is non-negative and it is Lebesgue integrable (see: Lemma 1). It remains to show that $f^iR(j)$ is also Lebesgue measurable, i.e. $(\forall a\in\mathbb{R})\{x\in\mathbb{R}: a\le f^iR(j)(x)<\infty\}$ is a measurable set. Meanwhile $f^iR(j)$ is

[10] The similar definition may be found in: [99], pp. 422, 424

[11] $\log^+ t=\max\{logt,0\}$, for $0<t\infty$.

Lebesgue integrable in $\mathbb{R}$, and Lebesgue integral may be rendered – due to [99] (p. 164) – by the following operator L:

$$L(f^i R(j)) = \sup\Big\{ \sum_{k=1}^{\infty} \inf\{f^i(x)R(j)(x) : x \in A_k\}\mu(A_k) : \tag{2.40}$$

$$\text{each } A_i \text{ is measurable for } 1 < i < k \text{ and } \bigcup_{k=1}^{\infty} A_k = \mathbb{R}\Big\}, \tag{2.41}$$

where μ is a given Lebesgue measure. Thus, $f^i R(j)$ may be viewed as a functions defined on a finite family of measurable sets A_k covering the whole $\mathbb{R}$. It means that it is Lebesgue measurable. Thus $f^i R(j)$ satisfies all assumptions of the theorem. In this moment, the thesis follows from Hardy-Littlewood Minimal Theorem.
□

2.2.4 Algebraic Aspects of the Convolution-based Fuzzy Allen's Relations

Mathematical foundation of fuzzy Allen's relations in the convolution approach have just been expressed in terms of real and abstract analysis. Some naturality of such a solution is clear in the light of the fact that convolutions and Fourier transforms belong to the 'core' of real analysis. Nevertheless, Allen's algebra – as depicted in [57] forms a purely algebraic object. Therefore, a kind of an algebraic complementation of earlier results may be required.

Fuzzified Allen's Algebra *versus* Convolution Algebra? In the light of earlier results, one could venture to define a fuzzified version of Allen's Interval Algebra. Allen's Algebra is based on 13 basis relations between intervals with the composition and the inverse operation as admissible operations on these relations. Fuzzified Allen's Algebra (FAA) relies on an idea to represent fuzzy Allen's relations by norms of convolutions. The convolutions play a role of the 'surrogates' and they form convolutively-modified Ohlbach's fuzzy Allen's relations. Formally, FAA is defined as follows.

Definition 18 *Assume that i, j are (arbitrary) fuzzy intervals and $f^i, g^i \in L(\mathbb{R}^1)$ are functions characterizing these intervals. Let also $\| \bullet \|$ be a norm in $L(\mathbb{R}^1)$.* ***`Fuzzified Allen's Algebra`*** *(FAA) is defined as a tuple of the form:*

$$FAA = \langle \mathcal{A}, +_{\mathcal{A}}, \bullet_{\mathcal{A}} \rangle, \tag{2.42}$$

where:

$$\mathcal{A} = \{\|C\| : C = f^i * g^j, \quad \text{for } \ f^i, g^j \in L(\mathbb{R}^1)\} \tag{2.43}$$

and $+, \bullet$ are operations defined as follows:

$$(\| \ \|_1 +_{\mathcal{A}} \| \ \|_2)(C) = \|C\|_1 +_{\mathcal{A}} \|C\|_2, \quad \|\alpha \bullet_{\mathcal{A}} C\| = |\alpha| \bullet_{\mathcal{A}} \|C\|. \tag{2.44}$$

It arises a natural question of algebraic features of FAA. Many of then may be inferred from the fact that FAA as defined in (2.37)-(2.39) forms a vector space. The following theorem asserts this fact.

Theorem 7 *$FAA = \langle \mathcal{A}, +_{\mathcal{A}}, \bullet_{\mathcal{A}} \rangle$ with $+_{\mathcal{A}}, \bullet_{\mathcal{A}}$ defined by 2.41 forms a vector space. In particular, there exist the neutral and the inverse elements of both $+_{\mathcal{A}}$ and $\bullet_{\mathcal{A}}$.*

Proof: To show the thesis of theorem, it is enough to find the appropriate interpretation $I : \mathcal{A} \to \mathbb{R}$, which 'interprets' operations on $\mathcal{A}$ for operations in $\mathbb{R}$. In fact, the homomorphism $H : \mathcal{A} \to \mathbb{R}$ such that, for any $C_1, C_2 \in L(\mathbb{R})$:

$$H\Big(\|C_1\| +_{\mathcal{A}} \|C_2\|\Big) = H(\|C_1|) +_{\mathbb{R}} H(\|C_1|), \tag{2.45}$$

$$H\Big(|\alpha| \bullet_{\mathcal{A}} \|C\|\Big) = |\alpha| \bullet_{\mathbb{R}} H(\|C\|) \tag{2.46}$$

is already a required homomorphism. Obviously, there exist the neutral and the inverse elements of both operations as homomorphic images of the corresponding neutral and inverse elements in $\mathbb{R}$. In particular,

$$H(\mathbf{0}) = \mathbf{0}_{\mathcal{A}}, H(\mathbf{1}) = \mathbf{1}_{\mathcal{A}}. \qquad \square$$

Let us alternatively introduce a similar algebraic structure, somehow, related to Allen's algebra.

Definition 19 *Define the set*

$$|\mathcal{A}| = \{C_i : C_i f_i * g_i, \text{ for } f^{i(x)}, g_i \in L(\mathbb{R}^1)\}, \tag{2.47}$$

where each C_i forms a basis for a fuzzy Allen's relation R^{fuzzy} (i.e. the standard $L(\mathbb{R})$-norm of C_i represents such a fuzzy Allen's relation R^{fuzzy}). Then the structure:

$$\mathcal{A} = \langle |\mathcal{A}|, * \rangle, \tag{2.48}$$

, where $*$ is a convolution operator, is to be called `Convolution Allen's Algebra` *(CAA). If $\mathcal{A}$ is given, the set $|\mathcal{A}|$ forms a universe of CAA.*

It appears that CAA does not form so algebraically 'nice' structure as it contains no multiplicative unit in the sense of $*$-operation. This proof adopts1 the proof reasoning for Banach algebras defined on $L^1(\mathbb{R})$ from [99], pp. 76, 399.

Theorem 8 *The Convolution Allen's Algebra $\mathcal{A} = \langle |\mathcal{A}|, * \rangle$ with $*$-operations has no multiplicative unit in the sense of $*$-operator, i.e. there is no function $e \in |A|$ such that $e * f = f * e = f$, for each $f \in |\mathcal{A}|$.*

Proof: Assume that $\mathcal{A}$ has a multiplicative unit u, i.e, $u \in \mathcal{A}$ and $u * f = f$ almost everywhere (i.e. we admit that does it not hold for sets of a Lebesgue measure 0) for $f \in \mathcal{A}$. The 'core' of the proof is to obtain a contradiction. We will try to show that:

A On one hand, there exists such a real number $\delta > 0$ such that $\int_{-2\delta}^{2\delta} |u(t)| \mathrm{d}t < 1$.

B On the other hand, $1 \leq \int_{-2\delta}^{2\delta} |u(t)| \mathrm{d}t$.

Prove the first fact **A**. For that reason, assume that $u \in L^1(\mathbb{R}, \sigma, \mu)$. The fact **A** will be shown if we show that for every $\epsilon > 0$ there exist such a $\rho > 0$ that for all sets $E \in \sigma$ satisfying $\mu(E) < 0$, we have:

$$\int_E |u| \mathrm{d}\mu < \epsilon. \tag{2.49}$$

Establish a function sequence $\phi_n(t) = |u(t)|$, for $|u(t)| \leq n$ and $\phi_n(t) = n$, otherwise. Obviously, $\phi_n(t)$, as a nondecreasing sequence of σ-measurable functions, converges to $|u|$, i.e. $\lim_{n\to\infty} \phi_n = |u|$. By Bepo-Levi's Theorem we get:

$$\lim_{n\to\infty} \int_E \phi_n \mathrm{d}\mu = \int_E \lim_{n\to\infty} \phi_n \mathrm{d}\mu = \int_E |u| \mathrm{d}\mu. \tag{2.50}$$

It allows us to select such n that, for E that $\mu(E) < \rho$, the following hold:

$$\int_E \phi_n \mathrm{d}\mu \leq \int_E n = n\mu(E) < \frac{1}{2}\epsilon \text{ and } \int_E (|u| - \phi_n) \mathrm{d}\mu < \frac{1}{2}\epsilon, \tag{2.51}$$

if put $\rho = \frac{\epsilon}{2n}$.Thus,

$$\int_E |u| \mathrm{d}\mu = \int_E (|u| - \phi_n) \mathrm{d}\mu + \int_E \phi_n \mathrm{d}\mu < \int_E (|u| - \phi_n) \mathrm{d}\mu + \frac{1}{2}\epsilon < \epsilon. \tag{2.52}$$

Indeed, there really exists such a real number $\delta > 0$ such that $\int_{-2\delta}^{2\delta} |u(t)|\mathrm{d}t < 1$.

Move to the proof of **B**. Let be $f = \xi_{[-\delta,\delta]}$ now. Then $f \in L(\mathbb{R}^1)$ and for almost all $x \in \mathbb{R}$ we have

$$f(x) = u * f(x) = \int_{\mathbb{R}^1} u(x-y)f(y)\mathrm{d}y \tag{2.53}$$

$$= \int_{-\delta}^{\delta} u(x-y)dy = \int_{x-\delta}^{x+\delta} u(t)\mathrm{d}t. \tag{2.54}$$

Since a measure $\mu([-\delta,\delta]) > 0$, there must exist such an $x \in [-\delta,\delta]$ that

$$1 = f(x) = \int_{x-\delta}^{x+\delta} u(t)\mathrm{d}t. \tag{2.55}$$

On the other hand, $[x-\delta, x+\delta] \subset [-2\delta, 2\delta]$, it implies that

$$1 = |\int_{x-\delta}^{x+\delta} u(t)\mathrm{d}t| \leq \int_{x-\delta}^{x+\delta} |u(t)|\mathrm{d}t \leq \int_{-2\delta}^{2\delta} |u(t)|\mathrm{d}t < 1. \tag{2.56}$$

It generates the expected contradiction what proves the theorem. □

Theorem 7 and Theorem 8 elucidate some interesting facts about FAA and CAA. Namely, FAA – as a more algebraic structure – remains more complete in a unique sense, i.e. having neutral and inverse elements in the sense of the operations defined in it. By contrast, CAA as an algebraized fragment of $L(\mathbb{R})$-space does not share the same 'nice' property. In fact, there is no multiplicative(convolution) unit for each function from the the CAA-univese.

2.3 Modeling of Fuzzy Temporal Constraints and Preferences

In Section 2.2, fuzzy Allen's relations in a new conceptual framework were introduced and discussed. In the current section, fuzzy Allen's relations are used to construct a more cumulative approach to Fuzzy Temporal Constraints. More precisely, fuzzy Allen's relations will be combined with the quantitative temporal constraints imposed on Multi Agent Problem – as defined in Chapter 1. This combination is intended to constitute a basis for a new approach to modeling of preferences.

Recently, fuzzy Allen's relations were recently proposed with no reference to a subject-specification of temporal planning with different constraints. By contrast, fuzzy temporal constraints and preferences, defined in this subsection, will be discussed in the context of Multi Agent Problem from Chapter 1.

2.3.1 Fuzzy Temporal Constraints in a New Representation

The idea to propose a new representation for fuzzy temporal constraints stems from the following reasoning and observation. Assume that MAP[12] with the associated: set of agents N and of actions $\mathcal{A}$. Consider now a pair (d,z), where d represents a temporal compact interval in $\mathbb{R}$ (day) and z, such that $z \subset d$, represents an subinterval of it (shift). (A point $x \in (d,z) \iff x \in z \subset d$). Intentionally, a pair (d,z) represents a working shift z of a day d in MAP. More precisely, $(d,z)^a_n$ denotes a shift z of a day d of a given agent $n \in N$ with respect to a given action $a \in \mathcal{A}$. Since, as earlier mentioned, a normalized (cost, effectiveness, etc.) function F may be associated to each such a pair $(d,z)^a_n$, we can also consider the extended pairs of the form:

[12]For simplicity, we will use an abbreviation MAP for a denotation of the Multi-Agent Schedule-Planning Problem in this chapter and in chapter 3, when it does not lead to any confusion.

$$\Big((d,z)^n_a, F((d,z)^n_a)\Big). \tag{2.57}$$

F-functions are usually rendered by normalized sums of $\{0,1\}$-valued characteristic functions $X(i)^{n,a}_{d,z}$, where each $X(i)$ specifies a separate aspect of n-activity (effectiveness, ability, motivation, etc.). As result, we are in position to consider the extended pairs of the following form:

$$\Big((d,z)^n_a, \sum_i X(i)^{n,a}_{d,z}\Big). \tag{2.58}$$

Whereas a pair $(d,z))^n_a$ may be seen as an interval in $\mathbb{R}$ (x-axis), the normalized $\sum_i X(i)^{n,a}_{d,z}$ takes values from [0,1] (y-axis), so the whole pair (2,49) is in $\mathbb{R}^2$. Each of (2.49) may be represented by a vertical polygon as $\sum_i X(i)^{n,a}_{d,z}$ is constant in (d,z). All of such pairs (2.49) (vertical polygons) form the 'core' of a fuzzy interval – as depicted in Fig.2.2. This fact approximate the fact fuzzy temporal constraints may be encoded by fuzzy intervals. In fact, we need two fuzzy intervals instead of a single one (It will be explained in detail below.)

Meanwhile, the exact chronology of defining fuzzy temporal constraints consists of the following steps:

Step1: Encoding quantitative temporal constraints of MAP by pairs $((d,z), F(d,z))$ of two fuzzy temporal intervals:

- the first fuzzy interval, say I_1, will represent the real situation (of planning and scheduling in Multi-Agent Problem),
- the second one, say I_2, will represent the required planned situation.

Step2: Indicating the common region of these fuzzy intervals representing (hard) temporal constraints imposed on Multi-Agent Problem.

Step3: Defining a mutual relation between these fuzzy intervals in terms of a fuzzy Allen's relation (as a new qualitative temporal constraints on them).

Step4: Identifying the new *fuzzy temporal constraints* with the normalized convolutions representing the fuzzy Allen's relation between the two intervals.

Briefly, these ideas may be visualized as depicted on a diagram – Figure 2.1.

Steps 1 and 2, or Fuzzy Temporal Constraints for Multi-Agent Problem

In order to show how quantitative temporal constraints may be encoded by fuzzy intervals in detail, return to the earlier observation that quantitative temporal constraints of MAP may be represented by pairs: $\Big((d,z)^n_a, \sum_i X(i)_{n,d,z,a}\Big)$, for $d \in D = \{d_1, d_2 \ldots, d_k\}$, $z \in Z = \{z_1, z_2 \ldots, z_l\}$, $n \in N, a \in \mathcal{A})$ – where:

- the pair $(d,z)^n_a$ represents a shift z of a day d indexed by an action a and an agent (nurse) n associated to (d,z)[13] and
- $\sum X(i)_{n,d,z,a}$ represents a quantitative temporal constraint imposed on $(d,z)^n_a$, which may be normalized such that $0 \leq \sum_i X(i)_{n,d,z,a} \leq 1$, as described in chapter 1 of 'Contribution'.

It was also mentioned that each such a pair is represented by a vertical polygon in the 'core' of a fuzzy interval. It remains to define the initial and the final parts of this interval. The next, a characterizing functions for such a fuzzy interval must be given. In the light of Definition 18 (section 4.1 of 'Introduction') such a function

[13] As earlier we assume that $(d_i, z_j) \neq (d_k, z_l)$ are disjoint for $i \neq k$ and $j \neq l$.

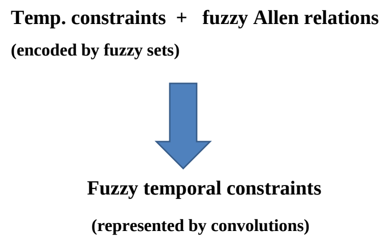


Figure 2.1: The main idea of defining fuzzy temporal constraints and preferences.

must be Lebesgue integrable (so not necessary continuous). However, a diagram of this function cannot be completely arbitrary. In fact, it should approximate a trapezoidal form of fuzzy intervals. It means that this function should linearly increase in the initial parts of a fuzzy interval and linearly decreases in the final part.

Fuzzy intervals and their characteristic functions. The construction of this function may run as follows. Initially, consider a support-function $\phi : \mathbb{R} \to \mathcal{S}$, for $\mathcal{S} = \{(d,z)^n_a : d \in D, z \in Z\}$ and such that $\phi(x) = (d,z)^n_a$. Informally, ϕ associates each real point to a day d and a shift z of d, for a given agent n and action a.

Secondly, let us define now the set $\mathcal{C} = \{c = \sum_{i=1}^{k} X(i)^{n,a}_{(d,z)} : \frac{\sum_{i=1}^{m} X(i)^{n,a}_{(d,z)}}{N} \in [0,1]\} \cup \{ax + b : a, b \in \mathbb{R}\}$ containing constant functions c's as normalized sums $\sum_{i=1}^{k} X(i)^{a,n}_{(d,z)}$ and linear functions.

The next, let us define a new function $F : \mathcal{S} \to \mathcal{C}$, for $\mathcal{S}$ and $\mathcal{C}$ as above, defined by the clause:

$$F((d,z)^a_n) = \begin{cases} A(x) & \text{if } x \in (d_1, z_1)^a_n \\ -B(x) & \text{if } x \in (d_k, z_l)^a_n \\ c = \frac{1}{N}\sum_{i=1}^{m} X(i)_{n,d,z,a} & \text{otherwise.} \end{cases} \tag{2.59}$$

The sense of this function may be explained in the following way. We take a pair (d,z) and associate to $(d,z)^n_a$ a function of x-argument dependently on the situation. If x belongs to the initial interval $(d_1, z_1)^a_n$, we take $A(x)$, for $A > 0$, so Ax is linearly increasing. If x belongs to the final interval $(d_k, z_l)^a_n$, we take $-B(x)$, for $B > 0$, so the function is linearly decreasing in this interval. For each other interval, we take a constant function c being a normalized sum of $\{0,1\}$-valued characteristic functions $X_{(d,z)}$, for a given agent n and action a – as defined in Chapter 1.

Example 25 . *Assume that a pair $(d,z) = (monday, night)$, for a given agent n and some action a = 'to be on duty,' is given. Assume that a characteristic function $X(1)^{n,a}_{(d,z)} = 1$ defines a physical presence of n in the work place and $X(2)^{n,a}_{(d,z)} = 0$ defines a real ability to perform a. By a normalization factor, say $N = 5$, we get the sum $c = \frac{1}{5}\sum_{i=1}^{2} X(i)^{n,a}_{(d,z)} = \frac{1}{5}(1+0) = \frac{1}{5}$.*

Finally, one can define a new function $f : \mathbb{R} \to \mathcal{C}$ defined as the composition: $f = F \circ \phi$.

$$f(x) = \begin{cases} A(x) & \text{if } x \in (d_1, z_1)_n^a \\ -B(x) & \text{if } x \in (d_k, z_l)_n^a \\ c = \frac{1}{N}\sum_{i=1}^{m} X(i)_{n,d,z,a} & \text{otherwise.} \end{cases} \tag{2.60}$$

Exploiting this terminological framework, let us introduce the set

$$\mathcal{S}^* = \Big\{ \Big((d,z)_a^n, F(d,z)_a^n \Big) : 0 \leq \Big| (F(d,z)_a^n) \Big| \leq 1 \Big\}, \tag{2.61}$$

where $\Big| F((d,z)_a^n) \Big|$ denotes a value of $F((d,z)_a^n)$, for the established a, n. It is not difficult to note that $\mathcal{S}*$ algebraically defines a *fuzzy interval* in the sense given by Definition 18. The following corollary confirms this hypothesis.

Colollary 2 *The set* $\mathcal{S}^* = \Big\{ \Big((d,z)_a^n, F((d,z)_a^n) \Big) \text{: } 0 \leq \Big| F((d,z)_a^n) \Big| \leq 1 \Big\}$ *– with* $F((d,z)_a^n)$ *defined in (2.50) – algebraically defines a fuzzy interval (in a sense given by Definition 18) with a finite support.*

Proof: In order to show that $\mathcal{S}$ defines a fuzzy set with a finite support– in the light of Definition 18 – one needs to prove the following facts:

1. the function $F((d,z)_a^n)$ is Lebesgue integrable,
2. $0 \leq \Big| F(d,z)_a^n) \Big| \leq 1$ and
3. $\sum_{1 \leq j \leq l}^{1 \leq i \leq k} \Big| (d_i, z_j) \Big| < \infty$ (sum of lengths of all intervals (d_i, z_j) must be finite.)

The property (1) follows from a definition of $F((d,z)_a^n)$. In fact, this function may be rendered either as $\sum_{i=1}^{m} X(i)_{n,d,z,a}$ or as the appropriate linear functions. In the first case, it is Lebesgue integrable as a sum of Lebesgue integrable characteristic functions. In the second case, this function is Lebesgue integrable as a continuous function and Riemann integrable. The property (2) immediately follows from a definition of $\mathcal{S}^*$. Finally, (3) follows from the fact that $|(d_i, z_j)|$ is always bounded (for example, by $M = \max\{|(d_i, z_j)|$, for $1 \leq i \leq k$ and $1 \leq j \leq l\}$ and a number of these intervals is finite ($= i$). Hence, $\sum_{1 \leq j \leq l}^{1 \leq i \leq k} \Big| (d_i, z_j) \Big| < \infty$. It finishes the proof. □

The geometric form of S^* is similar to the fuzzy interval as visualized in Figure 2.2.

In this way, a fuzzy interval, say $i(x)$ was introduced. It encodes a piece of information about fuzzy temporal constraints that must be satisfied to perform actions from $\mathcal{A}$ by agents from $\mathcal{N}$. A role of the second interval is to encode a piece of information about fuzzy temporal constraints that *really* were satisfied by performing of actions from $\mathcal{A}$ by agents from $\mathcal{N}$.

Obviously, the second fuzzy interval may be achieved in a similar way. In fact, a new support function $\phi : \mathbb{R} \to \mathcal{S}$ may be introduced as a basis for a new function $F : (d,z) \in \mathcal{S} \to g \in \mathcal{C}$, where g is either a normalized sum or a linear function, defined as earlier. Finally, the appropriate function $f : \mathbb{R} \to \mathcal{C}$ as the composition $F \circ \phi$ can be defined. In order to distinguished functions: ϕ, f and F for i from the corresponding functions for j, we could denote both families of functions as follows: ϕ^i, f^i and F^i and ϕ^j, f^j and F^j (resp.). Observe that F^j and F^i are identical for a (non-empty) common region of both intervals i, j. Let us denote the restriction of both F^i and F^j to their common region by $F^{\triangle}$. Formally:

$$F^{\triangle} = F^i|_{i \cap j} = F^j|_{i \cap j}. \tag{2.62}$$

This common restriction $F^{\triangle}$ enables of defining a special type of fuzzy temporal constraints – *the hard (quantitative) temporal constrains* (HTC). Recall that such constraints must be necessary satisfied to perform the required actions or to achieve a desired goal by agents. It means that HTC are not only

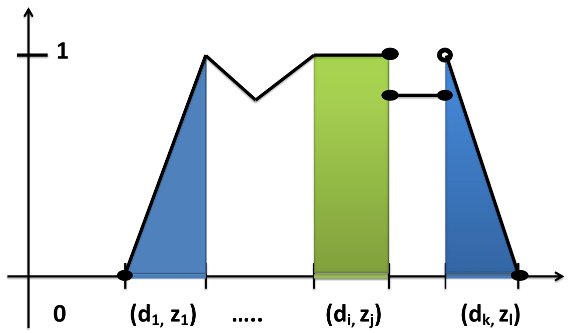


Figure 2.2: Intervals (d_i, z_j) and function $f((d_i, z_j)$ (dark line on a picture) determining a fuzzy interval. In fact, this picture illustrates a more general situation, when $\sum_{i=1}^{m} X(i)_{d,z,n,a}$ is not necessary a constant function and f is not necessary continuous.

required or planned to be satisfy, but they must be *really* satisfied. In a geometric interpretation: they belong to the common region of both intervals i and j, as just defined above. In a formal algebraic treatment, they may be defined as follows.

Definition 20 ***Hard (Quantitative) Temporal Constrains*** *are determined by the following set*

$$HTC = \Big\{p = \Big((d,z)_a^n, F^{\triangle}((d,z)_a^n)\Big) : p \in i(x) \cap j(x)\Big\} \tag{2.63}$$

for fuzzy intervals $i(x)$ and $j(x)$ – obtained as above and algebraically described by (2.52).

If HTC was an empty set, it would mean that no hard temporal constraint is satisfied (in a given situation of MAP). An exemplary visual representation of HTC is presented in Fig. 2.3.

It finally remains a question of a fuzzy nature of HTC – so important for the current analysis. One could venture to state that HTC introduce a piece of fuzziness. In fact, temporal constraints on MAP are normalized and they built trapezoidal fuzzy intervals i and j. Nevertheless, it constitutes a kind of *implicit* fuzziness. The next subsection is aimed at introducing a piece of *explicit* fuzziness *via* fuzzy Allen's relations – as defined in Section 2.2.

Steps 3 and 4, or Fuzzy Allen's relations imposed on HTC

Observe that both fuzzy intervals i and j were built up from temporal constraints on MAP in a mutually independent way. Only their common region was indicated to define HTC in the final step of that procedure. Now, let us assume that we want to describe a mutual relation between intervals i and j in terms of a fuzzy Allen's relations. Since both intervals must have a common region, it introduces a unique 'relevancy hierarchy' between them: 'meet'-relation is admissible, but 'during' or 'overlap' is more appropriate because their convolutions 'promote' the common regions of both intervals more than the 'meet'-convolution.

Independently of the chosen fuzzy Allen's relation, they are considered here as given in the convolution-based representation, as proposed in Section 2.2. Consider the same MAP with a given set of actions $\mathcal{A}$ and agents $\mathcal{N}$. Assume also that the following entities are given:

1 *fuzzy intervals* i and j determined by the functions: f^i, F^i and f^j, F^j (resp.) with Hard Temporal Constraints on MAP determined by the set

$$HTC = \Big\{p = \Big((d,z)_a^n, F^{\triangle}((d,z)_a^n)\Big) : p \in i(x) \cap j(x)\Big\},$$

2 and *fuzzy Allen's relation 'meet'* in a convolution-based depiction that represents a mutual relation between i and j (see: Fig. 2.3).

Since 'meet'-relation is here a *fuzzy* Allen's relation, it geometrically covers not only the common region of both intervals (HTC), but also a region in a neighborhood of it. This situation is depicted in Fig. 2.3. The 'acceptable' region is broader than the common blue triangle $\triangle = i \cap j$ and it is determined by the broken convolution-line (the pink area in Fig. 2.3). To describe this acceptable region formally, it is convenient to make use of Ohlbach's approach and to describe fuzzy 'meet'-relation in terms of the functions: $Fin(i)$ and $St(j)$. Recall that $Fin(i)$ cuts a final part of i and $St(j)$ cuts the initial one of j. In terms of $Fin(i)$ and $St(j)$, we could describe the acceptable pink region as bounded by diagrams of these functions and x-axis (see: Figure 2.3). Since this region contains not only the common triangle $\triangle = i \cap j$, we should use F^i (for points of i) and F^j (for points of j) instead of $F^{\triangle}$. It allows us to define *Fuzzy Temporal Constraints based on 'meet'-relation* as follows.

Definition 21 *Assume that the Allen's 'meet' relation is given in its convolution-based depiction. Then* ***`Fuzzy temporal constraints based on 'meet'-relation`*** *is determined by the following set*

$$FTC = \Big\{\Big((d,z)_a^n, F^i(d,z)_a^n)\Big) \in i(x) \text{ and } F^i(d,z)(x) \leq Fin(i)(d,z)(x)\Big\} \tag{2.64}$$

$$\bigcup \Big\{\Big((d,z)_a^n, F^j(d,z)_a^n)\Big) \in j(x) \text{ and } F^j(d,z)(x) \leq St(j)(d,z)(x)\Big\}. \tag{2.65}$$

for fuzzy intervals i and j.

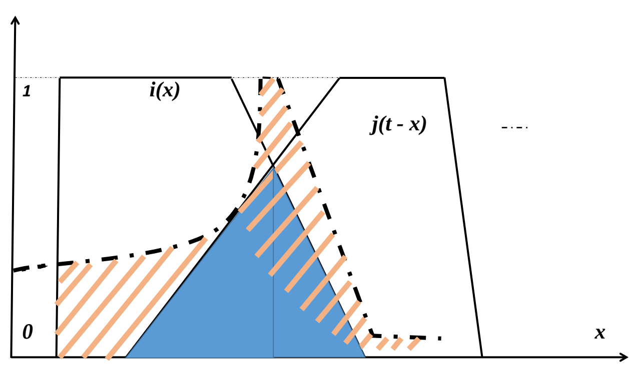


Figure 2.3: Fuzzy temporal constraints (FTC) determined by the convolution for Allen's 'meet'-relation over a common (blue) region HTC.

Example 26 . *Consider an agent N performing regularly a single action* `writing` *each day and each shift. Assuming that a rhythm of its job and temporal constraints imposed on its action give two fuzzy intervals i and j such that their meet-relation is determined by linear functions $St(i)(x) = -x$ and $Fin(j)(x) = 3x$.*

Then the FTC are given as follows:

$$FTC = \Big\{ \Big((d,z)^N_{writing}, F^i((d,z)^N_{writing}) \Big) \in i(x) \text{ and} \tag{2.66}$$

$$\Big((d,z)^N_{writing}, F^j((d,z)^N_{writing}) \Big) \in j(x) \text{ and} \tag{2.67}$$

$$F^i(d,z)^N_{writing}(x) \leq -x \text{ and } F^j(d,z)^N_{writing}(x) \leq 3x \Big\} \tag{2.68}$$

Dependently on the used Allen's relation – we will exploit a name of *fuzzy temporal constraint based on 'meet'-relation*, *fuzzy temporal constraint based on during-relation* or of *fuzzy temporal constraint based on during-relation*.

2.3.2 Preferences in a New Representation.

The same reasoning may be exploited to propose a new representation for *preferences*. The term 'preferences' will be interpreted now as a *preferred area*, as a *set of pairs* $((d,z)^n_a, F(d,z)^n_a\Big)$, where $(d,z)^n_a, F(d,z)^n_a$ are defined above.

In order to approximate this new representation, consider namely that fuzzy (hard) temporal constraints (HTC) have already been defined relatively to 'during'-relation – as depicted on Figure 2.4. Consider also that a line k – not necessary parallel to x-axis – is given. Obviously, this line cuts some uniformly pink region – as represented on the same Figure 2.4. The common part of this pink region and fuzzy intervals $i(x)$ and $j(x)$ consists of all points of these pairs $\Big((d,z)^n_a, F(d,z)^n_a\Big)$ which 'lie' above the line k.

If we consider all points of both intervals lying above the line k as the *preferred* ones[14], then all such points of pairs $\Big((d,z)^n_a, F(d,z)^n_a\Big)$ of the pink region can be interpreted as representing *a global preference* of Multi-Agent Problem. Since these preferences have a more global nature in a comparison to the 'local' agent preferences, they may be called '*the global preferences*' (of MAP). Taking, for simplicity, the pairs $(x, f(x)) \in \mathbb{R}^2$ instead of $\Big((d,z)^n_a, F(d,z)^n_a\Big)$ if only $x \in (d,z)^n_a$, we can obtain the following definition of *global preferences*[15].

Definition 22 ***The Global Preference*** *is determined by the set* $\mathcal{P} = \{(x, f(x)) \in \mathbb{R}^2 : \texttt{Pref(x)} \leq f(x) \leq C(x)$ *and* $\texttt{Pref(x)} \in [0,1]$ *is an arbitrary continuous function – called* **`the preferential`** *and* $C(x)$ *is a normalized convolution.*

At the end of the current analysis, we venture to propose the following definition of a schedule-plan in terms of the proposed approach.

Definition 23 ***The Schedule-Plan*** *(for Multi-Agent Problem) is each finite sequence of pairs – parametrized by agents from* $\mathcal{N}$ *and actions* $\mathcal{A}$ *– of the form:* $\{(d,z)^{a_1}_{n_i}, (d,z)^{a_1}_{n_i}, \ldots, (d,z)^{a_{fin}}_{n_i}\}$ *for* $n_i \in N$ *for* $i \in I$ *and* $a \in \mathcal{A}$*, which satisfy hard quantitative temporal (HTC) constraints imposed on this problem.*

The schedule-plan is ***optimal*** *if it satisfies both HTC and Global Preferences of Multi-Agent Problem, i.e. if all* F*-values for parametrized arguments* $(d,z)^{a_j}_{n_i} \geq \texttt{Pref(x)}$*, where* `Pref` *is a given preferential line.*

Example 27 *Consider a 2-day schedule* $\mathcal{H}$ *of agent N's job:*

$$\mathcal{H} = \{(\text{Mon}, \text{morning}), (\text{Mon}, \text{afternoon}), (\text{Tue}, \text{morning}), (\text{Tue}, \text{afternoon})\}$$

[14] We are not interested now in a criterion of this preference.

[15] Obviously – as signalized – we still think about preferences in a sense of a set of preferred points here. We *do not* understand them as relations or as pragmatic intentions of a human, of an agent, etc. It seems that computational tools are not any appropriate tools to grasp these two non-'measurable' sides of preferences.

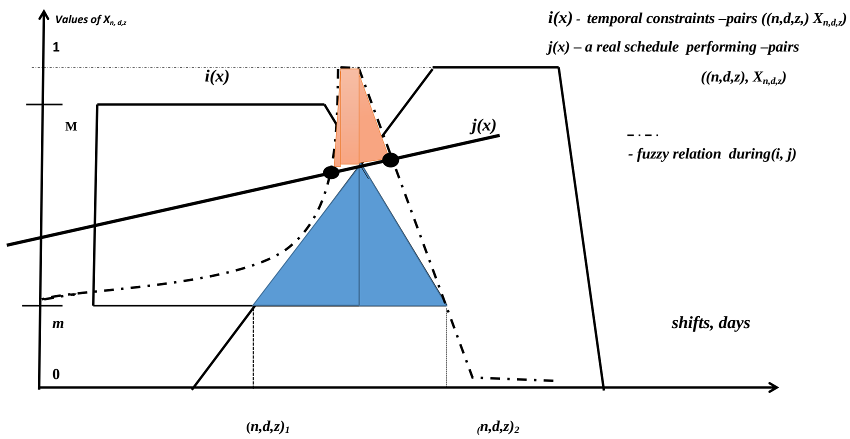


Figure 2.4: The blue common triangle represents HTC. FTC are represented by a fuzzy region 'around' the triangle – determined by the broken line. Finally, preferences are represented by a common region of both intervals and the pink region cut by the preferential linear function y (the dark line). The interval $i(x)$ encodes (the constraints of) the planned situation. The normalized values of F-functions are found on the vertical axis.The interval $j(x)$ encodes the real situation.

associated to the sequence of planned actions $Pl = \{$`opening, writing, monitoring, closing`$\}$.
Assume also that F-function is now an effectiveness function of N's work alway taking (after the appropriate normalization) values $> 0,5$, but the preferred values of them should be greater than 0,7.

Consider now two sets of pairs of days and shifts with values of an effectiveness function associated to them:

$$\begin{aligned}\mathcal{H}_1 = \Big\{\Big(&(\text{Mon}, \text{morning})_N^{\text{open}}, (\text{Mon}, \text{afternoon})_N^{\text{write}}, (\text{Tu}, \text{morning})_N^{\text{monitor}}, (\text{Tu}, \text{afternoon})_N^{\text{close}}\Big) : \\ &F((\text{Mon}, \text{morning})) = 0,71, F(\text{Mon}, \text{afternoon}) > 0,8, \\ &F(\text{Tu}, \text{morning})) = 0,72, F(\text{Tu}, \text{afternoon}) = 0,77\Big\}\end{aligned}$$

and

$$\begin{aligned}\mathcal{H}_2 = \Big\{\Big(&(\text{Mon}, \text{morning})_N^{\text{open}}, (\text{Mon}, \text{afternoon})_N^{\text{write}}, (\text{Tu}, \text{morning})_N^{\text{monitor}}, (\text{Tu}, \text{afternoon})_N^{\text{close}}\Big) : \\ &F((\text{Mon}, \text{morning})) = 0,51, F(\text{Mon}, \text{afternoon}) > 0,8, \\ &F(\text{Tu}, \text{morning})) = 0,62, F(\text{Tu}, \text{afternoon}) = 0,77\Big\}\end{aligned}$$

Due to Definition 26 – both $\mathcal{H}_1$ and $\mathcal{H}_2$ can be considered as the admitted schedule-plans. However, only $\mathcal{H}_1$ respects all the temporal constraints and preferences (all values of function f are greater than 0,7), so it may be considered as an `optimal` *schedule-plan.*

2.4 Fuzzy Temporal Constraints and Preferences in STRIPS and Davis-Putnam Procedure

The conceptual apparatus, elaborated in 2.2 and 2.3, will be exploit in temporal planning now. We intend to refer this apparatus to temporal planning: as determined by STRIPS-method and as based on Davis-Putnam procedure[16]. The appropriate temporal-preferential extensions of both methods is proposed in this section. At first, some temporal-preferential extension of STRIPS-method is introduced in 2.4.2, called later: TP-STRIPS. The analog temporal-preferential extension of Davis-Putnam procedure is proposed in 2.4.3.

2.4.1 Fuzzy Temporal Constraints and Preferences in STRIPS

STRIPS – a brief repetition. As mentioned in Part I of Introductions (pp. 7-9), STRIPS method was conceived by N. Nilsson as a unique forward search procedure. Beginning from the initial state s_0 to achieve a goal g it works as follows.

1. This algorithm works if a set of *states* is not empty,
2. Then we choose a state $s \in$*states*.
   - If $g \subseteq s$, then we take $\pi(s)$ as a desired plan[17].
   - Otherwise, we take a set $E(s)$ of actions applicable to s.
     (a) if $E(s)$ is empty – we remove s from *states*,
     (b) if does not – we choose an action $a \in E(s)$ and exchange s for $s^{'}$ by removing effects of a from s. Then a current plan $\pi(s^{'}) = \pi(s).a$ (The action a is added at the end.)
     (c) The same procedure is repeated for a set $E(s^{'})$, etc. until g will be achieved.

Elements of this procedure may be represented in the STRIPS-algorithm as follows ($\texttt{Effect}^{-}$ and $\texttt{Effect}^{+}$ denote negative and positive effects of actions a. For a detailed definition – see: [3]).

`STRIPS-algorithm`($\mathcal{A}, s_0$, g)
begin
states= $\{s_0\}$
$\pi(s_0) = \langle\rangle$
  $E(s_0) = \{a|$ a is a ground instance of an operator in $\mathcal{A}$ and `precond(a)`$\subseteq s_0\}$
  while true **do**
   if `states` $= \emptyset$ **then return** `failure` **end if**
   choose a state $s \in$ `state`
   if $g \subseteq s$ **then return** $\pi(s)$ **end if**
   if $E(s) = \emptyset$ **then**
     remove s from *states*
  else
    choose and **remove** an action $a \in E(s)$
    $s^{'} \leftarrow$ s/ $\texttt{Effect}^{-}(a) \cup \texttt{Effect}^{+}(a)$
    if$s^{'} \notin$ *states* **then**
      $\pi(s^{'}) = \pi(s).a$
      $E(s^{'}) = \{a|a$ is a ground instance of an operation $\in \mathcal{A}$ and `precond(a)`$\subseteq s^{'}\}$
    end if
  end if
end

[16]The discussed issues may be also found in [100, 101] and were presented during the ICAISC2017 in Zakopane in June 2017.
[17]We can see π as a functions from a set of states S and $E(S)$ — as set of actions applicable to S.

2.4.2 Towards Temporal-Preferential STRIPS

The "core" of STRIPS consists in the construction of a plan by successive adding new (nondeterministically chosen) actions from a given set of admissible actions. The main choice criterion of them is their relevance to a task goal. The same idea should be preserved in the new STRIPS-extension denoted later as: 'TP-STRIPS' (*Temporal-Preferential STRIPS*).

It arises a natural question how to extend the STRIPS-algorithm by considering new temporal and preferential components. It seems that there is no determinant with respect to the way of extending of STRIPS. For example, one could make it 'step by step', choosing a single action only in each step. By contrast – in a maximalistic version – one could choose the whole subsets of actions respecting temporal constraints in a single step before moving to the preferential part of the algorithm.

In the extension that will be proposed now, a compromise solution is chosen. In fact, pairs of actions are chosen in the 'temporal' step of the extension. This *manouvre* seems to correspond well to a comparative nature of preferences, when an entity (an object, an action, a value, etc.) is chosen for a cost of another one. Finally, we decide for the following chronology of the extending steps.

Ext-step1: After a nondeterministic choosing an action $a \in \mathcal{A}$, we also nondeterministically chose other action a_1 from $\mathcal{A} - \{a\}$ relevant to a given goal g,

Ext-step2: If a set of temporal constraints C imposed on $\mathcal{A}$ and also actions a and a_1 are given, we check which actions from this pair respect temporal constraints from C (By the way, a new subset $\mathcal{A}^C \subseteq \mathcal{A}$ of actions respecting constraints from C is constructed),

Ext-step3: If both the chosen actions a and a_1 respect temporal constraints from C, we compute $F(d,z)^a_n$ and $F(d,z)^{a_1}_n$ (for a fixed agent n) and compare preferences associated to each of them with values of `the preferential function Pref(x)`.

More precisely, the procedure is the following now: Taking two actions, say $\{a_1, a\}$ for a given pair (d,z) and the agent set N, $F(d,z)^{a_1}_{\mathcal{N}}$ and $F(d,z)^{a}_{\mathcal{N}}$ are computed.

- if $\forall x \in (d,z)$ $\Big(F(d,z)^{a_1}_{\mathcal{N}} > \texttt{Pref(x)} > F(d,z)^{a}_{N}\Big)$, then we take a_1 and we take $\pi.a_1$ as a current plan.
- if $\forall x \in (d,z)$ $\Big(F(d,z)^{a}_{N} > \texttt{Pref}(\mathrm{x}) > F(d,z)^{a_1}_{N}\Big)$, then take a and we take $\pi.a$ as a current plan.

The steps of STRIPS-extending illustrate the difference between the inputs of STRIPS and TP-STRIPS. Recall also that **the STRIPS input** contains:

- a set of actions $\mathcal{A}$,
- an initial state s and
- a goal g.

The TP-STRIPS input should also contain:

- `a fixed pair (d,z)`, where d denotes a fixed day and z is a fixed shift,
- `a set of temporal constraints C` and
- `the preferential Pref`(x): $(d,z) \mapsto [0,1]$ and
- `function` $F(d,z)^a_n$ – defined as earlier.

It enables of introducing TP-STRIPS as presented in the table.

TP-STRIPS ($\mathcal{A}$, $\mathcal{N}$, C, `Pref(x)`, s, goals, `(d,z)`, $F(d,z)^{\mathcal{A}}_N$)
begin
  $\pi \leftarrow$ the empty plan
  loop
  if s satisfies g **then return** π
  $\mathcal{A} \leftarrow \{a|$ a is a ground instance of an opertor in O,
      and a is relevant for $g\}$
  if $\mathcal{A} = \emptyset$ **then return** `failure`
  choose nondeterministically an action $a \in \mathcal{A}$
  choose nondeterministically an action $a_1 \in \mathcal{A} - \{a\}$
  check whether a and a_1 respect temporal constraints from C
      if a respects C and a_1 does not, **then take** a
        $s \leftarrow \gamma(s, a)$
        $\pi \leftarrow \pi.a.$
        $\mathcal{A} \leftarrow \mathcal{A}^C = \{a\}.$
      if a_1 respects C and a does not **then take** a_1
        $s \leftarrow \gamma(s, a_1)$
        $\pi \leftarrow \pi.a_1$
        $\mathcal{A} \leftarrow \mathcal{A}^C = \{a_1\}.$
      otherwise take $\{a, a_1\}$
        $s \leftarrow \gamma(s, a_1)$
        $\pi \leftarrow \pi.a_1 \vee \pi \leftarrow \pi.a_2$
        $\mathcal{A} \leftarrow \mathcal{A}^C = \{a_1, a\}.$
        if$\mathcal{A}^C = \{a_1, a\}$, **then compare** $F(d,z)^{a_1}_N$ and $F(d,z)^{a}_N$ with `Pref(x)`
          if $\forall x \in (d,z)\Big(F(d,z)^{a_1}_N >$`Pref`$(x)> F(d,z)^{a}_N\Big)$ **then take** a_1
            $s \leftarrow \gamma(s, a_1)$
            $\pi \leftarrow \pi.a_1.$
          if $\forall x \in (d,z)\Big(F(d,z)^{a}_N >$`Pref`$(x)> F(d,z)^{a_1}_N\Big)$ **then take** a
            $s \leftarrow \gamma(s, a)$
            $\pi \leftarrow \pi.a.$
          if $\forall x \in (d,z)\Big(F(d,z)^{a_1}_N > F(d,z)^{a}_N >$ `Pref(x)`$\Big)$ **then take** a_1
            $s \leftarrow \gamma(s, a_1)$
            $\pi \leftarrow \pi.a_1.$
          if $\forall x \in (d,z)\Big(F(d,z)^{a}_N > F(d,z)^{a_1}_N >$ `Pref(x)`$\Big)$ **then take** a
            $s \leftarrow \gamma(s, a)$
            $\pi \leftarrow \pi.a.$
          otherwise take $\emptyset$
end

TP-STRIPS in use will be described in Chapter 3. Meanwhile, it is not difficult to observe that TP-STRIPS is more restrictive than its original version. In fact, not only it is checked whether actions from $\mathcal{A}$ are relevant to g (as in STRIPS), but also – which of them satisfy constraints from C and the imposed preferences.

Unfortunately, it may have some unexpected consequences. Namely, no plan may respect all the constraints considered in TP-STRIPS. (For example, no actions is preferable in the sense, described above.)

Fortunately, TP-STRIPS is a decidable procedure. It may be comprehended as follows this context: it always requires a finite number of steps only – even if no plan exists according to TP-STRIPS requirements. In addition, its output polynomially depends on the input size. The following lemma renders the meta-logical features of this procedure.

Lemma 2 *Assume that a set of action $\mathcal{A}$ with card($\mathcal{A}$) $= n$ and a set of temporal constraints C imposed on actions from $\mathcal{A}$ with $card(C) = m$ are given. Then TP-STRIPS is decidable and its output size polynomially depends on the input size.*

Proof: Let $\mathcal{A}$ and C are such as described in the lemma above, i.e let $card(\mathcal{A}) = n$ and $card(C) = m$. Our output is N_{Choice} – a number of possible choices of actions from $\mathcal{A}$ in the whole TP-STRIPS procedure. Since in both the 'temporal' and the 'preferential' step of TP-STRIPS we always choose 2 actions from n-elemental set $\mathcal{A}$ of them, we can do it in $\binom{n}{2}$ ways. Each of such pairs of actions, say (a_i, a_j), for $1 \leq i, j \leq n$ should be now checked m-times to check which constraints from C are satisfied by it. Thus, we need $m\binom{n}{2}$ moves in the temporal step of TP-STRIPS. For each such a choice in the temporal step, we can choose 2 actions from maximally n-elemental set $\mathcal{A}$ in the 'preferential' step[18] Thus, the number of possible choices in these two steps N_{Choice} is constrained by $m\binom{n}{2}\binom{n}{2} = m\binom{n}{2}^2$. Since

$$m\binom{n}{2}\binom{n}{2} = m\binom{n}{2}^2 = m\Big(\frac{n!}{2!(n-2)!}\Big)^2 = \frac{m}{2}\Big((n-1)n\Big)^2 = \frac{m}{2}n^2(n-1)^2, \tag{2.69}$$

thus $N_{Choice} \leq O(n^4)$, what justifies its polynomial dependence on the input size. □

It is noteworthy that the same property of under exponential complexity of TP-STRIPS is preserved by a more general situation, when temporal constraints are imposed not only on single actions, but also on pairs of actions, on their triples, ..., on k-tuples of actions from $\mathcal{A}$. This feature is formulated by the following lemma.

Lemma 3 *Assume that TP-STRIPS is modified such that one consider single actions, their pairs, 3-tuples up to $n-1$-tuples from $\mathcal{A}$ in its 'temporal' step – according to different types of temporal constrains from a finite set C imposed on single actions, pairs of actions, etc.). Then number of possible moves in 'temporal' step of TP-STRIPS is under exponentially dependent on the input size.*

Proof: Assume, as earlier, that $card(\mathcal{A}) = n$ and $card(C) = m$. Suppose that C_1 denotes constraints from C imposed on single actions, C_2 – denotes constraints imposed on pairs of actions, C_3 – imposed on 3-tuples,$\ldots, C_{n-1}$ – on $n-1$-tuples. Finally, assume that $card(C_1) = m_1, card(C_2) = m_2, \ldots, card(C_{n-1}) = m_{n-1}$.

Then, obviously, $\sum_{k=0}^{n} m_{k-1} = m$ and we must successively choose 2 actions, the next – 3 actions, $\ldots, n-1$ actions from n-elemental $\mathcal{A}$. (Note that we can choose the same actions in new steps). Thus, all possible choices in the temporal step N_{choice}^{Temp} (our output):

$$N_{choice}^{Temp} = m_1\binom{n}{1} + m_2\binom{n}{2} + m_3\binom{n}{3} + m_4\binom{n}{4} + ... + m_{n-1}\binom{n}{n-1} \leq m\sum_{k=0}^{n}\binom{n}{k}. \tag{2.70}$$

Because of the known equality:

$$\sum_{k=0}^{n}\binom{n}{k} = 2^n \tag{2.71}$$

[18]This possibility holds if all actions remain 'good' as respecting temporal constrains from C.

we can get:

$$N_{choice}^{Temp} = m_1\binom{n}{1} + m_2\binom{n}{2} + m_3\binom{n}{3} + m_4\binom{n}{4} + ... + m_{n-1}\binom{n}{n-1} < m2^n. \tag{2.72}$$

Hence, $N_{choice}^{Temp} < O(exp(n))$, what already shows the thesis. $\square$

2.4.3 Fuzzy Temporal Constraints and Preferences in Davis-Putnam Procedure

The same idea of adopting the conceptual apparatus, elaborated in 2.2 and 2.3, will be materialized with respect to Davis-Putnam Procedure. The proper analysis is proceed by a brief repetition of the main ideas of this procedure.

Davis-Putnam procedure – a brief repetition. The original Davis-Putnam procedure (see: 3.1 of 'Introduction' pp. 11-13), forms a two-valued-based proof procedure for propositional languages. It may be very briefly given by the algorithm:

`Davis-Putnam procedure` (Φ in CNF)
 Input: A set of clauses Φ
 Output: A Truth Value

More detailed:

`Davis-Putnam`(Φ, μ, v)
begin
 if $\emptyset \in \Phi$ then return
 if $\Phi = \emptyset$ then exit with μ
 otherwise `Unit-propagate`
 select a variable P such that P or $\neg P$ occurs in Φ
 `Davis-Putnam` ($\Phi - \{P\}, \mu \cup \{\neg P\}$)
 `Davis-Putnam` ($\Phi - \{\neg P\}, \mu \cup \{P\}$)
end

The crucial role of two-valued logic as a 'basis' of this procedure also manifests itself by the fact that formulas are valid if and only if their negation are unsatisfiable.

Towards 'temporal' Davis-Putnam procedure. It arises a natural question of a generalization of this procedure by admitting more than two truth values. Suppose – for simplicity – that new fuzzy values are admitted in branches in, say k-level, for $k \leq n$ of Davis-Putnam procedure tree (preferable at the final step). Obviously, this new P-D procedure should radically deviate from the original one in cases of conjunctions $P \wedge \neg P$ (of literals). Since a fuzzy value of $P \wedge \neg P$ is generally different from 0 and is determined by some t-norm, it allows us to specify the new Davis-Putnam procedure as follows.

Consider a n-level tree Davis-Putnam procedure for some formulas Φ (of a first-order propositional language) given in CNF.

A Until a formula $P \wedge \neg P$[19] occurs in k-level for $k \leq n$, Davis-Putnam procedure with unit propagation works without changes,

B When a formula $P \wedge \neg P$ occurs (if any), P-D works as follows:

- fuzzy values are associeted to P and $\neg P$, i.e $v(P), v(\neg P)$,
- $v(P \wedge \neg P)$ as a given t-norm$(v(P), v(\neg P))$ is computed.

[19] P is an atomic variable or a term of a given language built up from atomic variables.

If now $v(P \wedge \neg P) = 0$, any model exists, so $\mu = \emptyset$. If $v(P \wedge \neg P) = t\text{-norm}(v(P), v(\neg P)) \neq 0$, we get a model μ with a degree $v(P \wedge \neg P)$.

It allows us to propose the following 'temporal' extension of Davis-Putnam procedure in (Łukasiewicz logic with min-norm).

`Davis-Putnam`(Φ, μ, v)
begin
 if $\emptyset \in \Phi$ then return
 if $\Phi = \emptyset$ then exit with μ
 otherwise `Unit-propagate`
 select a variable P such that P or $\neg P$ occurs in Φ
 `Davis-Putnam` $(\Phi - \{P\}, \mu \cup \{\neg P\})$
 `Davis-Putnam` $(\Phi - \{\neg P\}, \mu \cup \{P\})$
  if $P \wedge \neg P$ associate $v(P)$ and $v(\neg P)$
  compute $v(P \wedge \neg P) = \min(v(P), v(\neg P))$.
   if $v = 0$, then `failure` and $\mu = \emptyset$
   if $v \neq 0$, then return model μ with a degree v.
end

A similar 'temporal' extension of Davis-Putnam procedure may be proposed in Product Fuzzy Logic with a product norm as follows.

`Davis-Putnam`(Φ, μ, v)
begin
 if $\emptyset \in \Phi$ then return
 if $\Phi = \emptyset$ then exit with μ
 otherwise `Unit-propagate`
 select a variable P such that P or $\neg P$ occurs in Φ
 `Davis-Putnam` $(\Phi - \{P\}, \mu \cup \{\neg P\})$
 `Davis-Putnam` $(\Phi - \{\neg P\}, \mu \cup \{P\})$
  if $P \wedge \neg P$ associate $v(P)$ and $v(\neg P)$
  compute $v(P \wedge \neg P) = v(P) \bullet v(\neg P))$.
   if $v = 0$, then `failure` and $\mu = \emptyset$
   if $v \neq 0$, then return model μ with a degree v.
end

As earlier, it arises a natural question of a complexity of this procedure. Unfortunately, a solving of this problem in the whole generality seems to be problematic. Nevertheless, one can approximate this solution if only immerse this TP-Davis-Putnam procedure in the context of Four-Valued Fuzzy Language. This language was described in detail in [102, 103]. It may be obtained as described.

Namely, assume $\mathcal{L}_4$ is a propositional language with connectives $\vee, \wedge, \neg$. Assume that to each proposition of $\mathcal{L}_4$ one can associate a one of the four values: *true, false, unknown and contradiction.* If A is a formula of $\mathcal{L}_4$, its derivability from a set $\sum$ of propositions, called 'Knowledge Base' will be denoted as $\sum \models_4 A$. We can enlarge $\mathcal{L}_4$ to $\mathcal{L}_4^{fuzzy}$ by introducing fuzzy propositions of the form $[A \geq n]$, for $n \in [0, 1]$ (for example: $[A \geq 0, 7]$). It mean that a fuzzy value of a proposition $A \geq n$.

Whenever we refer to formulas of $\mathcal{L}_4^{fuzzy}$, we can also say about$\sum$-derivability (as derivability with a fuzzy degree $\alpha \in [0, 1]$ for $\sum$-formulas)). Denote it by $\sum |\!\approx_4 [A \geq n]$. As usual, denote also by CNF a *Conjunctive Normal Form* for $\mathcal{L}_4^{fuzzy}$-formulas. Then it holds the following theorem.

Theorem 9 *(Straccia) [103]. $\sum |\!\approx_4 [A \geq n]$ is coNP-complete as $\sum \models_4 A$ is. Given $\sum$ and $[A \geq n]$ in CNF (i.e. both $[A \geq n]$ and all formulas in $\sum$ are formulas in CNF), then a checking $\sum |\!\approx_4 [A \geq n]$ can be done in $O(|\sum||A|)$, i.e. it depends on the cardinality of $|\sum|$ multiplied by $|A|$.*

It allows us to formulate the following theorem describing a complexity problem of TP-Davis-Putnam procedure.

Theorem 10 *Assume that TP-Davis-Putnam procedure – as based on Product norm – refers to formulas of $\mathcal{L}_4^{fuzzy}$ (given as above) a set of Knowledge Base $\sum$ and fuzzy propositions $[A \geq n]$ with $\sum$-derivability $\sum |\approx_4 [A \geq n]$ for them. Given $\sum$ and $[A \geq n]$ in CNF (i.e. both $[A \geq n]$ and all formulas in $\sum$ are formulas in CNF), checking derivability for TP-Davis-Putnam procedure can be done in $O(|\sum||A|)$.*

Proof: Note that checking derivability for TP-Davis-Putnam procedure may be identified with:

1. checking $\sum |\approx_4 [A \geq n]$ for a finite set of such fuzzy formulas $[A \geq n] \in \mathcal{L}_4^{fuzzy}$ or
2. checking $\sum |\approx_4 [A \wedge B \geq nm]$ for a finite set of formulas $[A \geq n], [B \geq m] \in \mathcal{L}_4^{fuzzy}$ (since we consider fuzzy values $v(A \wedge B) = v(A) \bullet v(B)$ in TP-Davis-Putnam procedure).

In fact, TP-Davis-Putnam procedure always contains a finite set of such formulas (in practice – a very small set). Assume the first case. In this case the thesis immediately follows from the theorem above, which asserts that $\sum |\approx_4 [A \geq n]$ for each such a formula $[A \geq n]$ in $\mathcal{L}_4^{fuzzy}$ can be done in $O(|\sum||A|)$.

It is easy to see that the second case with formulas $[A \wedge B \geq nm]$ forms a unique subcase of the earlier one. The case of Davis-Putnam procedure based on Łukasiewicz norm is similar to consider. We should only exchange mn for $\min\{m, n\}$. $\square$.

2.5 Concluding Remarks

As it has been just shown some new operational approach to modeling of fuzzy temporal constrains and preferences can be proposed on a base of earlier ideas of H-J. Ohlbach form [86, 85] in terms of such basis notions of real analysis and measure theory Lebesgue integrals and convolutions.

It is true that – independently of a transparently graspable computational power of the convolution-based approach – has not yet exploited and fully elucidated in examples of this section. In fact, this issue requires a more detailed attention and it deserves a broader analysis. All these computational and programing-wise aspects of the proposed approach will be a subject of the next paragraph.

Meanwhile, the meta-logical aspects of these considerations (a construction of a complete fuzzy logic system for fuzzy Allen's relations) will be presented in details in Appendix on a base of author's works [104, 105, 76].

Chapter 3

Computable and programming-wise aspects of temporal planning with fuzzy constraints and preferences

Inhaltsangabe

3.1 Introduction . . . 118

3.1.1 Motivation of the Chapter . . . 118

3.1.2 Objective and Novelty of this Chapter . . . 118

3.2 Computational Aspects of Temporal Planning with Fuzzy Temporal Constrains and Preferences . . . 119

3.2.1 The Extended STRIPS in Use . . . 119

3.2.2 Extended Putnam-David Procedure in Use . . . 122

3.3 Programming-wise Aspects of Multi-Agent Schedule-Planning Problem . . . 127

3.4 Concluding remarks . . . 132

This chapter provides a particular discussion of computational and programming-wise aspects of temporal planning with fuzzy temporal constraints and preferences. The subject context is determined by Temporal Multi-Agent Problem (TM-AP), earlier introduced. At first, it will be shown how the convolution-based fuzzy Allen's relations may support temporally extended STRIPS (TP-STRIPS) and Davis-Putnam procedure. Secondly, some paradigmatic subproblem of (TM-AP) – in terms of Simple Temporal Problem under Uncertainty – will be introduced and a class of its solutions will be specified thanks the convolution-based representation of fuzzy Allen's relations. Finally, programming-wise aspects of TM-AP will be discussed in terms of PROLOG-solvers for chosen efficient subcases of this problem.

3.1 Introduction

Last chapter of this part of the thesis was aimed at proposing a new convolution-based conceptualization of fuzzy temporal constraints and preferences. It emerges that this conceptual apparatus may support such planning methods as: STRIPS and Davis-Putnam procedure.

3.1.1 Motivation of the Chapter

Nevertheless, a presentation of this conceptual from chapter 1 forms a purely theoretic construct and requires further complementation. It was also presented without a broader illustration of the computational power and utility of the introduced tools. As the matter of fact, this theoretic approach suffers from the following lacks – from the more practical point of view:

Lack1: It is now clear how (and whether at all) the convolution-based Allen's relations can allow to specify solutions of Multi-Agent Problem and its subproblems.

Lack2: It is not clear how the definitions of fuzzy temporal constrains and global preferences cooperate with temporally extended STRIPS in a more practical contexts.

Lack3: Finally, which practical solutions of the scheduling aspects of Multi-Agent Problem may be achieved by the PROLOG-solvers?

All these open questions motivate the analysis of current chapter.

3.1.2 Objective and Novelty of this Chapter

According to these lacks, just identified, objectives of this chapter are the following:

Obj1: To show that convolution-based depiction of Allen's relations allows to radically enlarge a class of admissible solution of (some subproblems of) Multi-Agent Problem.

Obj2: To illustrate how new definitions of fuzzy temporal constrains and preferences support such planning procedures as STRIPS and Davis-Putnam procedure.

Obj3: To give exemplary schedules for some efficient subcases of Multi-Agent Problem in terms of PROLOG-solvers.

Since we exploit definitions of fuzzy temporal constraints and preferences from chapter 1, we should refer to both qualitative and quantitative temporal constraints. The first category is represented here by chosen fuzzy Allen's relations. Quantitative temporal constrains will encode fuzzy temporal constrains – as points of fuzzy intervals with finite supports, i.e. as built up from such pairs $(x_i, f(x_i))$ that there exist such a and b that $a \leq x_i \leq b$. Since the second condition introduces a context of Simple Temporal Problem under Uncertainty, we can assume that we will consider Multi-Agent Problem here in terms of STPU and fuzzy

Allen's relations[1].

Novelty of this chapter consists in:

Nov1 adopting convolutions – as representing Allen's relations – to solving STPU,

Nov2 adopting convolutive representation of Allen's relations to STRIPS and Davis-Putnam procedure in order to temporally extend them,

Nov3 exploiting PROLOG-solvers to Multi-Agent Problem in its fuzzy and temporal extension.

3.2 Computational Aspects of Temporal Planning with Fuzzy Temporal Constrains and Preferences

3.2.1 The Extended STRIPS in Use

TP-STRIPS, as described in Section 2.4, was introduced in a purely theoretic way. Return now to TP-STRIPS once again to illustrate how this extension works in the more practical contexts. In fact, we are interested in the 'preferential' part of TP-STRIPS. Repeat that this part was introduced in terms of `the preferential function Pref(x)` and F-function defined for pairs (d, z) for a given action $a \in \mathcal{A}$ and a set of agents N.

However, a slight reformulation of this TP-STRIPS in terms of f-function instead of F (see: Section 2.3, pp. 75-76.) seems to be even more convenient for computational reasons. In fact, f forms a function of x-argument, but F is a function of pairs (d, z). For that reason, let us recall the appropriate part of TP-STRIPS beginning with 'Compute of Preferential' in f-reformulation.

Compute the Preferential (3.1)

TP-STRIPS ($\mathcal{A}$, $\mathcal{N}$, C, `Pref(x)`, s, goals, `(d,z)`, $f(d,z)^{\mathcal{A}}_N$)
begin
....
compute the Preferential (`Pref(x)`-function)
compare $f(x)^{a_1}_N$ and $f(x)^{a}_N$ with `Pref(x)`
if $\forall x \in (d,z)\Big(f(x)^{a_1}_N > \texttt{Pref}(x) > f(x)^{a}_N\Big)$ then take a_1
$s \leftarrow \gamma(s, a_1)$
$\pi \leftarrow \pi.a_1.$
if $\forall x \in (d,z)\Big(f(x)^{a}_N > \texttt{Pref}(x) > f(x)^{a_1}_N\Big)$ then take a
$s \leftarrow \gamma(s, a)$
$\pi \leftarrow \pi.a.$
if $\forall x \in (d,z)\Big(f(x)^{a_1}_N > f(x)^{a}_N > \texttt{Pref(x)}\Big)$ then take a_1
$s \leftarrow \gamma(s, a_1)$
$\pi \leftarrow \pi.a_1.$
if $\forall x \in (d,z)\ \Big(f(x)^{a}_N > f(x)^{a_1}_N > \texttt{Pref(x)}\Big)$ then take a
$s \leftarrow \gamma(s, a)$
$\pi \leftarrow \pi.a.$
otherwise take $\emptyset$
end

[1]This slight modification does not change a nature of consideration – in a comparison with chapter 1 – but enables a more flexible computations.

In order to illustrate how it works in detail let us consider the following sub-problem of Multi-Agent Problem.

SUB-PROBLEM of MAP. Assume that (in some planning component of MAP) the following plans by means of STRIPS was found for some agents.

$$\pi_1 = \langle a_2, a_7, a_3, a_4, a_5, a_1 \rangle, \quad \pi_2 = \langle a_2, a_7, a_3, a_4, a_6, a_1 \rangle. \tag{3.2}$$

Assume that also the following facts are known:

- temporal constrains imposed on actions a_5 and a_6 are as presented on the picture and
- temporal constraints imposed on a_6 takes values $(\frac{3}{2}, 4\frac{1}{2})$ a constant value 0,1 for a_5 and over a time intervals (1,2).
- preferences imposed on actions are determined by a linear function.
- fuzzy temporal constraints imposed on the set of actions: $a_1 \ldots, a_7$ are approximated by the convolution[2]:

$$C(x) = g(x-t) * h(t) = \int e^{x-t} \sin t \, \mathrm{d}t^3. \tag{3.3}$$

Compute the preferential line and decide which plan π_1 or π_2 is the appropriate one.

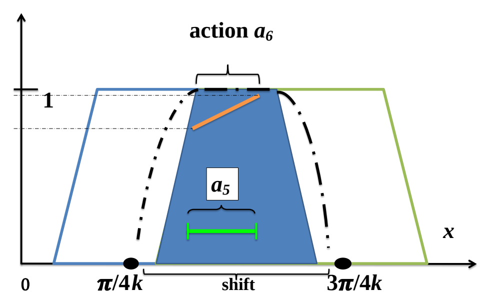


Figure 3.1: Actions a_5 and a_6 with temporal constrains imposed on them (the blue region) and fuzzy temporal constrains determined by the convolution (black broken line).

SOLUTION. To solve the problem:

A one needs compute the line determining preferences, say `Pref(x)`, using convolution $C(x)$,

B one needs to check which action a_5 or a_6 satisfies the preferences, i.e. whether $\phi(x)^{a_5} >$ `Pref(x)` or $\psi(x)^a_6 >$ `Pref(x)`.

In order to simplify the solution, assume that a_5 and a_6 are representable as follows:

$$a_5 = \left\{ \Big(x, \phi(x)\Big) : 0 \leq x \leq \frac{3\pi}{4} \text{ and } \phi(x) \leq 1 \right\}, \tag{3.4}$$

[2]We can assume that $h(x)$ represents $meet(i,j)(x)$ Allen relation. Anyhow, this identification is redundant from the point of view of the current analysis.

[3]Obviously, this functions is exemplary, but it is relatively convenient for computations as $(e^x)' = e^x$.

$$a_6 = \Big\{ \Big(x, \psi(x) \Big) : 0 \le x \le \frac{3\pi}{4} \text{ and } \psi(x) \le 1 \Big\} \tag{3.5}$$

for some (Lebesgue integrable) functions ϕ and ψ defined over $[0, \frac{3\pi}{4}]$.

Computing the preferential line using convolution $C(x)$**.**. It remains to consider the temporal constrains approximated by the given convolution.

$$C(x) = g(x-t) * h(t) = \int e^{x-t} \sin t \, \mathrm{d}t. \tag{3.6}$$

Let us also define `Pref(x)` $= \varphi(C)(x)$. This function $\varphi(C)(x)$ will be considered as the 'preferential line'. Due to earlier guidelines it remains to check whether:

$$\forall x \in [0, \frac{3\pi}{4}] \ \Big(\phi(x) \le \texttt{Pref}(x) \text{ or } \texttt{Pref}(x) \le \phi(x) \Big) \text{ and} \tag{3.7}$$

$$\forall x \in [0, \frac{3\pi}{4}] \ \Big(\psi(x) \le \texttt{Pref}(x) \text{ or } \texttt{Pref}(x) \le \psi(x) \Big). \tag{3.8}$$

Therefore – due to Convolution Theorem – we obtain:

$$F(C(x)) = \int e^{x-t} \, \mathrm{d}z \int \sin t \, \mathrm{d}t, \tag{3.9}$$

where F denotes Fourier's transform of $C(x)$. Since

$$\int e^{x-t} \, \mathrm{d}z \int \sin t \, \mathrm{d}t = e^x \cos\, t, \tag{3.10}$$

thus $F * (h)(x)$ – as $F(h)(x)$ considered in integration limits $a, b \in [0, \frac{3\pi}{4}]$ is as follows:

$$F * (h)(x) = e^x [\cos\, t]_b^a = e^x (\cos a - \cos b) = \tag{3.11}$$

$$= -2 \sin \frac{a+b}{2} \sin \frac{a-b}{2} e^x. \tag{3.12}$$

Putting $A = -2 \sin \frac{a+b}{2} \sin \frac{a-b}{2}$, we obtain:

$$F * (C)(x) = Ae^x, \text{ so } \ C(x) = \frac{d}{dx} F * (C)(x) = Ae^x. \tag{3.13}$$

Thus, we can already treat $C(x)$ as an explicit function of x-argument.

Assume now two points $P_0 = (x_0, C(x_0))$ and $P_1 = (x_1, C(x_1))$ belonging to the diagram of $C(x)$. Because $P_0 = (x_0, \varphi(A, g(x_0)))$ and $P_1 = (x_1, \varphi(A, h(x_1)))$ a line passing through these points satisfies the following equation.

$$\varphi(C)(x) = \frac{\varphi(C)(x_1) - \varphi(C)(x_0)}{x_1 - x_0} (x - x_0) + \varphi(C)(x_0) \tag{3.14}$$

Hence $\texttt{Pref}(x) := \varphi\Big((C)(x), A \Big)$ is as follows in our case:

$$\varphi((C(x), A) = \frac{A\Big(e^{x_1} - e^{x_0} \Big)}{x_1 - x_0} (x - x_0) + e^{x_0}. \tag{3.15}$$

For $x_0 = 0$ and $x_1 = \frac{3\pi}{4}$:

$$\varphi(C(x), A) = \frac{4A\Big(e^{\frac{3\pi}{4}} - e^0 \Big)}{\pi} x + e^{\frac{3\pi}{4}} = \frac{4A\Big(e^{\frac{3\pi}{4}} - 1 \Big)}{\pi} x + e^{\frac{3\pi}{4}}, \tag{3.16}$$

for $A = -2\sin\frac{a+b}{2}\sin\frac{a-b}{2}e^x$. It is easy to check that :

$$e^{\frac{3\pi}{4}} \leq \varphi(f(x), A) \leq 2(e^{\frac{3\pi}{4}} - 1), \text{ for } x \in [0, \frac{3\pi}{4}]. \tag{3.17}$$

After dividing by the normalization factor, say $N = 10^4$, we get:

$$\frac{e^{\frac{3\pi}{4}}}{10} \leq \frac{\varphi(f(x), A)}{10} \leq \frac{2(e^{\frac{3\pi}{4}} - 1)}{10}. \tag{3.18}$$

Recall that $\phi(x)_{a_5} = \frac{1}{10}$ for all $x \in [0, \frac{3\pi}{4}]$, so

$$\phi(x)_{a_5} \leq \frac{\varphi(C(x), A)}{10}. \tag{3.19}$$

Simultaneously, $\frac{2(e^{\frac{\pi}{4}}-1)}{10} \leq \frac{e}{5} \leq \phi(x)_{a_6}$, thus

$$\frac{\varphi(C(x), A)}{10} \leq \phi(x)_{a_6}. \tag{3.20}$$

Therefore, we accept the action a_6 and reject a_5 and $\pi = \{a_1, a_2, a_3, a_6, a_7\}$ as the required plan.

Towards a generalization

Obviously, not always the situation is so ideal as the presented above. In fact, it is possible that either both of the following conditions

1. If $\forall x \in (d, z)(f(d, z)^{a_1}_N > P(x) > f(d, z)^{a}_N)$ and
2. If $\forall x \in (d, z)(f(d, z)^{a}_N$ >P(x)> $f(d, z)^{a_1}_N)$

or (at least) one of them do not completely hold. In other words, we think about a situation when fuzzy values of an action, say, a not for all $x \in (d, z)$ are greater (smaller) than values of computed `Pref`-function. It also arises another question: what to in a case when both fuzzy values of two actions, say, a_1 and a_2 not for all $x \in (d, z)$ are greater/smaller than values of `Pref`-function. How to decide which action is better?

In such a situation, one can venture to associate a Lebesgue measure μ to these appropriate parts of linear diagrams of actions a_1 and a_2. Assume that `Pref` cut from an interval representing a_1 some subinterval, say I_1 and a similar subinterval I_2 from the interval representing a_2. If $\mu(I_1) > \mu(I_2)$, then action a_1 is better than a_2. Otherwise – action a_2 better respects preferential requirements.

3.2.2 Extended Putnam-David Procedure in Use

One of ideas of last chapter was an introducing a new fuzzy TP-Davis-Putnam procedure. It forms a temporal-preferential extension of the original P-D procedure. TP-Davis-Putnam procedure was proposed in two variants – dependently of t-norms used to compute fuzzy values of formulas of the form $P \wedge \neg P$. In this chapter we intend to exemplify this new (fuzzy) TP-Davis-Putnam procedure.

In oder to make it, return to the following planning example solved via Davis-Putnam procedure from section 3.1. of chapter 1 of 'Introduction'.

Example 28 *Consider a very basic planning situation described by a formula describing a possible situation of an agent's activity. The reservoir of the agent actions is the following :* $Move(P)$ *(the agent can move from a point P),* $Put(A)$ *(the agent can put a block A somewhere),* $Load(B)$ *(the agent can load B somewhere). Find a possible consistent plan of agent's activity.*

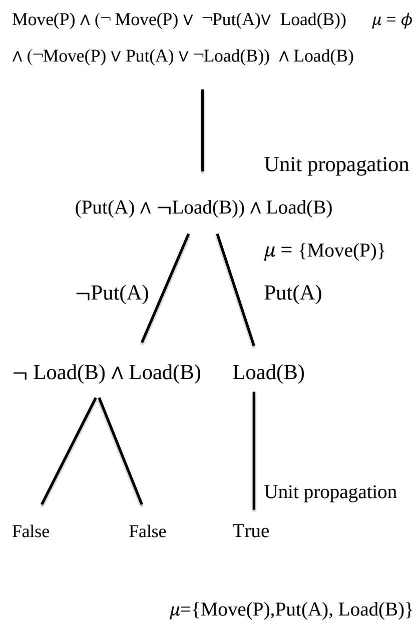


Figure 3.2: Davis-Putnam procedure for a basic planning case

SOLUTION: The solution via Davis-Putnam is illustrated in Figure 3.2.

Assume now that we also have – from some *other sources* – a knowledge about temporal constrains imposed on actions `Load(B)` and ¬ `Load(B)`[5]. For simplicity assume that these temporal constraints – in terms of normalized values of the appropriate function[6] – are as depicted in Figure 3.4. Let us assume that the `unit-propagate` has already put forward the formula

$$\Phi = \mathrm{Load(B)} \wedge \neg\mathrm{Load(B)}. \tag{3.21}$$

However, two fuzzy values (0.8 for `Load(B)`) and 0.2 for ¬ `Load(B)` was already associated to these atomic action formulas. As earlier mentioned, `Load(B)`∧¬ `Load(B)` should not imply any contradiction in a fuzzy case as this conjunction must not necessary take 0. In order to check it, assume that we decide to consider this conjunction in Łukasiewicz logic. Recall that TP-Davis-Putnam algorithm in this fuzzy logic type is as follows.

[4]We should guarranty that we obtain values from $[0, 1]$ after the normalization.
[5]The action ¬ `Load(B)` should be interpreted as some action different from `Load(B)`.
[6]Intentionally, we think about function f – defined as earlier.

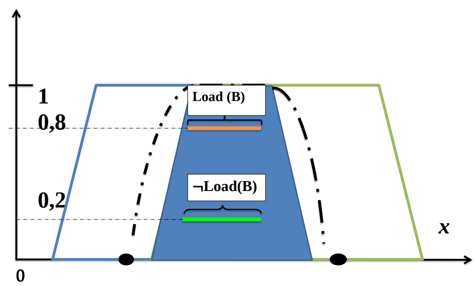


Figure 3.3: Actions `Load(B)` and ¬ `Load(B)` with associated values 0.8 and 0.2 (*resp.*)

`Davis-Putnam`(Φ, μ, v)
begin
if $\emptyset \in \Phi$ then return
if $\Phi = \emptyset$ then exit with μ
otherwise `Unit-propagate`
select a variable P such that P or $\neg P$ occurs in Φ
`Davis-Putnam` $(\Phi - \{P\}, \mu \cup \{\neg P\})$
`Davis-Putnam` $(\Phi - \{\neg P\}, \mu \cup \{P\})$
if $P \wedge \neg P$ associate $v(P)$ and $v(\neg P)$
compute $v(P \wedge \neg P) = \min(v(P), v(\neg P))$.
if $v = 0$, then `failure` and $\mu = \emptyset$
if $v \neq 0$, then return model μ with a degree v.
end

According to it, a fuzzy value v(`Load(B)`∧¬ `Load(B)`) $= \min\{0.8, 0.2\} = 0.2$ in Łukasiewicz fuzzy logic. In this situation (of conjunction interpreted by Łukasiewicz t-norm) the solution via modified Davis-Putnam procedure is given as visualized in Figure 3.4. In results, we obtain the following models[7]:

1. an 'old' model $\mu_1 = \langle Move(P), Put(A), Load(B)\rangle$– with a true degree 0.8,

2. a new model $\mu_2 = \langle Move(P), Put(A), Load(B) \wedge \neg Load(B)\rangle$ with a true degree 0.2.

A similar 'temporal' extension of Davis-Putnam procedure may be proposed in Product logic case as follows.

[7]As it was signalized in 'Introduction' models μ of Davis-Putnam procedure play a role o plans.

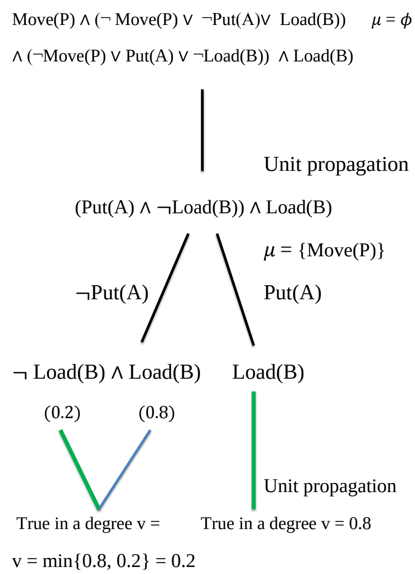


Figure 3.4: Davis-Putnam procedure in a fuzzy case. In this case we do not reject `Load(B)`$\wedge\neg$ `Load(B)`, which gives us a model μ with a true degree v = 0.2.

`Davis-Putnam`(Φ, μ, v)
begin
  if $\emptyset \in \Phi$ then return
  if $\Phi = \emptyset$ then exit with μ
  otherwise `Unit-propagate`
  select a variable P such that P or $\neg P$ occurs in Φ
  `Davis-Putnam` $(\Phi - \{P\}, \mu \cup \{\neg P\})$
  `Davis-Putnam` $(\Phi - \{\neg P\}, \mu \cup \{P\})$
    if $P \wedge \neg P$ associate $v(P)$ and $v(\neg P)$
    compute $v(P \wedge \neg P) = v(P) \bullet v(\neg P)$).
      if $v = 0$, then `failure` and $\mu = \emptyset$
      if $v \neq 0$, then return model μ with a degree v.
end

Due to this algorithm – the same v(`Load(B)`$\wedge\neg$ `Load(B)`$) = 0.8 \bullet 0.2 = 0.16$ in Product Logic[8].

Further preferential extension of Davis-Putnam procedure. Assume now that our knowledge about performing actions `Load(B)` and ¬ `Load(B)` is slightly more extended. Namely, assume that the preferential line – computed as in earlier STRIPS-algorithm case – is given and it only admits points over its diagram as the admissible ones. Thus, only action `Load(B)` is admissible, ¬ `Load(B)` should be rejected. In a

[8]Note that the so-called *strong conjunction* `Load(B)`$\otimes\neg$ `Load(B)` really gives $v = \max\{0, 0, 8 + 0.2 - 1\} = 0$.

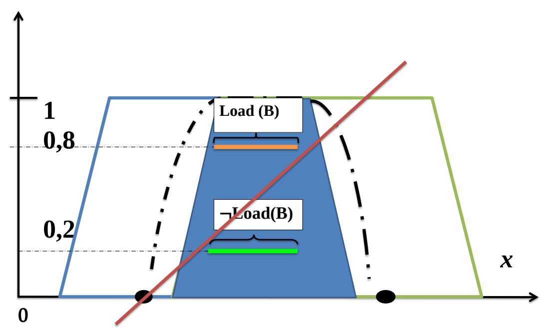


Figure 3.5: Actions `Load(B)` and ¬ `Load(B)` with associated values 0.8 and 0.2 (*resp.*) and the preferential line, which orders to reject ¬ `Load(B)`.

consequence, the solution of the initial example should lead to the final situation as depicted here: In results,

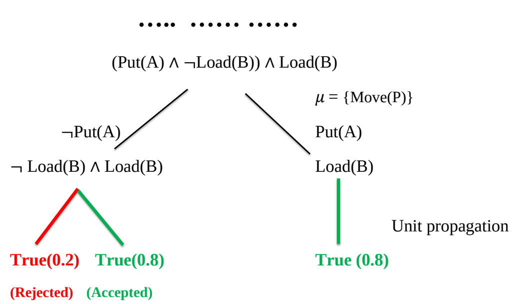


Figure 3.6: Fragment of solution of the initial example with fuzzy temporal constraints and preferences. The red line leads to the rejected solution. The green ones – to the accepted ones.

we obtain two times the following model as a solution: an 'old' model $\mu_1 = \langle Move(P), Put(A), Load(B)\rangle$– with a true degree 0.8.

Let us summarize the reasoning leading to this solution. Let us observe that:

1. until a formula $P \wedge \neg P$ occurs, Davis-Putnam procedure with unit propagation works without changes,
2. when a formula $P \wedge \neg P$ occurs, we work as follows:
   - we associate fuzzy values to P and $\neg P$, or $v(P), v(\neg P)$,
   - we compute the preferential function`Pref`,

- we compare $v(P), v(\neg P)$ with $|\texttt{Pref(x)}|$ – values of the preferential function `Pref` (in a given interval)[9].
  - If $v(P) < \forall x|\texttt{Pref(x)}| < v(\neg P)$, then take $v(\neg P)$ and return model μ with $\neg P$ and degree $v(\neg P)$,
  - If $v(\neg P) < \forall x|\texttt{Pref(x)}| < v(P)$, then take $v(P)$ and return model μ with P and degree $v(P)$,
  - If $v(\neg P) < \forall x|\texttt{Pref(x)}|$ and $v(P) < \forall x|\texttt{Pref(x)}|$, then return model $\mu = \emptyset$.

It allows us to propose the following temporal-preferential extension of the Davis-Putnam procedure.

```
Davis-Putnam(Φ, μ, v, Pref)
begin
    if ∅ ∈ Φ then return
    if Φ = ∅ then exit with μ
    otherwise Unit-propagate
    select a variable P such that P or ¬P occurs in Φ
    Davis-Putnam (Φ − {P}, μ ∪ {¬P})
    Davis-Putnam (Φ − {¬P}, μ ∪ {P})
            if P ∧ ¬P associate v(P) and v(¬P)
            compare v(P) and v(¬P) with |Pref|.
                if v(P) < ∀x|Pref(x)| < v(¬P), then return μ with ¬P and degree v(¬P),
                if v(¬P) < ∀x|Pref(x)| < v(P), then return μ with P and degree v(P),
                if v(¬P) < ∀x|Pref(x)| and v(P) < ∀x|Pref(x)|, then return μ = ∅.
end
```

It is not difficult to observe that no computing of a fuzzy value for a conjunction $v(P \wedge \neg P)$ intervene in this procedure – neither in a sense of Łukasiewicz Logic, nor the Product Logic. I makes this procedure more independent of a logical foundation of analysis. It is also easy to note that the 'preferential' components of this extended Davis-Putnam procedure intervene so deeply in it that they exchange some steps of the purely temporal extension of this procedure. It constitutes some difference in a comparison with a purely theoretic component of the convolution-based approach to representing fuzzy temporal constrains and preferences. In fact, we defined preferences as 'overdefined' on fuzzy temporal constrains. Even computational aspects of their modeling in a context of STRIPS-algorithm allow us to preserve this chronology: at first – temporal constraints, preferences – the next. Meanwhile, computational analysis of Davis-Putnam procedure destroys this chronology as it has already been shown.

3.3 Programming-wise Aspects of Multi-Agent Schedule-Planning Problem

In chapter 2 of the thesis a new convolution-based formalism and models for temporal constraints description were proposed and developed. From another perspective – a skeleton of our considerations was based on 2 key engineering problems: the *Traveling Salesman Problem* and the *Multi-Agent Schedule-Planning Problem.*

In this section, we intend to face the *Multi-Agent Schedule-Planning Problem*[10] in order to illustrate how to solve some of its workable subcases with a support of PROLOG-solvers. In order to illustrate a general method of a construction of these solvers, let us assume that a non-empty set $N = \{X, Y \ldots\}$ of agents and a non-empty set $D = \{1, 2, 3, 4, 5\}$ of working days (for simplicity we omit shifts during a day) are given.

[9]Let us observe that this step is justified as both $v(P), v(\neg P)$ and $|\texttt{Pref(x)}|$ are objects of the same type. They are some values from a set [0,1].

[10]As earlier mentioned, we will use the abbreviation: MAP for this problem in this section, too.

In such a framework, the PROLOG-solver task is to give a schedule respecting the temporal constrains imposed on agent task performing and agent activity. The obtained solutions will be returned in a form of lists of the form:

$$X = [X1, X2, X3, \ldots, Xk], \;\; Y = [Y1, Y2, Y3, \ldots, Yl], \tag{3.22}$$

where $X(i), Y(j)$ are characteristic functions representing activity of agents X and Y during i-day and j-day (*resp.*) for $k, l \in \{1, 2 \ldots, 7\}$.

We can consider two types of situations:

- *crisp*-type situations, when $X(i)$ and $Y(j)$ take only two values: 0 or 1 for $i, j \in \{1, 2, \ldots, 7\}$ or
- *fuzzy*[11]-type situations, when $X(i)$ and $Y(j)$ take more than two values for $i, j \in \{1, 2, \ldots, 7\}$, for example: 0,1,2,3,4. We adopt natural numbers because of restrictions of PROLOG-syntax, which is not capable of representing values from $[0, 1]$. Nevertheless, we intend to think about these values as about normalized values. Namely, we will interpret 1 – taken from a sequence $1, 2 \ldots, k$ – as $\frac{1}{k}$, 2 as $\frac{2}{k}$, k as $\frac{k}{k} = 1$, etc.

An analysis of fuzzy-type cases will be prefaced by the analysis of crisp-type cases.

A) Crisp cases. We will consider 3 subcases of M-AS-PP dependently on a degree of complication. In all these cases 1 – returned in solution-lists – represents a state when an agent n_i occurs in a work in a distinguished shift and day and 0 – otherwise.

As the first one, a case of 2 agents working 5 days in a week (with a week restriction for each of them equal to 4) will be considered. It will be shortly denoted as MaP(2, 5, 4). We assume that a maximal sum of working days of each of them does not exceed 5.

Case 1 – MAP(2, 5, 4) Define at first the PROLOG-plan in this case by the following clauses (The sense of lines of the PROLOG-code is explained on the right side):

```
plan(X,Y):=
X = [X1,X2,X3,X4,X5],          /*list of days of agent X*/
Y = [Y1,Y2,Y3,Y4,Y5],          /*list of days of agent Y*/
X ins 0..1,                     /*Fuzzy degrees of disposition of X */
Y ins 0..1,                     /*Fuzzy degrees of disposition of Y*/
sum(X, #<, 4),                  /*Restriction on X-activity during a week*/
sum(Y, #<, 4),                  /*Restriction on Y-activity during a week*/
X1 # / Y1,# (X1 #/ Y1),        /*Logical conditions on activity of agents each day*/
X2 # / Y2,# (X2 #/ Y2),
X3 # / Y3,# (X3 #/ Y3),
X4 # / Y4,# (X4 #/ Y4),
X5 # / Y5,# (X5 #/ Y5),
label([X1,X2,X3,X4,X5,Y1,Y2,Y3,Y4,Y5]).
```

As a result, the following list of solutions is obtained:

```
X = [0, 0, 1, 0, 1], Y = [1, 1, 0, 1, 0] ;
X = [0, 0, 1, 1, 0], Y = [1, 1, 0, 0, 1] ;
X = [0, 0, 1, 1, 1], Y = [1, 1, 0, 0, 0] ;
X = [0, 1, 0, 1, 0], Y=[1, 0, 1, 0, 1];
X = [0, 1, 0, 1, 1], Y = [1, 0, 1, 0, 0];
```

[11] More precisely: multi-valued situations.

```
X = [0, 1, 1, 0, 0], Y = [1, 0, 0, 1, 1];;
X = [0, 1, 1, 0, 1] Y = [1, 0, 0, 1, 0];
X = [0, 1, 1, 1, 0], Y = [1, 0, 0, 0, 1] ;;
X = [1, 0, 0, 0, 1], Y = [0, 1, 1, 1, 0] ;;
X = [1, 0, 0, 1, 0], Y = [0, 1, 1, 0, 1] ;;
X = [1, 0, 0, 1, 1], ;;;;
X = [1, 0, 1, 0, 0], Y = [0, 1, 0, 1, 1] ;;
X = [1, 0, 1, 0, 1], X = [1, 0, 1, 1, 0], Y = [0, 1, 0, 0, 1] ;
X = [1, 1, 0, 0, 0], Y = [0, 0, 1, 1, 1] ;;
X = [1, 1, 0, 0, 1], Y = [0, 0, 1, 1, 0] ;;
X = [1, 1, 0, 1, 0], Y = [0, 0, 1, 0, 1] ;;
```

Case 2 – MAP(2, 6, 4) Define once again our PROLOG-plan for this case by the clauses:

```
plan(X,Y):=
X = [X1,X2,X3,X4,X5,X6],
Y=[Y1,Y2,Y3,Y4,Y5, Y6],
X ins 0..1,
Y ins 0..1,
sum(X, <, 5),
sum(Y, <, 5),
X1 # / Y1, # (X1 #/  Y1),
X2 # / Y2, # (X2 #/  Y2),
X3 # / Y3, # (X3 #/  Y3),
X4 # / Y4, # (X4 #/  Y4),
X5 # / Y5, # (X5 #/  Y5),
X6 # / Y6, # (X6 #/  Y6),
label([X1,X2,X3,X4,X5, X6, Y1, Y2, Y3, Y4, Y5, Y6]).
```

In result, we obtain the following list of admitted solutions:

```
X = [0, 0, 0, 1, 1, 1], Y = [1, 1, 1, 0, 0, 0] ;
X = [0, 0, 1, 0, 1, 1], Y = [1, 1, 0, 1, 0, 0] ;
X = [0, 0, 1, 1, 0, 1], Y = [1, 1, 0, 0, 1, 0] ;
X = [0, 0, 1, 1, 1, 0], Y = [1, 1, 0, 0, 0, 1] ;
X = [0, 1, 0, 0, 1, 1], Y = [1, 0, 1, 1, 0, 0] ;
X = [0, 1, 0, 1, 0, 1], Y = [1, 0, 1, 0, 1, 0] ;
X = [0, 1, 0, 1, 1, 0], Y = [1, 0, 1, 0, 0, 1] ;
X = [0, 1, 1, 0, 0, 1], Y = [1, 0, 0, 1, 1, 0] ;
X = [0, 1, 1, 0, 1, 0], Y = [1, 0, 0, 1, 0, 1] ;
X = [0, 1, 1, 1, 0, 0], Y = [1, 0, 0, 0, 1, 1] ;
X = [1, 0, 0, 0, 1, 1], Y = [0, 1, 1, 1, 0, 0] ;
X = [1, 0, 0, 1, 0, 1], Y = [0, 1, 1, 0, 1, 0] ;
X = [1, 0, 0, 1, 1, 0], Y = [0, 1, 1, 0, 0, 1] ;
X = [1, 0, 1, 0, 0, 1], Y = [0, 1, 0, 1, 1, 0] ;
X = [1, 0, 1, 0, 1, 0], Y = [0, 1, 0, 1, 0, 1] ;
X = [1, 0, 1, 1, 0, 0], Y = [0, 1, 0, 0, 1, 1] ;
X = [1, 1, 0, 0, 0, 1], Y = [0, 0, 1, 1, 1, 0] ;
X = [1, 1, 0, 0, 1, 0], Y = [0, 0, 1, 1, 0, 1] ;
```

```
X = [1, 1, 0, 1, 0, 0], Y = [0, 0, 1, 0, 1, 1] ;
X = [1, 1, 1, 0, 0, 0], Y = [0, 0, 0, 1, 1, 1].
```

Case 3 - MAP(2, 7, 5) (2 agents are working in a 7-day week rhythm with a restriction of maximal 5 working days for each of agents.) We define the PROLOG-plan in this case by the following clauses:

```
plan(X,Y):=
X = [X1,X2,X3,X4,X5,X6,X7],
Y=[Y1,Y2,Y3,Y4,Y5, Y6,Y7],
X ins 0..1,
Y ins 0..1,
sum(X, <, 5),
sum(Y, <, 5),
X1 # / Y1,# (X1 #/  Y1),
X2 # / Y2,# (X2 #/  Y2),
X3 # / Y3,# (X3 #/  Y3),
X4 # / Y4,# (X4 #/  Y4),
X5 # / Y5,# (X5 #/  Y5),
X6 # / Y6,# (X6 #/  Y6),
X7 # / Y7,# (X7 #/  Y7),
label([X1,X2,X3,X4,X5,X6,X7,Y1,Y2,Y3,Y4,Y5,Y6,Y7]).
```

These requirements generate a very large list of admissible solutions – over 50 pairs of lists. For example, the following pairs belong to a class of admissible solutions.

```
X = [0, 0, 0, 0, 1, 1, 1], Y = [1, 1, 1, 1, 0, 0, 0] ;
X = [0, 0, 0, 1, 0, 1, 1], Y = [1, 1, 1, 0, 1, 0, 0] ;
X = [0, 0, 0, 1, 1, 0, 1], Y = [1, 1, 1, 0, 0, 1, 0] ;
X = [0, 0, 0, 1, 1, 1, 0], Y = [1, 1, 1, 0, 0, 0, 1] ;
X = [0, 0, 0, 1, 1, 1, 1], Y = [1, 1, 1, 0, 0, 0, 0] ;
X = [0, 0, 1, 0, 0, 1, 1], Y = [1, 1, 0, 1, 1, 0, 0] ;
X = [0, 0, 1, 0, 1, 0, 1], Y = [1, 1, 0, 1, 0, 1, 0] ;
X = [0, 0, 1, 0, 1, 1, 0], Y = [1, 1, 0, 1, 0, 0, 1] ;
.
```

B) Fuzzy cases. In all these examples, PROLOG-solutions were rendered in the form of the appropriate lists of 0's and 1's. In the current cases we extend a list of admissible values modifying their initial sense. At first, assume the following values:

- 0 – for a representation of the fact that an agent A is absent (on a shift),
- 1 – for a representation of a physical absence of the agent A, but a real disposition to be present.
- 2 – for a representation of a physical presence of A, which is only in a partial disposition to work.
- 3 – for a representation of a full disposition of A to work[12].

Let us begin with an exemplary case of two agents: A1 and A2 working 5 days in a week and having four degrees of disposition denoted by 0,1,2 and 3 as above. We denote this problem as MAP(2, 5, 4)^{Fuzz}.

[12]As mentioned, we rather prefer to think about these values as normalized to [0,1] – as $\frac{1}{3}$, $\frac{2}{3}$ etc. instead of 1, 2, or 3. We make use values 0, 1, 2, 3 because of restrictions imposed on PROLOG-syntax.

MAP(2,5,4)^{Fuzz}. This situation may be reflected in the following PROLOG-program (As earlier, the sense of lines of the PROLOG-code is explained on the right side of the programm):

```
plan2(A1,A2) :- A1 = [A1D1,A1D2,A1D3,A1D4,A1D5],      /* list of days of agent A1*/
A2 = [A2D1,A2D2,A2D3,A2D4,A2D5],    /* list of days of agant A2 */
A1 ins 0..3,                          /* Fuzzy degrees of disposition of A1 */
A2 ins 0..3,                          /* Fuzzy degrees of disposition of A2 */
sum(A1, #<, 12),                     /* Restriction on A1-activity during a week */
sum(A2, #<, 9),                      /* Restriction on A2-activity during a week */
sum([A1D1, A2D1], #<, 4), (A1D1 #> 1) # / (A2D1 #> 2),      /* Restriction on D1 */
sum([A1D2, A2D2], #<, 4), (A1D2 #> 2) # / (A2D2 #> 2),      /* Restriction on D2 */
sum([A1D3, A2D3], #<, 4), (A1D3 #> 2) # / (A2D3 #> 2),      /* Restriction on D3 */
sum([A1D4, A2D4], #<, 4), (A1D4 #> 2) # / (A2D4 #> 2),      /* Restriction on D4 */
sum([A1D5, A2D5], #<, 4), (A1D5 #> 2) # / (A2D5 #> 2),      /* Restriction on D5 */
sum([A1D1, A1D2, A1D3], #<, 6),    /* Restrictions on the next 3 days*/
sum([A1D2, A1D3, A1D4], #<, 7),
sum([A1D3, A1D4, A1D5], #<, 6),
sum([A2D1, A2D2, A2D3], #<, 7),
sum([A2D2, A2D3, A2D4], #<, 5),
sum([A2D3, A2D4, A2D5], #<, 7),
label([A1D1,A1D2,A1D3,A1D4,A1D5, A2D1,A2D2,A2D3,A2D4,A2D5]).
```

This PROLOG-solver returns us the following solution-lists:

```
A1 = [2,3,0,3,0]
A2 = [0,0,3,0,3] and
A1 = [2,3,0,3,0]
A2 = [1,0,3,0,3].
```

MAP(2,5,5)^{Fuzz}. Let us slightly modify temporal conditions imposed on work conditions of agents A1 and A2 as follows:

- instead of 0,1,2,3 we consider five values: 0,1,2,3,4 as admissible fuzzy disposition degrees of A1 and A2,
- restriction on A2-activity during a week is relaxed – instead of `sum(A2, #<, 9)` we adopt a condition `sum(A2, #<, 10)`.

In a consequence, we obtain the following program:

```
plan2(A1,A2) :- A1 = [A1D1,A1D2,A1D3,A1D4,A1D5],      /* list of days of agent A1*/
A2 = [A2D1,A2D2,A2D3,A2D4,A2D5],    /* list of days of agant A2 */
A1 ins 0..4,                          /* Fuzzy degrees of disposition of A1 */
A2 ins 0..4,                          /* Fuzzy degrees of disposition of A2 */
sum(A1, #<, 12),                     /* Restriction on A1-activity during a week */
sum(A2, #<, 10),                      /* Restriction on A2-activity during a week */
sum([A1D1, A2D1], #<, 4), (A1D1 #> 1) # / (A2D1 #> 2),      /* Restriction on D1 */
sum([A1D2, A2D2], #<, 4), (A1D2 #> 2) # / (A2D2 #> 2),      /* Restriction on D2 */
sum([A1D3, A2D3], #<, 4), (A1D3 #> 2) # / (A2D3 #> 2),      /* Restriction on D3 */
sum([A1D4, A2D4], #<, 4), (A1D4 #> 2) # / (A2D4 #> 2),      /* Restriction on D4 */
sum([A1D5, A2D5], #<, 4), (A1D5 #> 2) # / (A2D5 #> 2),      /* Restriction on D5 */
```

```
sum([A1D1, A1D2, A1D3], #<, 6),    /* Restrictions on the next 3 days*/
sum([A1D2, A1D3, A1D4], #<, 7),
sum([A1D3, A1D4, A1D5], #<, 6),
sum([A2D1, A2D2, A2D3], #<, 7),
sum([A2D2, A2D3, A2D4], #<, 5),
sum([A2D3, A2D4, A2D5], #<, 7),
label([A1D1,A1D2,A1D3,A1D4,A1D5, A2D1,A2D2,A2D3,A2D4,A2D5]).
```

This PROLOG-solver returns us the following (slightly longer) list of solutions:

```
A1 = [0, 3, 0, 3, 0],
A2 = [3, 0, 3, 0, 3] ;
A1 = [2, 3, 0, 3, 0],
A2 = [0, 0, 3, 0, 3] ;
A1 = [2, 3, 0, 3, 0],
A2 = [1, 0, 3, 0, 3].
```

It emerges that if we admit a new value 5 and exchange 4 for 5 in each place of the program for MAP(2,5,5) – but without any further modification – the PROLOG-solver returns 16 new solutions. It appears that the relaxation of these conditions: `sum(A1, #<, 10)` for `sum(A1, #<, 13)` and `sum(A2, #<, 10)` for `sum(A2, #<, 13)` – preserves the same number of solutions (16).

Further relaxations of temporal constrains usually changes combinatorial explosion of the algorithm relatively quickly. For example, if exchange also a requirement `sum([A1D1, A1D2, A1D3], #<, 6)` for `sum([A1D1, A1D2, A1D3], #<, 13)` we get more than 60 solutions. As a nature of things, we omit their exact presentation.

3.4 Concluding remarks

The computational and programming-wise aspects of fuzziness in the context of Multi-Agent Schedule-Planning Problem have just been discussed. It emerges that the convolution-based representation of fuzzy Allen's relations may support the planning procedure, for example, with respect to a choice of the preferred actions. This representation 'interacts' with planning procedures based on both STRIPS and Davis-Putnam procedure. It may be explained by a very general character of these two procedures as capable of interacting with different supporting tools, methods and forms of representation of fuzzy temporal constraints.

Finally, the PROLOG solver gives a couple of exemplary schedules of MAP dependently of temporal constraints imposed on the initial MAP-situations.

Chapter 4

Temporal Planning with Fuzzy Temporal Constraints and Preferences. The Logic-based Depiction

Inhaltsangabe

4.1 Introduction . . . 134

4.1.1 Motivation of this Chapter . . . 134

4.1.2 Objectives and Novelty of this Chapter. . . . 135

4.1.3 The Problem Definition. . . . 136

4.2 Preliminaries . . . 136

4.3 Multi-Valued Preferential Logic (MVPL) . . . 138

4.3.1 Multi-Valued Preferential Logic (MVPL) – Syntax . . . 139

4.3.2 Interval-based Semantics for MVPL. . . . 139

4.4 Multi-Valued Preferential Halpern-Shoham logic (PHS) and its Subsystem $\mathrm{PHS}^{\mathrm{L}}$ . . . 141

4.4.1 Multi-Valued Preferential HS-logic (PHS). . . . 141

4.4.2 Subsystem $\mathrm{PHS}^{\mathrm{L}}$ for Preferences and (Delayed) Actions. . . . 142

4.4.3 Modeling of the Salesman Situation from PTSP-problem. . . . 143

4.5 Conclusions . . . 145

This chapter[1] is aimed at presenting an alternative logics-based approach to representation and modeling fuzzy temporal constraints and preferences. These entities will be represented now in terms of Preferential Halpern-Shoham logic and modeled (semantically interpreted) by the so-called fibred semantics. We also exchange a subject context of the analysis from Multi-Agent Schedule-Planning Problem for Temporal Traveling Salesman Problem.

4.1 Introduction

In the last chapter a new convolution-based approach to the representation of temporal constraints and preferences was put forward. Furthermore, preferences were defined analytically and their definition required a definition of temporal constrains as a basis definition. Simultaneously, a conceptual inspiration for such an attempt was the Multi-Agent Schedule-Planning Problem. Nevertheless – as earlier observed – Temporal Traveling Salesman Problem (TTSP) is not suitable to be represented in the same way. In fact, it rather require a logical representation. It follows from that fact that a combination of temporal and spatial components plays more significant role in this problem than arithmetic temporal restrictions imposed on agent activity. It is also implied by the fact that preferences should be rather understood in TTSP as relations between objects (chosen cities) – more than as sets of preferred values (as earlier in MAP). In a result, they require a kind of relational semantics.

4.1.1 Motivation of this Chapter

Temporal constraints of Allen's sort are logically expressible in terms of Halpern-Shoham logic – invented by Shoham and Halpern in [59]. This temporal-modal system - broadly developed from metalogical point of view in such works as: [64, 66, 65, 108, 63, 109, 110] – constitutes the only one formalism capable of representing temporal constraints (Allen's relations in particular) up today. In addition, other logical approaches to temporal constraints representation exploit Halpern-Shoham logic as their basis [67][2]. For that reason, this logical system forms a comfortable foundation of the current approach. Unfortunately, this formalism is involved in the following difficulties to overcome.

D1 Halpern-Shoham logic (in generality) is too excessive from the application point of view.

D2 We have a restricted knowledge about operational side of Halpern-Shoham logic (HS) – some knowledge about HS restricted to operators $A\bar{A}B\bar{B}$ might be found in [109].

D3 HS itself – even in the extended epistemic version from [67] – rather refers to regular and operationally ideal comfortable situation of the agent's behavior, so it is suitable to be interpreted by an excessively ideal process semantics of Honda-Fagin – as in: [112, 113].

Meanwhile, preferences –as uninterpretable in this "ideal" and operationally convenient way – should be interpreted in a slightly modified Honda-Fagin's interval-based semantics. These difficulties motivate us to restrict HS and modify the agent's process semantics.

However – modeling TTSP – we need to represent and interpret combined entities such as: fuzzy temporal constraints and preferences. It requires the appropriate combined semantics. On the one hand, it should not impose any additional and excessive conditions in syntax of the proposed system (such as product semantics does). On the second one, it should be capable of interpreting combined, mixed formulas. For that reason, the so-called fibring(fibred) semantics –invented by D. Gabbay and A. Kurucz in [114, 115]– seems to be the

[1] Considerations from this chapter were contained in [106] and presented during *World Computational Congress* in the framework of the conference 'Fuzzy Sets and Systems' (FUZZ-IEEE) organized on 24-31th July 2016 in Vancouver. Some of results were earlier published in [107] and presented during the conference INISTA in Madrid in September 2015.

[2] One needs to underline that an idea to combine epistemic operators with timporal logic system is relatively rare. In [67], an epistemic Halpern-Shoham logic was elaborated. In [111], a logic for knowledge, correctness and time was proposed.

appropriate one. Nevertheless, fibred semantics in this original depiction has the following difficulty from our point of view:

D4 It forms a kind of *a point-wise semantics*, but we need the interval-based semantics.

This difficulty motivates us to adopt Gabbay's and Kurucz's idea to propose the interval-based semantics.

The problem how to logically represent fuzziness of temporal constraints seemingly is clear: one need to exploit a fuzzy logic system. Unfortunately, fuzzy logic systems – such as in [79] – suffer from the following difficulties from our point of view. Namely,

D5 Fuzzy logic systems are usually interpreted in algebraic semantics, but HS - in the relational one.

D6 Many tools – typical for modal logic – such as: satisfiability conditions, Kripke-based frames, etc. have not elaborated in a case of fuzzy logic systems yet.

D7 Fuzzy logic – partially as HS itself – is not suitable to represent fuzzy intervals as geometric objects in $\mathbb{R}^2$.

These 3 difficulties motivate us to built a compromise between fuzzy and modal logic and to adopt some solutions for modal logic systems. In results, we introduce a kind of a multi-valency instead of fuzziness. Since fuzziness is not directly applicable to temporal constrains (fuzzy logic is not directly adaptable to HS), we introduce fuzziness by preferences. This 'transfer' of fuzziness from fuzzy intervals for preferences allows us to consider one-dimensional discrete intervals (logically representable and interpretable) instead of fuzzy intervals as $\mathbb{R}^2$-objects.

4.1.2 Objectives and Novelty of this Chapter.

In line with the above motivation factors, this chapter has the following objectives:

Obj1 To state a multi-valued extension of the HS logic (PHS) suitable for representation of temporal constraints and preference,

Obj2 To describe how to extract subsystems from PHS for a use of temporal reasoning,

Obj3 To propose the *interval-based fibring semantics* for interpretation of combined formulas of PHS,

Obj4 To adopt this type of semantics for modeling some aspects of the Preferential Traveling Salesman Problem.

Furthermore, some subsystem of PHS – suitable for a representation of preferences and actions with delay – will be abstracted and called later by PHL^L. According to this list of chapter objectives its *novelty* consists in:

N1 proposing a new multi-valued extension of Halpern-Shoham logic (PHS) for representation of Allen's relations and preferences,

N2 specifying a method of abstraction of some subsystems of PHS,

N3 generalizing the Gabbay's *point-wise fibring semantics* to the *interval fibring semantics*.

N4 approximate a method of exploiting of this interval fibring semantics for modeling problems of the class of Traveling Salesman Problem.

4.1.3 The Problem Definition.

As it has been already said – current investigations will refer to the *Preferential Traveling Salesman Problem* as a subject of a semantic modeling in terms of the proposed interval fibring semantics. For that reason, let us consider the following simplified aspect of *Traveling Salesman Problem*.

Example 29 *Consider a salesman K and a list of n cities (with temporally measured distances between them). Assume, as usual, that K is visiting all cities (from a given list) in such a way to find the shortest possible route that visits each city exactly once and leads to the origin city. Assume also that K is in a city C_1 in some temporal interval I_1 and must deliver some packages from C_1 to other cities. Assume that K decided to begin with a city C_2 (its strong preference) and a temporal distance between C_1 and C_2 amounts 3 hours and a package, say A, must be delivered in C_2 in some temporal interval I_2. Thus, a situation of K can be rendered by the following expression with a combined modal prefix*[3]

$$[\text{K strongly prefers}_{C_1}]\langle\text{Later in 3hours}\rangle\text{Deliver}^{A}_{C_2}. \tag{4.1}$$

The outer operator [K(strongly)prefers]ϕ plays a role of a box-type operator for representation of preference of K and $\phi = \langle$ Later in 3 hours $\rangle\psi$ plays a role of (an additionally specified) $\langle L\rangle$-operator of HS logic. Finally, $\psi = \text{Deliver}^{A}_{C_2}$ is already a unique atomic formula.

In this perspective, this chapter objective is to give an outline of a semantic interpretation of the Salesman's situation (expressed by the above formula) in terms of fibred semantics[4].

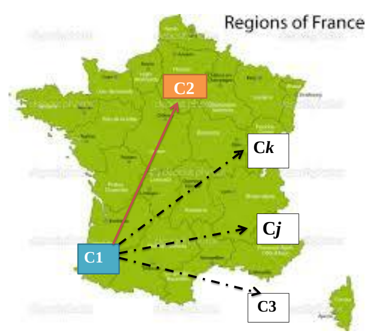


Figure 4.1: Illustration of the Traveling Salesman Problem in the considered case.

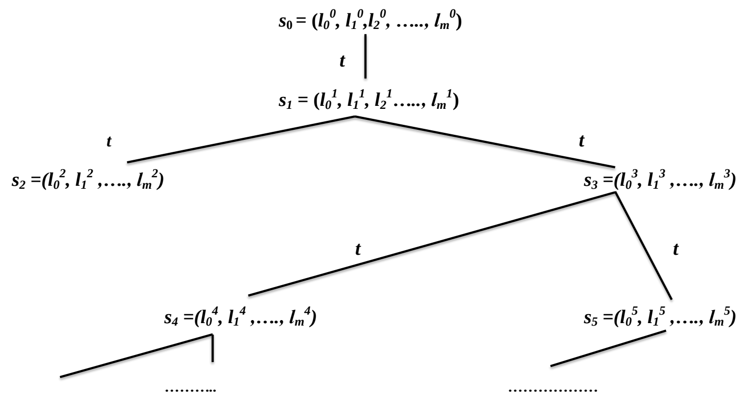
4.2 Preliminaries

In the framework of a terminological background of the analysis we adopt some definitions from [67], modifying some conditions for the so-called labeling function. We adopt the so-called homogeneity principle from [109] w. r. t. this function – as we intend to be equipped in a knowledge about the whole structure of intervals in semantics of MVPL for preferences instead of a knowledge about their endpoints only.

An interval-based semantics. We begin our interval-based semantics presentation with a definition of a tree-like order. We say that $\langle S, \leq\rangle$ is a tree-like order iff:

[3]More precisely, a sentence: "K strongly prefers to begin with city C2 and the temporal distance between C1 and C2 is 3 hours and package A must be delivered in C2" – is implied by the considered formula.

[4]Note that this task is different from a solving of this variant of TSP-problem in a sense of combinatorial optimization.



Examples of intervals: $I_1 = s_0 s_1 s_2$, $I_2 = s_0 s_1 s_3 s_5$, $I_3 = I_2 = s_0 s_1 s_3 s_4$

Figure 4.2: An example of IBIS-system

- it is strongly discrete, i.e. there are only finitely many points between any two points and the the order contains the least element ,
- for any a, b, $c \in$ S, if $a \leq c$ and $b \leq c$, then either $a \leq b$ or $b \leq a$.

We also consider a finite set $\mathcal{A} = \{1, \ldots, m\}$ of agents such that each agent i is endowed with a set of global states $L_i, i \in \mathcal{A}$ and set of local actions which produce a transition relation t. In such a framework we adopt the following definitions.

Definition 24 *Let $\mathcal{L}an$ be a propositional language with a set of propositional variables* Prop*. An* Interval-Based Interpreted System *(IBIS) is a tuple (S, s_0, t, Lab) such that*

- *$S \subseteq L_1 \times \ldots\ L_m$ is a set of tuples of global states reachable from the initial global state s_0 via t ,*
- *$t \subseteq S^2$ is a transition relation between states such that $\langle S, t\rangle$ forms a tree-like order with s_0 as its least element ,*
- *Lab: $S \mapsto 2^{\mathrm{Prop}}$ is a labeling function, which for $I = s_1, s_2, \ldots, s_k$ is defined as follows: $Lab(I) = \phi \in$* Prop*, for $i \leq k$ provided that the following homogeneity principle is true: $\phi \in Lab(I)$ if and only if $\phi \in Lab(s_1), \phi \in Lab(s_2), \ldots, \phi \in Lab(s_k)$.*

The homogeneity principle ensures that the same formula ϕ labels the whole interval I and each internal point of it.

Definition 25 *An* interval *is a finite path in* IBIS*, or a sequence $I = s_1 s_s \ldots s_k$ such that $s_i t s_{i+1}$ for $1 \leq i \leq k-1$, and a transition t. A restriction of an interval I (with a length l) to a sequence of its first k-states (for $k \leq l$) will be denoted by $I|_k$ and called a k-prefix of I.*

Note that this homogeneity principle ensures that a proposition letter p holds over an interval I if it holds over all its subintervals – due to ideas from [109].

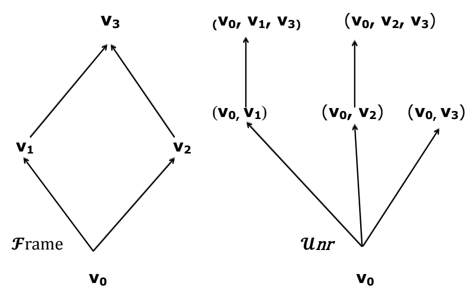


Figure 4.3: $\mathcal{U}$nr is the unraveling of $\mathcal{F}$

Definition 26 *An IBIS-system is an unraveling of the generalized Kripke frame.*

We also assume that each global state s from S of *IBIS* consists of local states. For a global state $s = (l_1, l_2 \ldots, l_m)$ the local state $l_i \in L_i$ in a global state s will be denoted by $l_i(s)$, for $i \in \mathcal{A}$.

Example 30 *Consider some IBIS-system with an agent 1, which is endowed with a global state* $L = \{l_A, l_B, l_C\}$ *with three local states and with a 2-elemental set of actions* $\{a_1, a_2\}$. *As an example of the transition one can consider:* $t = \{\langle l_A a_1 l_A\rangle, \langle l_A a_2 l_B\rangle, \langle l_B a_2 l_C\rangle\}$.

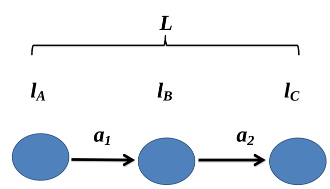


Figure 4.4: An illustration of *IBIS* from Example 27

The mutual relationship between an *IBIS*-ontology and epistemology of agents can be explained as follows. Intervals states I and $I^{'}$ with a length k and l (resp.) are indistinguishable by an agent (formally: $I \sim_i I^{'}$) if and only if $k = l$ and $l_i(s_j) = l_i(s_j^{'})$, for all $j \leq k$, $s_j \in I$, $s_j^{'} \in I^{'}$, i.e. the corresponding local states s_j and $s_j^{'}$ are identical. This property will be called – due to [112, 113] – *the behavioral equivalence* and it plays a role of a meta-assumption w.r. t. MVPL, introduced later.

4.3 Multi-Valued Preferential Logic (MVPL)

It has been already said in paragraph 2.6 of chapter 1b that we adopt the Davidson's convention to interpret preferences as transitive, reflexive and anti-symmetric relations. In this way, we adopt a "weaker" convention of the interpretation of preferences – as in [83]. For that reason, preferences will be syntacticly expressed by modal logic S4 – as just interpretable by partial orders[5].

[5]In essence, models based on Kripke frames with partial orders as its accessibility relations do not exhaust the whole class of possible S4-models. For example, S4 is complete w. r. t. any dense-in-itself metric space – see:[116]. Thus, we do not get

4.3.1 Multi-Valued Preferential Logic (MVPL) – Syntax

It has been said that preferences are intended to be interpreted in an interval-based semantics as partial orders in MVPL and in a fuzzy manner. This postulate will be reflected in two types of operators: of a $\Box$-type $[\mathrm{Pref}]_i^\alpha\phi$ read: "an agent i (strongly) prefers ϕ with a degree α belonging to a finite set $G \subset [0,1]$, and $\langle\mathrm{Pref}\rangle_i^\alpha\phi$ "an agent i weakly prefers ϕ with a degree α" as a $\Diamond$-operator.

Language of MVPL. The language of MVPL, $\mathcal{L}(MVPL)$, is given by a grammar:

$$\phi \ := \ p \,|\, \neg\phi \,|\, \phi \wedge \psi \,|\, [\mathrm{Pref}]_i^\alpha\phi \,|\, \langle\mathrm{Pref}\rangle_i^\alpha\phi\,,$$

where $i \in \mathcal{A}$, $\alpha \in G$ and G is a finite subset of $[0,1]$. The following definitions and axioms are adopted in the MVPL-syntax.
Def.: $[\mathrm{Pref}]_i^\alpha\phi \iff \neg\langle\mathrm{Pref}\rangle_i^\alpha\neg\phi$ for each $\alpha \in G$.

Axioms: The axioms of MVPL are:

axioms of Boolean propositional calculus

axiom K $[\mathrm{Pref}]_i^\alpha(\phi \to \chi) \to ([\mathrm{Pref}]_i^\alpha\phi \to [\mathrm{Pref}]_i^\alpha\chi)$

axiom 4 $[\mathrm{Pref}]_i^\alpha\phi \to [\mathrm{Pref}]_i^\alpha[\mathrm{Pref}]_i^\alpha\phi$

axiom T $[\mathrm{Pref}]_i^\alpha\phi \to \phi$

As *inference rules* we adopt *Modus Ponens*, substitution and a necessitation rule for the $[\mathrm{Pref}]_i^\alpha$-operator: $\frac{\phi\to\psi}{[\mathrm{Pref}]_i^\alpha\phi\to[\mathrm{Pref}]_i^\alpha\psi}$ for each $\alpha \in G \subset [0,1]$, G is finite.

For a use of further analysis we will consider α as restricted to a finite $G \subset [0,1]$ (even if it is not mentioned) – due to [117, 79] and to our arrangements – in order to ensure a context of a multi-valued logic. All these arrangements leads to the following definition of MVPL.

Definition 27 *MVPL is defined as the smallest theory in $\mathcal{L}(MVPL)$, which contains all axioms (above listed) and closed on the above inference rules.*

4.3.2 Interval-based Semantics for MVPL.

The introduced MVPL with preferential operators will be interpreted now in an interval-based semantics. A "core" of this semantics construction is to introduce the appropriate accessibility relation between intervals, denoted later by $\precsim_i$ and to specify it by α in the next construction stage.

Accessibility relation $\precsim_i$. For this reason, assume that intervals $I = s_1s_2\ldots s_k$ and $I^{'} = s_1^{'}s_2^{'}\ldots s_l^{'}$ are given for some k,l and establish also an agent $i \in \mathcal{A}$. Let us define a new accessibility relation $\precsim_i \subseteq \mathcal{P}(S\times S)$ between I and $I^{'}$ as follows: $I \precsim_i I^{'} \iff k \leq l$ and $l_i(s_j) = l_i(s_j^{'})$ for all $j < k$, i.e. an agent i cannot distinguish between the corresponding states of I and $I^{'}$ up to j, between j-prefixes of both intervals. In other words, $I \precsim_i I^{'} \iff I|_j \sim_i I^{'}|_j$, or if and only if the behavioral equivalence condition holds for j-prefixes of both intervals.

Accessibility relation $\precsim_i^\alpha$. In order to grasp the similarity degree between I and $I^{'}$ introduce a new relation between these intervals, denoted later by $\precsim_i^\alpha$. Thus, establish a finite $G \subset [0,1]$ and introduce some new function $\|\bullet\| : \mathcal{P}(S\times S)\times\mathcal{A} \mapsto \mathcal{G}$ defined as: $\|I \precsim_i I^{'}\| = \alpha$, for $\alpha \in G$. Intuitively, this new function

"only" partial ordered models, but some more models as the unintended ones. These "unintended" part of S4-semantics is, however, redundant from our point of view and our requirements with respect to preferences.

associates the earlier relation $I \precsim_i I^{'}$ to some α from G. The appropriate methods of achieving of α will be discussed later. Independently of this, one can define a desired relation $\precsim_i^{\alpha}$ in the product $\mathcal{P}(S \times S) \times G \times \mathcal{A}$ in the following way: $I \precsim_i^{\alpha} I^{'} \iff \|I \precsim_i I^{'}\| = \alpha$. It remains to describe how α can be found. A possible method is presented in the example below.

Example 31 *Consider a pair of discrete intervals* $(I_1, I_2) : I_1 \precsim_i I_2$ *having* $j = 10$ *common points and define* $\alpha = \|I_1 \precsim_i I_2\| = \frac{j(I_1,I_2)}{K}$ *for some* $K = 200$ *¿ length of each of intervals and each* $i \in \mathcal{A}$. *Thus, their similarity* $\alpha = \|I_1 \precsim_i I_2)\| = \frac{10}{200} = \frac{1}{20}$. *Moreover,* $\|I_1 \precsim_i I_2\| = \frac{1}{20} = \|I_1 \precsim_k I_2\|$ *for another agent* $k \in \mathcal{A}$, *since the common j-prefix of both intervals is equally recognized by both agents because of the behavioral equivalence assumption for all agents in* $\mathcal{A}$.

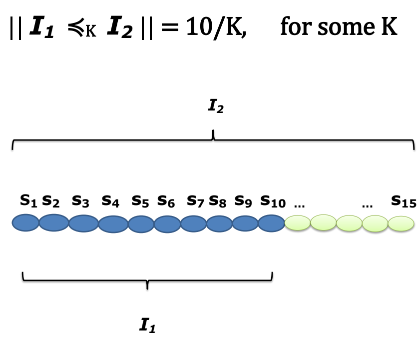


Figure 4.5: Two discrete intervals I_1 and I_2 having 10 common points with an example of computing a value $\|I_1 \precsim_K I_2\|$ for some K.

Definition 28 *(Satisfaction.) Given a formula* $\phi \in \mathcal{L}(MVPL)$ *with a set of propositions* Prop, *an IBIS, an interval* I, *a labeling function Lab, and a finite set* $G \subset [0, 1]$ *we define inductively the fact that* ϕ *is satisfied in IBIS and in an interval* I *(symb.*$I \models \phi$*) as follows:*

- *for all* $p \in$ Prop, *we have IBIS,* $I \models p$ *iff* $p \in Lab(I)$.
- *IBIS,* $I \models \neg\phi$ *iff it is not such that IBIS,* $I \models \phi$.
- *IBIS,* $I \models \phi \wedge \psi$ *iff IBIS,* $I \models \phi$ *and IBIS,* $I \models \psi$.
- *IBIS,* $I \models [\mathrm{Pref}]_i^{\alpha}\phi$, *where* $i \in A$, *iff for all* $I \precsim_i^{\alpha} I'$ *we have IBIS,* $I' \models \phi$ *for each* $\alpha \in G$.
- *IBIS,* $I \models \langle\mathrm{Pref}\rangle_i^{\alpha}\phi$, *where* $i \in A$, *iff there is I' such that* $I \precsim_i^{\alpha} I'$ *and IBIS,* $I' \models \phi$ *for each* $\alpha \in G$.

The key clause in the above definition is this one referring to the modal operators $[\mathrm{Pref}]_i^{\alpha}\phi$ and $\langle\mathrm{Pref}\rangle_i^{\alpha}\phi$. These conditions assert that such modal formulas are satisfied in an interval I and model IBIS iff the same formula ϕ holds in all intervals accessible from this I *via* $\precsim_i^{\alpha}$-relation.

Relation $\precsim_i^{\alpha}$ for representation of preferences. Nevertheless, not all relations $\precsim_i^{\alpha}$ are suitable to represent preferences – intentionally represented by partial orders (MVPL forms a unique S4-system). It is not difficult to observe that belonging to the appropriate relation class depends on a method of α-achieving. For example, $\alpha = \frac{j}{K}$ ensures reflexivity and transitivity of $\precsim_i^{\alpha}$, but unfortunately – symmetry. In order to ensure a desired anti-symmetry for $\precsim_i^{\alpha}$ one need distinguish the relation $I_1 \precsim_i^{\alpha} I_2$ from its inverse relation by means of α-degree like in this example.

Example 32 $\alpha = \begin{cases} \frac{1}{jK} & \textit{if length}(I_1) \leq \textit{length}(I_2),\ 2 \leq j \\ \frac{j}{K} & \textit{if length}(I_1) > \textit{length}(I_2),\ 2 \leq j \end{cases}$

Colollary 3 *The accessibility relation $I_1 \precsim_i^\alpha I_2$ defined as above forms a partial order.*

Proof: Reflexivity is obvious. Anti-symmetry follows from the fact that if $I \precsim_i^\alpha I^{'}$ holds, then does not hold $I^{'} \precsim_i^\alpha I$ as $\alpha = \frac{1}{jK}$ holds in mutually disjoint cases.

In order to show transitivity assume that $I_1 \precsim_i^\alpha I_2$ and $I_2 \precsim_i^\beta I_3$ and $\alpha = \frac{1}{jK}$ and $\beta = \frac{1}{JK}$ for some established K and j – denoting a number of common points of I_1 and I_2 and J – denoting a number of common points of I_2 and I_3 (*resp.*). Thus, obviously, $I_1 \precsim_i^A I_3$, where A $= \frac{1}{jK}$ as $I_1(\subseteq I_2 \subseteq I_3)$ has just j common points with I_3. Hence, $I_1 \precsim_i^\alpha I_3$. The same reasoning may be repeated for α defined alternatively. $\square$

Remark 2 *Note that the assumption $2 \leq j$ is essential in the above example. To illustrate this fact let us consider $I_1 = s_1$ and $I_2 = s_1 s_2$, so $j = 1$ and – in result – $\frac{1}{jK} = \frac{1}{2}$ and $\frac{j}{K} = \frac{1}{2}$, what implies that $\precsim_i^\alpha$ is symmetric.*

We do not decide on the concrete solution, we only advocate the most "ergonomic" solutions from the point of view of an 'epistemic behavior' of agents – which only base on an agent recognition of identity between interval prefixes in order to determine α– as suggested in [112, 67]. In this way, authors give vent to their meta-epistemic minimalism w.r. t. the agent role.

In this way we have just obtained a semantic interpretation of the sentence "an agent i (weakly) prefers ϕ with a degree α". It exactly means that there exists (at least one) such a pair of α-similar intervals having identical prefixes, which are observational recognized by an agent i as such ones.

4.4 Multi-Valued Preferential Halpern-Shoham logic (PHS) and its Subsystem PHS$^{\mathrm{L}}$

4.4.1 Multi-Valued Preferential HS-logic (PHS).

In last section, a preferential system MVPL was introduced. In this subsection, a new hybrid system – called later: "(Multi-valued) Preferential HS logic" and denoted by PHS – is proposed. This system is conceived to contain both preferential operators and the typical (modal) temporal operators of HS. PHS will be characterized both syntactically and semantically, but it is not considered as a precisely defined axiomatic system, but rather as an initial *reservoir* of subsystems. For some of PHS-subsystems we will propose concrete types of models.

Definition 29 *(Syntax of a (Multi-Valued) Preferential PHS.) The syntax of Preferential HS logic (PHS) is defined by*

$$\phi \ := p \,|\, \neg\phi \ \,|\, \phi \wedge \psi |[\mathrm{Pref}]_i^\alpha \phi \ \,|\, \langle \mathrm{Pref} \rangle_i^\alpha \phi \ \,|\, \langle X \rangle \phi \ \,|\, \langle \bar{X} \rangle \phi \,,$$

where p is a propositional variable, $i \in \mathcal{A}$, where $\mathcal{A}$ is a set of agents, $\alpha \in G \subset [0,1]$ and G is finite. X is one of Halpern-Shoham relations.

Definition 30 *(Satisfaction.) Let $\phi \in \mathcal{L}(\mathrm{PHS})$, IBIS and an I as earlier. Then we define the satisfaction relation for $[\mathrm{Pref}]_i^\alpha$ operator as presented earlier (def. 5) and for Halpern-Shoham operators by the condition: IBIS, $I \models \langle X \rangle \phi$ iff there is such an interval I' that $I\, X\, I'$ and IBIS, $I' \models \phi$.*

4.4.2 Subsystem $\mathrm{PHS^L}$ for Preferences and (Delayed) Actions.

In this section, we take into account another, more practically motivated criterion of choosing of HS-subsystems than their meta-logical features. We intend to single out a subsystem of preferential HS capable of describing preferences and actions with delay, as pointed out in the example with PTSP-problem.
Syntax of $\mathrm{PHS^L}$. The language $\mathcal{L}(\mathrm{PHS^L})$ includes the following components:

$$Prop \mid \phi \to \chi \mid \neg\phi \mid \langle \mathrm{Pref} \rangle_i^\alpha(\phi) \mid \langle L \rangle \phi \,, \mid [\mathrm{Pref}]_i^\alpha \phi \mid [L]\phi \,,$$

where $Prop$ is a countable set of propositional variables of $\mathcal{L}(\mathrm{PHS})$, $\alpha \in G \subset [0,1]$, where G is finite, and $\langle \mathrm{Pref} \rangle_i^\alpha(\phi), \langle L \rangle \phi$ and their duals are defined as earlier.

Let $\mathcal{L}^{\mathrm{Pref}}$ denote a preferential component of $\mathcal{L}(\mathrm{PHS})$ and $\mathcal{L}(\mathrm{HS}^L)$ as a language of HS restricted to the'**L**ater'-operator. In this system combined bi-modal formulas of diamond- and box-type are admitted – such as: $\langle \mathrm{Pref} \rangle_i^\alpha \langle L \rangle \phi$, $[\mathrm{Pref}]_i^\alpha \langle L \rangle \phi$. One component of them belongs to $\mathcal{L}^{\mathrm{Pref}}$, the other one belongs to the $\mathcal{L}(\mathrm{HS}^L)$.

Obviously, this distinction for combined and non-combined formulas should be reflected in the proposed semantics for PHS^L. Namely, a classical modal *Kripke-based semantics* will be introduced for combined formulas in a form of the *fibring semantics*.

Fibring interval-based semantics. In this section we demonstrate how the mechanism of fibred semantics works with respect to combined modalities that cannot be modeled by models $\mathcal{M}_1$ and $\mathcal{M}_2$, which – in general case – cannot recognize modal operators interpreted in the second model [6]. Consider therefore:

- a formula $\psi = \langle Pref \rangle_i^\alpha \langle L \rangle \phi$ and
- a model $\mathcal{M}_1, I^1 \models \langle Pref \rangle_i^\alpha p$, where p atomic.

It can arise a natural question what about satisfiability of an atomic formula p in $\mathcal{M}_1, I^1$, if $p = \langle L \rangle \phi$. More precisely:

- what about $\mathcal{M}_1, I^1 \models p$, if $p = \langle L \rangle \phi$?

As it has been mentioned – in a general case it may hold the following case:

- $\mathcal{M}_1, I^1$ can not to 'recognize' $p = \langle L \rangle \phi$!

This difficulty generates a natural question how to deal with this fact?We face this question now.

Fibring mapping. In order to evaluate $[L]\psi$ at $\mathcal{M}_1$ one need some mapping between $\mathcal{M}_1$ and $\mathcal{M}_2$ in order to transfer the validity checking from this first model to the validity checking within the second one. Thus, we introduce such a new function – called a *fibring mapping* $\mathbf{F}$ – that $\mathbf{F}(\mathcal{M}_1, I_1) = (\mathcal{M}_2^{I_1}, I_2)$ and the following equality holds:

$$\mathcal{M}_1, I_1 \models [L]\psi \iff \mathcal{M}_2^{I_1}, I_2 \models [L]\psi$$

for some interval I_2 of $\mathcal{M}_2$.

Since a model $\mathcal{M}_2$ is characterized by the interval I_2, we can identify $\mathbf{F}(I_1)$ with this new interval I_2 of the associated model $\mathcal{M}_2^{I_1}$. It allows us to formulate a new satisfaction condition in the form:

$$\mathcal{M}_1, I_1 \models [L]\psi \iff \mathcal{M}_2^{I_1}, \mathbf{F}(I_1) \models [L]\psi$$

As in a point-wise fibring semantics, one can impose on the fibred mapping $\mathbf{F}$ a condition of "switching semantics", i.e. for each $I \in \mathcal{M}_1$, it holds $\mathbf{F}(I) \in \mathcal{M}_2$ and for each $I \in \mathcal{M}_2$: $\mathbf{F}(I) \in \mathcal{M}_1$. We also assume that if $I_1 \neq I_2$, than also $\mathbf{F}(I_1) \neq \mathbf{F}(I_2)$. ($\mathbf{F}$-images of two different intervals are different, too.) The

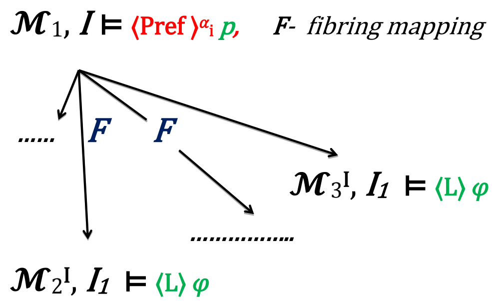


Figure 4.6: A fibring mapping which should not necessary be a function

same procedure can be repeated for other type of mixed formulas of $\mathcal{L}(\mathrm{PHS^L})$. (A slightly more detailed introduction of fibring semantics will be presented in Appendix.)

Fibred models and fibred satisfaction. Assume that the following interval-based Kripke models are given:

•$\mathcal{M}_1 = \langle S_1, R_1, h_1, \mathbf{F} \rangle$, where: $S_1 = \{J_1, J_2 \ldots, J_{2k}\}$ for some fixed $2k$, $J_j R_i J_l \iff J_j \precsim_1^{\alpha} J_l$ in IBIS (for $j, l \in \{1, \ldots, 2k\}, i \leq m$), h_1 is an assignment function and $\mathbf{F}$ is the fibring mapping defined as above;

•$\mathcal{M}_2 = \langle S_2, L, h_2, \mathbf{F} \rangle$, where: $S_2 = \{I_1 \ldots I_{2k}\}$ for some fixed $2k$, L is a "Later"-relation between intervals from S_2, h_2 and $\mathbf{F}$ are defined as earlier. Then a *fibred models* $\mathcal{M}$ will be defined as a tuple:

$$\mathcal{M} = \langle S_1 \otimes S_2, R_L \otimes R_i, h_1 \otimes h_2, \mathbf{F} \rangle, \tag{4.2}$$

where: $\mathbf{F}$ is the same fibring mapping and $\otimes$ denotes a simple sum or a fusion of the appropriate components. It means that elements of $\mathcal{M}$ cannot be 'mixed' – as in a usual set-theoretic sum of components – without considering the fibring mapping $\mathbf{F}$.

Definition 31 *Let $\mathcal{M}$ be such a model. The* `satisfaction` *condition for a formula $\phi = \langle \mathrm{Pref} \rangle_i^{\alpha} \langle L \rangle \psi$ in the fibred model $\mathcal{M}$ is put forward as follows:*

$$\begin{aligned} (\mathcal{M}, \boldsymbol{F}, I) &\models \langle \mathrm{Pref} \rangle_i^{\alpha} \langle L \rangle \psi \\ \iff \exists I_1 (I &\precsim_i^{\alpha} I_1 \text{and } \mathcal{M}_1, I_1 \models \langle L \rangle \phi) \\ \iff \exists I_1 (I &\precsim_i^{\alpha} I_1 \text{and } \mathcal{M}_2, \boldsymbol{F}(I_1) \models \langle L \rangle \phi) \\ \iff \exists I_1 (I &\precsim_i^{\alpha} I_1 \text{and } \exists I_2 (\boldsymbol{F}(I_1) \boldsymbol{L}ater I_2 \text{ and } \mathcal{M}_2, I_2 \models \phi). \end{aligned}$$

The satisfaction conditions for other mixed formulas are similar.

4.4.3 Modeling of the Salesman Situation from PTSP-problem.

Return now to the Traveling Salesman Problem with our salesman K, which intends to deliver a pocket A from a city C_1 to C_2 (his strong preference). Because of a temporal distance between C_1 and C_2 his preference could be satisfied not earlier than 3 hours and in some interval in the city C_2. It has been mentioned that the fact that a pocket A can be *potentially delivered* in C_1 can be rendered by a modal formula:

$$[\text{K (strongly) prefers}] \langle \text{Later in 3 hours} \rangle \text{Deliver}^{A}_{C_2}. \tag{4.3}$$

[6]The 'preferential' models do not recognize operators of HS^L and models for '**L**ater' do not recognize the 'preferential' operators.

(read: "K strongly prefers, later in 3 hours to deliver a pocket A to a city C_2".) Recall that the outer operator [K(strongly)prefers]ϕ plays a role of a box-type operator for representation of preference of K and $\phi = \langle$ Later in 3 hours $\rangle\psi$ plays a role of (an additionally specified) $\langle L \rangle$-operator of HS logic. Finally, $\psi = \text{Deliver}^A_{C_2}$ is already a unique atomic formula.

Model for the preferential component. In order to find the appropriate model for a preferential component of the formula (2) assume that P and $P^{Deliver}$ are some discrete intervals interpreting the Salesman's preference such that:

- P is an interval where the preference is "expressed" and
- $P^{Deliver}$ is the interval, which the subject of the salesman's preference is materialized in. Formally: $P^{Deliver} \models \text{Deliver}^A_{C_2}$, or a fact of delivering of a packet A to C_2 holds in this interval.

Assume also that $\precsim_K^{strongly}$ is an accessibility (preference) relation between them, i.e. it holds $P \precsim_K^{strongly} P^{Deliver}$. Thus, a model for the preferential component is given as follows:

$$\mathcal{M}_1 = \langle \{P, P^{Deliver}\}, \precsim_K^{strongly}, h_1 \rangle \tag{4.4}$$

for some valuation h_1.

Model for the temporal component. Similarly, we find a model for a temporal component. For that reason consider two *temporal discrete* intervals:

- I_1 for a representation of "now" and,
- I_2 for a representation of "sometimes in a future"[7].

Thus, the appropriate model in this case can be given as follows:

$$\mathcal{M}_2 = \langle \{\text{now}, \text{sometimes in a future}\}, \text{Later in 3 hours}, h_2 \rangle \tag{4.5}$$

for some valuation h_2. The *fibred model* for the whole formula (2) is determined by the tuple:

$$\mathcal{M} = \Big\langle S, R^*, h, \mathbf{F} \Big\rangle, \tag{4.6}$$

where:

- $S = \{P, P^{Deliver}\} \otimes \{I_1 = \text{now}, I_2 = \text{sometimes in a future}\}$,
- $R^* = \{\precsim_K^{strongly}\} \otimes \{\text{Later in 3 hours}\}$,
- $h = h_1 \otimes h_2$,
- $\mathbf{F}(P^{Deliver}) = I_1(= \textit{now})$.[8]

[7] We put aside a fact that such phrase may be naturally interpreted by continuous time interval. Because of a nature of current formal considerations we are only interested in a discrete-interval based interpretation.

[8] ($\mathbf{F}$ joins these two intervals such that the last preferential interval is connected with the first temporal one).

4.5 Conclusions

In this chapter a new multi-valued extension of Halpern-Shoham logic PHS for describing of preferences and Allen's temporal relations has just been introduced. The proposed extension PHS played a double role in the proposed approach. At first, it axiomatically described preferences and allows us to interpret them semantically in the interval Kripke frame-based semantics. Furthermore, combined formulas of PHS were interpreted in interval fibred semantics. This type of relational semantics seems to be convenient for such a formalism. In fact, it is stronger than a semantics for fusions of systems, but weaker than the product semantics that requires some additional conditions – redundant from our point of view.

Despite of this fact, the fibred semantics – in its original depiction – seems to be too 'static' and weakly operationally flexible in order to be directly exploited to temporal planning problems. Therefore, the fibring semantics will not be incorporated in next chapter directly, but as a reservoir of useful tools and concepts in the construction of the hybrid plan controller. More precisely, we adopt the following 'components' of this 'reservoir' of fibred semantics:

- Interval-Based Transition System (IBIS) in a basis of a construction of automata for this plan controller,
- An idea of a fibring mapping **F** as a 'communicator' between constructed automata,
- Halpern-Shoham logic – restricted to operators: L and D and interpreted in fibring semantics – as a linguistic support for Linear Temporal Logic in a specification of robot motion and environment.

Chapter 5

A General Method of the Hybrid Controller Construction for Temporal Planning with Preferences

Inhaltsangabe

5.1 Introduction . . . 147

5.2 Motivation of Current Analyses . . . 147

5.2.1 Objectives and Novelty of this Chapter. . . . 147

5.3 Preliminaries . . . 148

5.4 Problem Formulation and a General Algorithm of the Controller Construction 150

5.5 The Controller Construction for Temporal Planning . . . 151

5.5.1 From Triangulation to the Finite Transition System (FTS) . . . 151

5.5.2 From FTS System to the First Büchi Automaton . . . 153

5.5.3 From LTL and $HS^{L,D}$ to the Second Büchi Automaton . . . 153

5.5.4 Product Automaton $\mathcal{A}_{FTS} \times \mathcal{A}_{LTL,HS^D}$ . . . 155

5.6 Part II: Implementation . . . 156

5.6.1 The Robot Environment and its $LTL \bigcup HS^{D,L}$-specification . . . 156

5.6.2 From $LTL \cup HS^L$ to the Second B'uchi Automaton . . . 158

5.6.3 The Product Automaton . . . 159

5.6.4 PROLOG-description of the Product Automaton . . . 163

5.7 Conclusions . . . 163

Abstract. This chapter[1] describes a general method of the Hybrid Controller Construction for Temporal Planning with Preferences. This construction consists of many steps and begins with triangulation of the robot motion environment by its specification by automata and in terms of Linear Temporal Logic up to the construction of the appropriate product automaton and its description in PROLOG.

5.1 Introduction

The previous chapter described a formal side of modeling preferences and Allen's temporal constrains by means of the interval fibred semantics. In this chapter we address the question of how to exploit arrangements of last chapter for a construction of hybrid plan controller. A *Plan Controller* constitutes a machine (sometimes only an abstract one) suitable for supervising how different tasks are performed in a comparison with the initially developed plans or schedules. The plan controller[2] construction is often a multi-stage activity. It begins with a description of the robot environment and tasks in the appropriate formal language (for example: Linear temporal Logic (LTL)). This description should be later translated into the appropriate automata which encode the initial part of information in terms of automaton states and admitted transitions between them. Such controllers satisfy many different features – dependently on initial conditions and restrictions imposed on robot task performing.

5.2 Motivation of Current Analyses

Despite of the fact that Büchi automata (a basis of the plan controller construction) was already introduced in 1962 in [119, 120, 121] and exploited for translation of formulas of Motion Description Language in [122] and LTL-formulas to these automata in [123, 124, 125] – the construction of the hybrid plan controller automaton was made relatively recently in [126, 127]). Unfortunately, these original approaches suffer from the following difficulties:

D1 They discuss this issue rather in a form of extended outlines and without (often needed) technical details.

D2 The next, these constructions do not refer to metalogical restrictions of LTL and HS – such as the proved by L. Maximova in [128] and by A. Montanari in [129] with respect to encoding HS with $A\bar{A}B\bar{B}$-operators by formulas of ω-regular languages.

D3 Finally, the mutual relationships between relational semantics for LTL and Büchi automata is still unclear.

These difficulties motivate me to propose a more detailed outline of the hybrid plan controller construction taking into account recent metalogical results regarding LTL and HS. Finally, we intend to explain the relations between Büchi automata and fibred semantics for formulas of LTL extended by some operators of HS-logic.

5.2.1 Objectives and Novelty of this Chapter.

Due to the motivation factors and (slightly) against this state of art – this chapter is aimed at:

Obj1 proposing a preferential extension of the hybrid plan controller construction – based on product automata,

[1] The investigations of this chapter were published in [118] and presented during the conference FedCSIS in Gdańsk in September 2016.

[2] They are often called "hybrid controllers" as they join different automaton types and they refer to different features and 'entities' such as plans and the motion environments.

Obj2 construction of hybrid plan controller for a robot performing tasks in a polygonal environment,

Obj3 PROLOG-representation of chosen fragments of the plan controller, just constructed.

Novelty of current investigations with respect to the earlier ones may be listed as follows:

Nov1 A new general method of a plan controller construction for temporal planning with preferences is proposed. (Earlier approaches and constructions of automata only referred to temporal aspects of such a controller construction.)

Nov2 Description of robot motion environment and robot behavior is proposed in LTL extended by $\text{HS}^{D,L}$. (Earlier approaches were based on LTL or $\text{HS}^{D,L}$ – taking separately – as in [108].)

Nov3 Some new definitions – such as Preferential Büchi Automaton – is introduced.

Nov4 At the first time, Vardi's from [130] idea of a construction of automata for LTL-formulas is used in more practical contexts of temporal planning and preferences.

Moreover, the author of this thesis gives venture of his enthusiasm with respect to some utility of the proposed construction in different areas of utility of temporal logic systems: in engineering or – for example – in business processes and their management. Some applicability of temporal logic systems for engineering has been recently discussed in [131, 132], for a use of business management – in [133, 134]. Last, but not least, an expected future implementation of the automaton construction in the languages of a declarative paradigm, such as PROLOG or ASP, forms some additional motivating factor of this chapter analysis.

5.3 Preliminaries

Before moving to main chapter body, its terminological framework will be presented by introducing a new concepts of *preferential automata* and *preferential transition system*. The definitions of a (finite) transition system and a Büchi automaton are incorporated from [123, 124][3].

Definition 32 *[123, 124]. A* `Finite Transition System` *FTS is a n-tuple:*

$$FTS = (W, W_0, Act, Tran, Lab),^{4} \tag{5.1}$$

where:

- *W is the finite set of states (worlds),*
- *$W_0 \subseteq W$ is a distinguished set of initial states,*
- *Act denotes the set of possible actions,*
- *$Tran : W \times Act \mapsto W$ is a transition function, i.e such a total function that returns the next stage for a given state an an action,*

If FTS is considered in a context of a given language,say $\mathcal{L}$, one can also define a labeling function $Lab : 2^W \mapsto FOR$, where FOR is a set of well-formed $\mathcal{L}$-formulas.

[3]One needs to note that an alternative way to manage temporal aspects of automata was proposed in the concept of *timed automata* or *automata with the clocks*, as in [135, 136, 137]. Some matalogical features, such as model-checking, are discussed with respect to them in these papers.

[4]Vardi at all consider in [123, 124] finite transition systems with observations. Namely: FTS is defined there as n-tuple: $FTS = (W, W_0, Act, Tran, Lab, \Pi, Obs)$, where W, W_0, Act and $Tran$ are defined as below and Π denotes the set of possible observations, $Obs : W \mapsto \Pi$ is the observability function, which returns the observable part of the current state.

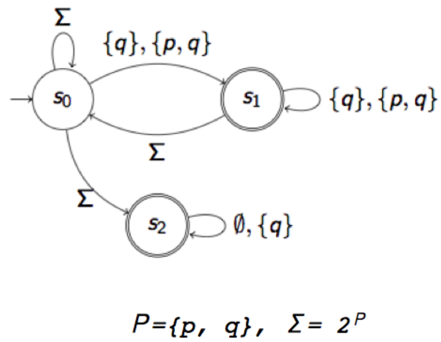


Figure 5.1: Fragment of Büchi automaton with states s_0, s_1 and s_2 over an alphabet Σ.

We define an *execution* on FTS as an infinite sequence of states $w_0, w_1, \ldots$, such that $w_0 \in W_0$ and $w_{k+1} = Tran(w_k, a)$ for some action $a \in Act$. The observable part of the execution will be called a *trace*.

Definition 33 *[123, 124]. A* ***Büchi automaton*** *is a tuple* $A = (\Sigma, S, S_0, \rightarrow, \rho, \mathcal{F})$, *where:*

- Σ *is the alphabet of the automaton,*
- S *is the set of states of the automaton,*
- $S_0 \subseteq S$ *is the set of initial states of the automaton,*
- $\rho : S \times \Sigma \mapsto 2^S$ *is the transition function of the automaton and*
- $\mathcal{F}$ *is the set of accepting words of automaton (over* Σ*).*

We expand this definition to a definition of preferential Büchi automaton by specification of the set of accepting words by introducing some degrees/parameters α's from an interval $[0, 1]$. The role of them is to measure a *degree of a preference* of the accepting words from F, each of them indexed by some α.

Definition 34 ***A Preferential Büchi automaton*** *is a tuple* $A = (\Sigma, S, S_0, \rightarrow, \rho, f, \Delta, \mathcal{F}^{\Delta})$, *where:*

- Σ *is the alphabet of automaton,*
- S *is the set of states of automaton,*
- $S_0 \subseteq S$ *is the set of initial states of automaton,*
- $\rho : S \times \Sigma \mapsto 2^S$ *is the transition function of automaton,*
- $f : \Sigma \mapsto [0, 1]$ *called* ***valuation*** *(function),*
- $\Delta = \{\alpha_1, \alpha_2, \ldots\}$ *is a set of values of a function* $f : \Sigma \mapsto [0, 1]$ *called* ***preferences****.*

$\mathcal{F}^{\Delta}$ *is the set of accepting words – each with associated* $\alpha \in \Delta$*.*

We assume that a set of α's degrees is finite – as they index accepting words from set F^{α}. Naturally, F^{Δ} should be finite as a a set of accepting words of a given finite automaton[5].

[5]This consideration may be enriched by further theoretic definitions such as a definition of a *run* of automaton and by criteria of word acceptance. Since they are slightly redundant in the perspective of current considerations, they be omitted in a main chapter text. Anyhow, they can be defined as follows: For such a defined automaton we can define a *run* of $\mathcal{A}$ (on an infinite word $a_0, a_1, a_2, \ldots$) as an infinite sequence of such states $s_0, s_1, \ldots \in S^{\omega}$ that $s_0 \in S_0$ and $s_{i+1} \in \rho(s_i, a)$. Thus, we could say that a run r is *accepting* iff a set $\{s | s$ occurs in r infinitely often $\} \cap F \neq \emptyset$. If F is finite, this general condition would mean that there exists at least one state s that occurs in a run r infinitely often. Compare:[123, 124]

Further considerations will exploit LTL-system, as described in "Introduction" (subsection 5.1). $\mathcal{L}(LTL)$ will be enriched now by modal operators $\langle L\rangle$- and $\langle D\rangle$ of $\mathcal{L}(HS)$. (HS in such a restricted language will be denoted by HS^{L,D}). The following example illustrates how formulas of $\mathcal{L}(HS^D)$ can support an expressive power of $\mathcal{LTL}$.

Example 33 *Some simple spatial-temporal requirements imposed on the robot's environment E can be expressed by formulas:*

- *Always if you take a block A, take B, as well:* $G(take(A) \rightarrow take(B))$,
- *For any intervals: if you take A, than also put B:* $[D](HOLDS(take(A)) \rightarrow HOLDS(put(B)))$.

5.4 Problem Formulation and a General Algorithm of the Controller Construction

Assume that E is a polygonal environment of robot motion operations. All possible admitted holes of E have to be enclosed by a single polygonal chain. The motion of robot may be rendered by the clauses:

$$x(t) \in E \subseteq R^2, u(t) \in U \subseteq R^2, u(t) \cap x(t) \neq \emptyset \tag{5.2}$$

where $x(t)$ is a trajectory of robot's motion (position of a robot in a time t) in E and $u(t)$ is a control time-dependent function (called also a control input). Non-emptiness of intersection $u(t) \cap x(t)$ ensures that a controller detects the robot's trajectory.

The construction objective and its stages. In such a framework, the construction objective is to give an outline of a construction of a hybrid controller that generates $u(t)$ for a trajectory $x(t)$ and environment E – specified by LTL-formulas and HS-formulas restricted to D-operator.

An exact chronology of the construction steps in a desired controller construction looks as follows.

step1 We begin with the environment E and its triangulation.

step2 Secondly, we consider some transition system FTS to describe a basic dynamism of E.

step3 The next, we specify E in terms of LTL (ϕ-formula) and of some subsystem of HS logic.

step4 In this step, we transform FTS to the appropriate Büchi automaton $\mathcal{A}_{FTS}$ for it. The similar automaton $\mathcal{A}_{LTL,HS}$ is constructed for representation of a specification of E (with a chosen point x_0) in terms of the considered temporal logic.

step5 Finally, having these automaton, we construct some product automaton $\mathcal{A}$ to 'reconcile' the activity of both automata.

This construction idea may be depicted as illustrated in Figure 5.2.

Alternatively, the same idea may be represented by the appropriate algorithm. Therefore, assume that some environment E of a robot and a formula $\phi \in \mathcal{L}(LTL \cup HS^{D,L})$ – describing this environment and the robot motion are given. Thus, the algorithm of the hybrid controller construction could be given as follows – as a specified version of algorithm from [126]:

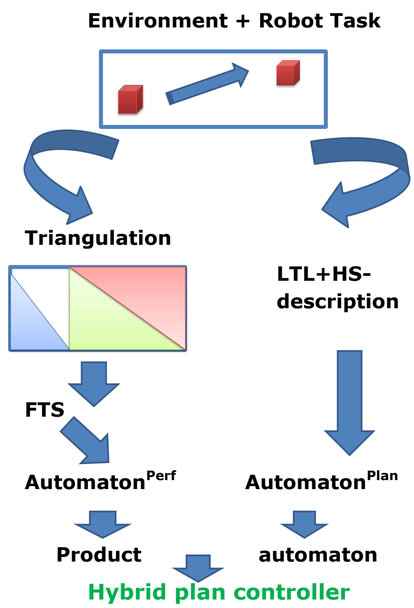


Figure 5.2: An outline of the procedure of the hybrid plan controller construction

Algorithm: The Hybrid Controller Construction

Procedure: CONTROLLER(E, ϕ)

1. $\triangle \leftarrow Triangulate(E)$
2. $FTS \leftarrow TriangulationToFTS(\triangle)$
3. $\mathcal{A}_{FTS} \leftarrow FTS\ to\ Buchi\ Automaton$
4. $\mathcal{A}_{LTL,HS^D} \leftarrow LTL \cup HS^D to\ Buchi\ Automaton$
5. $\mathcal{A} \leftarrow Product \mathcal{A}_{FTS}, \mathcal{A}_{LTL,HS^D}$
6. **return**: $Controller(\mathcal{A}, \triangle, \phi)$

End procedure

5.5 The Controller Construction for Temporal Planning

5.5.1 From Triangulation to the Finite Transition System (FTS)

In order to propose an exact construction of our path temporal planning controller we will represent a polygonal environment E as a finite set of partitions. One can use many methods of the initial polygonal environment's decomposition, presented in [126, 138].

Triangulation of environment. The main idea of such a triangulation consists in a mapping of each point $x \in E$ to a one of the disjoint equivalence classes determined by an equivalence relation $\sim$. The natural way is to define $\sim$ as follows: $\forall x, y \in E : x \sim y \iff x, y \in \triangle$, i.e each of such an equivalence class forms a triangle, what allows us to represent a quotient set $E| \sim$ as a sum of triangles. Assume now that $T : E \to Q$ for $Q = \{\triangle_1, \ldots \triangle_k\}$. Than each $T^{-1}(\triangle_i)$, for $i \in \{1, 2, \ldots, n\}$ contains some states $x \in E$ and a set $\{T^{-1}(\triangle_i) | \triangle_i \in Q$, for $i \in \{1, 2, \ldots, n\}\}$ of all such triangle anti-images is a desired partition of the initial motion environment E. In order to preserve a consideration generality, we do not impose any special requirements on E (concerning the temporal or spatial coverage etc.)

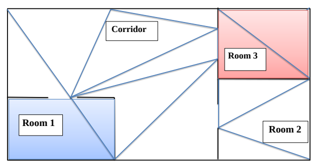


Figure 5.3: An example of a triangulation of some polygonal robot environment

For such a triangulated environment E – one can introduce a *Finite Transition System with Preferences*– FTSP, modifying a definition of of FST from [126, 127] as follows:

Definition 35 *Consider that a triangulated environment E is given, i.e. $E = \{\triangle_i : \triangle_i \neq \triangle_j \text{ for } i \neq j \in I\}$. Then* `the Finite Transition System FTS with Preferences` *is n-tuple*

$$FTS = \langle Q, Q_0, Act, T, tran_{FTS}, Pref \rangle, \tag{5.3}$$

where:

- *Q is the finite set of states (triangles) $\subset E$,*
- *$Q_0 \subseteq Q$ is the set of initial robot triangle,*
- *Act is the set of actions,*
- *$T : E \mapsto Q$,*
- *$tran_{FTS} : Q \times Act \mapsto Q$ is defined as a 'move' from $\triangle_i \to \triangle_j$ iff the cells $T^{-1}(\triangle_i)$ and $T^{-1}(\triangle_j)$ share a common edge,*

Pref: $Q \mapsto [0, 1]$ is a `preferential function` *which associates to $\triangle_1 \in Q$ some value $\alpha \in [0, 1]$, i.e. $Pref(\triangle_1) \in [0, 1]$.*

Remark 3 *Note that* `the preferential function Pref` *plays a similar role in FTS as a function $f : \Sigma \mapsto [0, 1]$ in the* `preferential automata`*.*

Example 34 *One can consider such a transition system with preferences as a system – depicted on a Fig. 5.3 with a function PREF, which satisfies the condition: $PREF(\triangle(room3)) = \frac{1}{2}$. (All triangles from a room 3 are preferable (in a sense of a robot task as places to visit) with a degree $\frac{1}{2}$).*

5.5.2 From FTS System to the First Büchi Automaton

It easy to observe that the finite transition system FTS naturally models a basic structure of the motion environment. It can play this role independently of a way of its presentation – in the standard form: $FTS = (W, W_0, Act, Tran, \Pi, Obs)$ or in the "triangle" depiction $FTS = \langle Q, Q_0, Act, T, tran_{FTS}, Pref \rangle$. Furthermore, the structure of such FTS's – as one can observe – show some similarity to a structure of automata, so they seem to be a natural base of a construction of the required automata. In fact, (finite) automata may be viewed as (finite) transition systems with an additionally given alphabet Σ and set o accepting words over Σ as the required components of the automaton structure. Due to some practice in this area – expressed in [126, 127] – we will consider the standard representation of FTS as more suitable for an expected automaton construction.

Authors of the approach from [126, 127] recommended an immediate use a structure of such an FTS for this construction. We only partially make use of this advise. Of course, we will base the automata construction on the FTS-structure with its states and transitions as states and transitions of the newly constructed automaton. Nevertheless, we decide for a more"descriptive" solution: we firstly describe a robot environment (FTS) in terms of LTL and we construct states of desired automaton from sets of the used formulas. For that reason, one need now a mechanism of such a description. Its presentation will be given below.

5.5.3 From LTL and $HS^{L,D}$ to the Second Büchi Automaton

The next step of our construction will consist in a constructing of the appropriate Büchi automaton for LTL- and HS-formulas expressing the robot motion environment E and the action sequencing. This situation is, however, not so comfortable as in the FTS-case for three purposes.

A At first, both LTL and HS form formal languages, so their translation for the Büchi automaton should be more sophisticated,

B Furthermore, we deal with two languages: LTL and HS of a different temporal and a modal-temporal nature,

C Last, but not least, HS logic (in generality) cannot be expressed by the Büchi automaton because of an enormous expressive power of this system in its full original version.

Fortunately – as it was already mentioned – it was proven in [108] that the subsystem of HS with D-operator can be represented by a finite state automaton. Unfortunately, it is not clear whether there exists any extension of this subsystem with the same property. Nevertheless, it is known that a subsystem $A\bar{A}B\bar{B}$ is too strong as ω-regular languages can be embedded in this system, but not in the inverse direction [129]. According to these remarks on the Büchi automaton construction in section 2, we begin with a definition of a closure of formulas of $LTL \cup HS^D$.

Definition 36 *[123, 124]. Let assume that $\phi \in \mathcal{L}(LTL)$. We define its* `closure` *$cl(\phi)$ as follows:*

- $g \in cl(\phi)$,
- $\phi \in cl(\phi) \iff \neg(\phi_1) \in cl(\phi)$,
- $\phi_1 \wedge \phi_2 \in cl(\phi)$ *then* $\phi_1, \phi_2 \in cl(\phi)$,
- $A(\phi_1, \ldots, \phi_k) \in cl(\phi)$, *then* $g_1, \ldots, g_n \in cl(\phi)$.
- $A(\phi_1, \ldots, \phi_k) \in cl(\phi) \rightarrow A_{sub}(\phi_1, \ldots, \phi_k) \in cl(\phi)$ *for* A_{sub} *denoting a sub-formula of* A.

Intuitively, the closure of ϕ consists of all subformulas of ϕ and their negations. In the similar way we will define a closure for $\phi \in \mathcal{L}(\mathrm{HS}^D)$ and for formulas of $\mathcal{L}(HS^D \cup LTL)$. For a distinction of these languages and their formulas we will denote them as: ϕ_{LTL} and ϕ^D_{HS}.

Definition 37 *Assume that an arbitrary formula $\phi \in \mathcal{L}(LTL \cup HS^D)$ is given. In accordance with the earlier statements an automaton $\mathcal{A}_{LTL,HS^D}$ for ϕ is defined as n-tuple:*

$$\mathcal{A}_{LTL,HS^D} = \left(2^{cl(\phi)}, \rho, S_0^{\phi_{LTL}}, S_0^{\phi^D_{HS}}, \mathcal{F}\right) \tag{5.4}$$

where

- $2^{cl(\phi)}$ *is the set of automaton states as the collection of all possible sums of subformulas of* $cl(\phi)$*,*
- ρ *is the transition and*
- $S_0^{\phi_{LTL}}, S_0^{\phi^D_{HS}}$ *are sets of initial states of automaton for* ϕ_{LTL} *and* $\phi^{D,L}_{HS}$*,*
- *finally,* $F \subseteq 2^{cl(\phi)}$ *is a set of accepting words of automaton*[6]*. (resp.).*

One can naturally expand this definition – based on [123, 124] to the definition of a *preferential automaton* as follows:

Definition 38 *Assume as earlier that a formula $\phi \in \mathcal{L}(LTL \cup HS^D)$ is given. Then* `the Preferential Automaton` *(for words of $\mathcal{L}(LTL \cup HS^{D,L})$) is defined as n-tuple:*

$$\mathcal{A}_{LTL,HS^D} = \left(2^{cl(\phi)}, \rho, S_0, f, \Delta, \mathcal{F}^\Delta\right) \tag{5.5}$$

- *where – as earlier –* $2^{cl(\phi)}$ *is the set of states of the automaton,*
- ρ *is the automaton transition,*
- $S_0 \subseteq 2^{cl(\phi)}$ *are sets of initial states of automaton,*
- $f : 2^{cl(\phi)} \mapsto [0,1]$ *is a function called: (a fuzzy) 'valuation*[7]*,*
- $\Delta = \{\alpha_1, \alpha_2 \ldots\}$ *is a set of values of a function* $f : 2^{cl(\phi)} \mapsto [0,1]$.
- $\mathcal{F}^\Delta \subseteq 2^{cl(\phi)}$ *is a set of accepting words of automaton.*

Remark 4 *It easy to see that a set $2^{cl(\phi)}$ plays a double role: of a set of the automaton states and a reservoir of accepting words. This situation is natural in the light of the fact that states of such an automaton are just built up from subformulas of ϕ.*

Remark 5 *It is not difficult to see that this general automaton depiction is not sensitive to a distinction between sets S_0 and $\mathcal{F}^\Delta$ – built from the same set $2^{cl(\phi)}$. Such a distinction is a matter of acceptance criteria, which should be however, omitted in such a general definition.*

These definitions is illustrated by the following example.

Example 35 *Let us consider $\phi = \neg\phi_1$ for some ϕ_1 as a LTL-formula and $\psi = \langle D\rangle\psi_1$ (for some ψ_1) as our HS^D-formula. We show how to construct a Büchi automaton in a case of these formulas. Due to definition of a closure of the formula we obtain:*

[6] Let us observe that defining of $\mathcal{F}$ as accepted words determines, somehow, a set of final states by this way of state definition – just by formulas, what seems to justify $\mathcal{F}$ as a subset of $2^{cl(\phi_{LTL}) \cup cl(\phi^D_{HS})}$.

[7] The function of fuzzy valuation is sometimes denoted by e is some metalogical contexts and called: 'evaluation' – see: [79].

- $cl(\phi) = \{\phi_1, \neg\phi_1\}$ *and* $cl(\psi) = \{\psi, \neg\psi, \langle D\rangle\psi, \neg\langle D\rangle\psi\}$,
- *thus* $2^{cl(\phi)} = \{\emptyset, \{\phi_1\}, \{\neg\phi_1\}, \{\phi_1, \neg\phi_1\}\}$ *and* $2^{cl(\psi)} = \{\emptyset, \{\psi\}, \{\neg\psi\}, \{\psi, \langle D\rangle\psi\}, \{\psi, \neg\langle D\rangle\psi\}, \{\neg\psi, \langle D\rangle\psi\}, \{\neg\psi, \neg\langle D\rangle\psi\}\}$,
- *Transitions* ρ_{LTL} *and* ρ_{HS^D} *are defined in such a way that sets from* N_ϕ *are initial in* ρ_{LTL} *and* N_ψ *are initial for* ρ_{HS^D}.

5.5.4 Product Automaton $\mathcal{A}_{FTS} \times \mathcal{A}_{LTL,HS^D}$

We have already defined two automata: the automaton $\mathcal{A}_{FTS}$, which describes the finite transition system and the automaton $\mathcal{A}_{LTL,HS^D}$ that represents the initial temporal logic-based specification of the motion environment. There is a need to reconcile both automata in order to construct our open-loop hybrid controller. For this purpose, it seem to be reasonable to restrict a *spectrum* of the possible transitions to these of them, which can ensure some form of observability.

For this purpose we introduce a product automaton[8] $\mathcal{A} = A_{FTS} \times \mathcal{A}_{LTL,HS^D}$ with a new transition $\rightarrow_\mathcal{A}$ between pairs: $(p_i, q_i) \rightarrow_\mathcal{A} (p_j, q_j)$. We assume that this transition holds between such pairs if and only if $p_i \rightarrow_{FTS} p_j$ and $(q_i; \pi(q_j)) \rightarrow_{LTL} w_j$. It leads to the following definition of *the preferential product automaton*.

Definition 39 *Let:* $\mathcal{A}_1 = \langle \Sigma_1, S, S_0, \rightarrow_{A_1}, f_1, \Delta_1, \mathcal{F}^{\Delta_1}\rangle$ *and* $\mathcal{A}_2 = \langle \Sigma_2, T, T_0, \rightarrow_{A_2}, f_2, \Delta_2, \mathcal{F}^{\Delta_2}\rangle$ *are preferential automata, than their* `Preferential Product Automaton` *is the automaton of the form:*

$$\left(\Sigma_1 \times \Sigma_2, S \times T, S_0 \times T_0, \rightarrow_{A_1 \times A_2}, F, \Delta_1 \times \Delta_2, \mathcal{F}^{\Delta_1} \times \mathcal{F}^{\Delta_2}\right), \tag{5.6}$$

where:

- $\rightarrow$ *is a product transition defined such that:* $(s_i, t_i) \rightarrow (s_{i+k}, t_{i+k})$ *holds for natural* $i \in I$ *and* $1 \leq k$ *if and only if* $s_i \rightarrow_{A_1} s_{i+k}$ *and* $t_i \rightarrow_{A_2} t_{i+k}$,
- $F = \begin{cases} f_1, & \text{for automaton } \mathcal{A}_1, \\ f_2, & \text{for automaton A}_2. \end{cases}$

(For simplicity, we will shortly write $\rightarrow$ instead of $\rightarrow_{A_1 \times A_2}$, when it will not lead to any confusion.) Examples of product automata will be given in the next part focused on an implementation.

Complementary Conditions for a Product Automaton

In fact, this construction requires a small complementation. Namely, some singletons of $\mathcal{A}$ can have no outgoing transitions. In order to ensure a normal work of the automaton, we add the so-called 'stutter extension' rule [139], which adds a self-transition on the blocking states. More formally, for all states $s \in$ domain of $\mathcal{A}$, a new transition is defined: $\rightarrow_{\mathcal{A}^*} = \rightarrow_\mathcal{A} \cup (s \rightarrow_\mathcal{A} s)$, where $\rightarrow_\mathcal{A}$ is the transition of the automaton $\mathcal{A}$ [9], earlier defined. In such a framework it holds the following:

Theorem 11 *(adopted from [125]) An execution of FTS that satisfies the specification in terms of LTL and HS-formulas exists iff the language of A is non empty.*

We omit the proof details. For a case of LTL-specification it can be found in [123, 124]. For a case of HS^D it follows from the existence of the automaton accepting formulas of this logic from [108].

[8] It seems that creating the product of automata is not an only method of their combining in order to reconcile a required piece of information encoded by these automata separately. In this moment I would like to thank to Prof. P. Stuzak for a stimulating discussion during an AAIA-session of FedCSIS'16-conference in Gdańsk.

[9] In fact, we consider a projection of $\rightarrow_\mathcal{A}$ for the set of s-states, because $\rightarrow_\mathcal{A}$ works for pairs of states.

5.6 Part II: Implementation

We have just given in last part of this chapter a theoretic outline of the hybrid controller construction – based on some product automaton. In addition, a description of the robot motion environment and the robot plan have been rendered in LTL extended by some fragment of HS-logic. In this part we intend to illustrate these ideas by proposing a concrete construction of such a controller. According to the earlier arrangements – this construction will be multi-stages and it will contain the following steps:

St1(appl) a presentation of the robot motion environment ,

St2(appl) a formal description of the environment and the plan of the robot in terms of $LTL \bigcup HS^{D,L}$,

St3(appl) a formal description of the environment and a real plan performing by the robot in terms of $LTL \bigcup HS^{D,L}$

St4(appl) a construction of the appropriate Büchi automata for both the cases (the first one – for a desired plan, the second one – for a real plan performing by the robot) ,

St5(appl) a construction of a product automaton built up from automata from a point (4).

St6(appl) a PROLOG-description of the product automaton in order to detect eventual discrepancies between a plan and its performing by the robot .

However, this construction slightly deviates from the general algorithm in the theoretic depiction, as described earlier. The difference consists in a fact that the robot environment was almost immediately used as a natural basis for the appropriate automaton construction. Only the plan specification was rendered in LTL-based automata. In this case, both the plan specification and the robot motion environment are described in automata built up from (closures of) LTL-formulas. This unification in our practical approach makes the comparison in the product automaton easier, more transparent and natural. This description has a form of an extended outline as it avoids some of the construction details. It is planned to complement these investigations in a more detailed way in a future research.

5.6.1 The Robot Environment and its $LTL \bigcup HS^{D,L}$-specification

Let us consider a robot, say R, in some polygonal environment with 4 rooms as depicted on a picture below. Assume that R performs a task to dislocate a black block A from a room 1 to the room 4 and put in on a block B there and the planned (preferred) move trajectory leads from the room 1 by a neighborhood of the room 3 to the room 4 (the blue line on a picture). Let also assume that our robot exchanged this trajectory for another one (marked by a red line).
Therefore, the robot motion environment and plan specification in $LTL \cup HS^{L,D}$ may be rendered as follows:
• **Plan + Preferences**:

1. Take a block A.
2. Move from R_1 to R_3 (more preferable) or Move from R_1 to R_4 (less preferable).
3. If you are in R_3, move from R_3 to R_4.
4. Go to the room R_4).
5. Put a block A on the block B.

We can also extract the following 'behavioral' rule for the robot as a condition of an effective plan performing.
• **Condition for the plan performing/behavioral rules**:

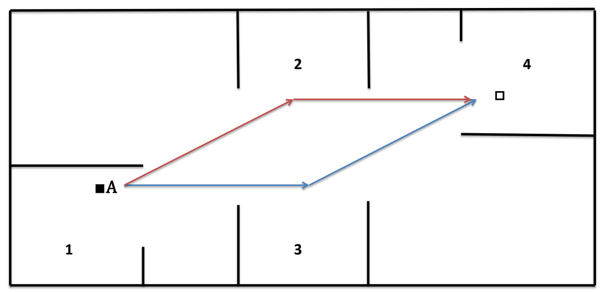


Figure 5.4: The polygonal environment of the robot motion with 4 rooms. The blue broken line illustrates the planned trajectory of the robot move from a room no. 1 to the room no. 4. The red one illustrates the deviated trajectory of the robot move.

1. Always in a future, if you take a block A, go to the room R_3.

Finally, we should give a short description of the polygonal robot motion environment.

- **The robot environment**:

1. Environments consists of 4 rooms ,
2. In a room R_1, block A is initially located ,
3. In a room R_4, block B is initially located ,
4. In rooms R_2, R_3 no blocks are located ,
5. In room R_4, block A is finally located ,
6. The robot motion area is always located on the left of a room R_1 ,
7. The robot's motion area is always located on the right of a room R_4 .

All these conditions may be regarded now in terms of $LTL \bigcup HS^{D,L}$ in the corresponding way as follows: • **Plan + Preferences**:

1. $Take(A)$.
2. $Move(R_1^A, R_3) \vee Move(R_1^A, R_4)$.
3. $HOLDS(R_3^R) \rightarrow Move(R_3, R_4)$.
4. $Put(A)$.
5. $HOLDS(R_4^A)$.

- **Condition for the plan performing/behavioral rules**:

1. $G(take(A) \rightarrow \langle L \rangle go(R_3)$.

- **The robot environment**:

1. $R_1 \wedge R_2 \wedge R_3 \wedge R_4$,
2. $R_1^A \wedge R_4^B \ \ (HOLDS_{Init}(R_1^A))$,
3. $R_4^A \ \ (HOLDS_{Fin}(R_1^A))$,
4. $[D](R_1 \rightarrow Left)$
5. $[D](R_4 \rightarrow Right)$.

Let us return now the initial assumption that the robot deviated from the planned path and has chosen a red line from $R_1 to R_4$, so via R_2. We can trace this deviation for the following juxtaposition of two formal descriptions in terms of $LTL \bigcup HS^{D,L}$ for both situations.

plan of the robot	**the real plan performing**
$Take(A)$	$Take(A)$
$Move(R_1^A, R_3) \vee Move(R_1^A, R_4)$	$Move(R_1^A, R_2)$
$HOLDS(R_3^R) \rightarrow Move(R_3, R_4)$	$Move(R_2^A, R_4)$
$Put(A)$	$Put(A)$
$HOLDS(R_4^A)$	$HOLDS(R_4^A)$
behavioral rule	**behavioral rule**
$G(take(A) \rightarrow \langle L \rangle go(R_3)$	?

5.6.2 From $LTL \cup HS^L$ to the Second B'uchi Automaton

The next stage of the plan controller construction consists in the appropriate translation of $LTL \cup HS^L$-formulas to states of B'uchi automaton, which usually is not given *a priori*, but it must be constructed in the appropriate way. The automata will be constructed due to ideas from [123, 124]. Assuming that a given formula $FOR \in \mathcal{L}(LTL \cup HS)$[10], the B'u chi automaton is constructed for FOR according to the following general rules:

Rule1 : Each state of the automaton is determined by some subformulas of FOR and their negations.

Rule2 : The initial state of automata is always the state with the empty word.

Rule3 : If a state s_k $(1 \leq k)$ of the automaton is identical to $\{\phi\}$ (for some $\phi \in FOR$), then the states immediately admissible from s_k by the transition function is identical to $\{N(\phi)\}$, such that $N(\phi) \in FOR$ and ϕ is a subformula of $N(\phi)$.

Rule4 : We avoid these states of automaton with these formulas, which are completely redundant from the point of view of the fixed goal. (For example, if the construction goal is a state based on a formula $A(\phi)$, then we avoid the state built from $\neg A(\phi)$ as redundant.)

Automata for LTL-words. In order to illustrate this procedure let us consider a unique atomic LTL-formula[11] $TAKE(A)$. Due to definition 40 – a closure of a formula ϕ contains all of its sub-formulas and its negations. According to Rule 1, the automaton states are determined by the following collection sets of formulas:

$$\emptyset, \{\neg A\}, \{A\}, \{TAKE(\neg A)\}, \{A, TAKE(\neg A)\}, \{TAKE(A)\}, \{A, TAKE(A)\}.$$

[10] We usually think about this language as based on a propositional descriptive or attributive language.

[11] In fact, there is an atomic formula of a descriptive or attributive version of LTL.

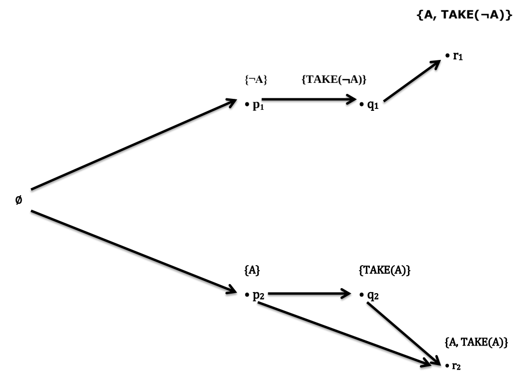


Figure 5.5: Fragment of the Büchi automaton with states for a closure of the LTL-formula Take(A).

According to Rule 2 – the initial state of the automaton is the state with the empty word. The automaton is further constructed due to Rule 3. In particular, the next states of the automaton are the singletons: $\{A\}$ and $\{\neg A\}$. In the next stage, we take $TAKE(A)$ and $TAKE(\neg A)$ as new formulas determining next states $\{TAKE(A)\}$ and $\{TAKE(A)\}$ (*resp.*) of the automaton. The whole construction is depicted in Fig. 5.5. Büchi automaton for $LTL \cup HS^L$-formula $Take(A) \rightarrow \langle L \rangle go(R_3)$ was presented in Fig.5.6.

The same principles enables of construction automata for $Move(R_1^A, R_3)$ (see: Fig. 5.7.) and for $Take(A)$ and $Move(R_1^A, R_3)$ (taken together) – as demonstrated in Fig. 5.8. Finally, automata for the 'behavioral rule' of the robot $Take(A) \rightarrow \langle L \rangle go(R_3)$ and for $Move(R_1^A, R_3)$ were presented in Fig. 5.9 and 5.10 (resp.).

It seems to be noteworthy to observe the transitions in Fig. 5.8 for $Move(R_1^A, R_3)$. Namely, the state $\{R_1^A\}$ allows us to move to $\{Move(R_1^A, R_3\}$ because R_1^A is a subformula of $Move(R_1^A, R_3$. However, a direct transition from $\{R_1^A\}$ to $\{R_1^A, Move(R_1^A, R_3\}$ is not admissible as it violates Rule 3.

It is not difficult to observe that a 'global size' of such automata for a complete plan of the robot and its motion environment is very large and its complete presentation would be difficult and non-suggestive. It is enough to observe that the full automaton is a composition of the appropriate fragments enriched by some 'move-lines' (the blue line in Fig. 5.9) in order to connect the appropriate fragments of this automaton.

As earlier mentioned, the construction of automata is aimed at encoding a piece of information both about:

A a required or planned performing of the robot/agent tasks and

B a real task performing by the agent/robot.

The discrepancies between both situations may be detected by a comparing the appropriate (fragments of) automata, Compare: automaton in Fig.5.7 and in Fig. 5.10. The first one describes the required plan execution. The second one – a real plan execution. The differences are marked in a blue color in the second of these automata (see: Fig. 5.10) Nevertheless, such a comparative analysis may be simplified. In fact, it is enough to encode both type of information in a single *product automaton*.

5.6.3 The Product Automaton

In fact, each product automaton preserves some portion of information encoded by two automata: by the plan-automata, say $\mathcal{A}^{plan}$ and by its 'rival' for the real task performing, say $\mathcal{A}^{perf}$. As each product structure,

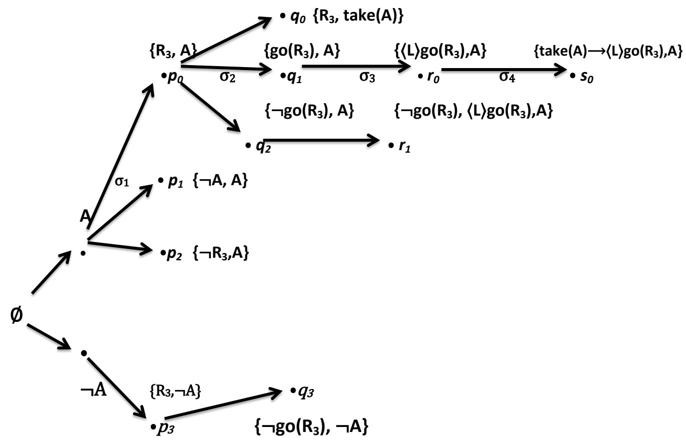


Figure 5.6: Fragment of the Büchi automaton with states for a closure of the LTL-formula $Take(A) \rightarrow \langle L \rangle go(R_3)$.

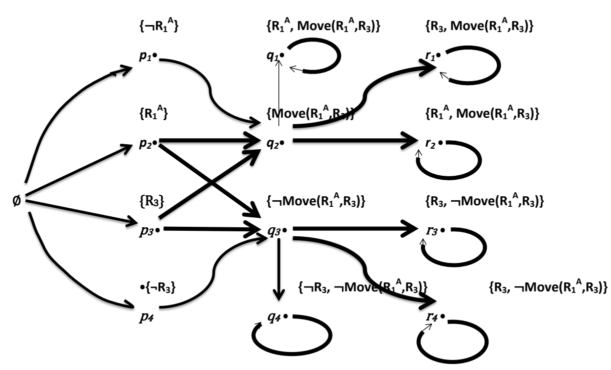


Figure 5.7: Fragment of the Büchi automaton with states for a closure of the LTL-formula a$Move(R_1^A, R_3)$.

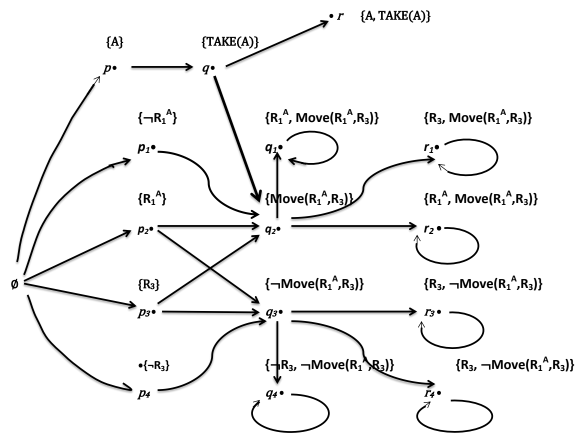


Figure 5.8: Fragment of the Büchi automaton with states for a closure of the LTL-formulas $Take(A)$ and $Move(R_1^A, R_3)$.

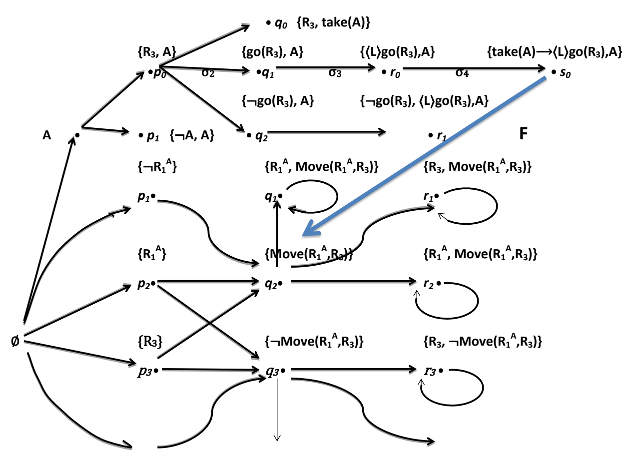


Figure 5.9: Fragment of the Büchi automaton for $Move(R_1^A, R_3)$ and for a fragment of a formula $take(A) \rightarrow \langle L\rangle go(R_3)$.

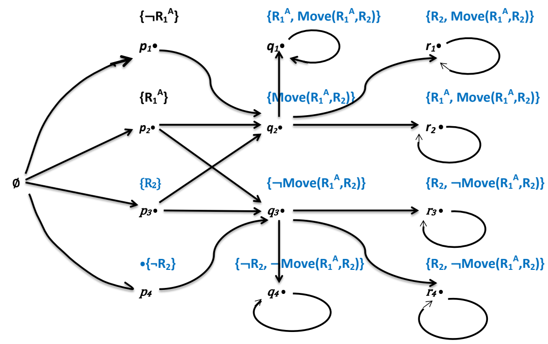


Figure 5.10: Fragment of the Büchi automaton for the real task performing for the closure of LTL-formula $Move(R_1^A, R_2)$.

the product automaton $\mathcal{A}^{plan} \times \mathcal{A}^{perf}$ is built up from pairs of states of the form: (s_1^{plan}, s_2^{perf}), where each $s_1^{plan} \in \mathcal{A}^{plan}$ and each $s_2^{perf} \in \mathcal{A}^{perf}$. [12] The similar pairs can be constructed with respect to transitions taken from $\mathcal{A}^{plan}$ and their equivalents from $\mathcal{A}^{perf}$ (if there are)[13].

Due to Definition 39 – *the preferential product automaton* for automata: $\mathcal{A}^{plan}$ and $\mathcal{A}^{perf}$ has the following algebraic form:

$$\mathcal{A}^{plan} \times \mathcal{A}^{perf} = \Big\langle \Sigma_1 \times \Sigma_2, S^{plan} \times S^{perf}, S_0^{plan} \times S_0^{perf}, \rightarrow, \Delta_{plan} \times \Delta_{perf}, F, \mathcal{F}^{\Delta_{plan}} \times \mathcal{F}^{\Delta_{perf}} \Big\rangle . \quad (5.7)$$

Example. Assuming that the choice of $Move(R_1, R_3)$ is preferable with a degree, say $\frac{2}{3}$, and the choice of $Move(R_1, R_4)$ – with a degree $\frac{1}{3}$ in the*robot plan*, the fragment of product automata $\mathcal{A} \times \mathcal{A}^{pref}$ for formulas $Move(R_1, R_3) \vee Move(R_1, R_4)$ and $Move(R_1, R_2)$ – due to definition – will have the following algebraic representation:

Alphabet: $\Sigma_1 \times \Sigma_2 = \mathcal{L}(LTL \cup HS^L) \times \mathcal{L}(LTL \cup HS^L)$,

States: $S^{plan} \times S^{perf} = 2^{cl\left(Move(R_1,R_3) \vee Move(R_1,R_4)\right)} \times 2^{cl(Move(R_1,R_2))}$,

States$_{Dist}$: $S_0^{plan} \times S_0^{perf} = \{\emptyset, \{R_1\}, \{\neg R_1\} \ldots\}^2$,

AccWord: $\mathcal{F}^{\Delta_{plan}} \times \mathcal{F}^{\Delta_{perf}} = \{R_1, R_3, R_4, Move(R_1, R_3)^{\frac{2}{3}}, Move(R_1, R_4)^{\frac{1}{3}}\} \times \{R_1, R_2, Move(R_1, R_2)^1\}$,

PrefValues: $\Delta_{plan} = \{\frac{2}{3}, \frac{1}{3}\}$, $\Delta_{perf} = \{1\}$ [14].

The construction of each product automaton consists in a combing states of the first automaton with states of the second one 'each to each'. For example, if automata for LTL-words: $TAKE(A)$ and $MOVE(R_1, R_3)$ are

[12] It is not completely correct, because we should pedantically state that s_1^{plan} belongs to some set S^{plan} of states of $\mathcal{A}^{plan}$, but we will omit this distinction for a simplicity and some suggestiveness of analysis.

[13] Of course, both automata $\mathcal{A}^{plan}$ and $\mathcal{A}^{perf}$ are defined over the same alphabet $\Sigma = \mathcal{L}(LTL \cup HS^{L,D})$.

[14] These last values may be arbitrarily taken as they essentially referred to the really performed action $Move(R_1, R_2)$. Anyhow, it seems to be reasonable to associate it a value 1.

producted, the final product automaton will contain such states as: $\{A, R_1\}$, $\{A, R_3\}$, $\{A, MOVE(R_1, R_3)\}$, etc.

It easy to observe that such a non-restricted combination is much redundant and may be troublesome. In fact, we need to compare the corresponding fragments of $\mathcal{A}^{plan}$ and $\mathcal{A}^{perf}$ only in order to detect possible discrepancies. This restricted producting of automata should also find its reflection in terms of programming languages.

5.6.4 PROLOG-description of the Product Automaton

The last step of the plan controller construction consists in encoding the product automaton in a declarative language such as PROLOG. An idea of encoding is simple: a 'status' and localization of automaton states will be described by such predicates as: final(x), initial(x), but the transitions between them may be rendered by 3-argument predicates arc(x,y,z). Since all states of automata are determined by the appropriate LTL-formulas, this fact may find its reflection in PROLOG-names characterizing these states. Thus, we can admit names of the type: `stateA, stateR2, stateNotR3, stateNotMove1R2`, etc.

In this framework some fragment of $\mathcal{A}^{plan}$-automaton for two formulas: $Take(A) \cup Move(R_1^A, R_3)$ may be rendered in PROLOG as follows:

```
arc(0, stateNotR1).  arc(0, stateR1).arc(0, stateR2).  arc(0, stateNotR2).
arc(stateNotR1, stateMoveR1R2).  arc(stateR1,stateMoveR1R2).
arc(state R1, stateNotMoveR1R2).  arc(stateR2, stateMoveR1R2).
arc(stateNotR2P, stateNotMoveR1R2) etc.
```

And corresponding part of $\mathcal{A}^{Pref}$-automaton:

```
arc(0, stateNotR1).  arc(0, stateR1).  arc(0, stateR3).  arc(0,stateNotR3).
arc(stateNotR1, stateMoveR1R3).  arc(stateR1, stateMoveR1R3).
arc(state R1, NotMoveR1R3).  arc(stateR2,stateMoveR1R3).
arc(stateNotR3P,stateNotMoveR1R3) etc.
```

The detected differences on a level of PROLOG-description are marked by red color.

It remains to enrich these PROLOG– descriptions by some preferential component – as it it has been made on a level of automata. Due to our convention – preferences are denoted by rational numbers from a fuzzy set $[0, 1]$ and they are associated to arcs between automaton states. For simplicity of the PROLOG-representation, we can assume that they can be associated to final states of such paths/arcs.

Assume, however, that we do not know how they are associated to concrete formulas or states, but we only known that each of formula can take one of values from the set $\{0, \frac{1}{2}, \frac{2}{3}, 1\}$. If we define the automaton branches for a $Take(A)$-formula by PROLOG lists, say X and Y, we also need to add a piece of information about possible fuzzy values that can be considered:

X`ins 0,`$\frac{1}{2}, \frac{2}{3}, 1, and$ `Y ins 0,`$\frac{1}{2}, \frac{2}{3}, 1$.

These coding examples do not exhaust the list of possible ways of PROLOG-encoding, but they are used for illustration and can be extended and specified in many ways.

5.7 Conclusions

We have just provided a multi-stages construction of the hybrid plan controller in terms of LTL extended by HS^D. Using such a description, we were led to encode features of robot motion environment and its behavior in terms of preferential Büchi automata. It has emerged that the initial discrepancy between a plan and its real performing by a robot can be encoded at each stage of the controller construction without losing of any portion of information. In fact, the same discrepancy at the stage of the LTL-description can be transformed to the stage of the automaton construction and – finally – could be visible at the stage of its PROLOG-representation, too. One could venture a thesis that attempts with other languages of a declarative paradigm give the similar results.

Naturally, the preferential extension of automata and the whole construction of the plan controller – that was proposed – forms a kind of an 'external' extension. In fact, we have not introduced any explicit preferential language to LTL extended by HS-language with $\langle D\rangle$ and $\langle L\rangle$. It seems that this task could be feasible in some preferential extension of HS or LTL.

Furthermore, we interrupted these investigations in a point, where interests of specialist in AI and computer science meet with a competence of experts in automatics and robotic. It was implied by the fact that we were not interested in a plan controller as a physically constructible machine, but (more) as a theoretic construct. Obviously, it does not mean that this theoretic abstract and ideal plan controller may not be implementable and constructed in reality. Its physical materialization is, however, conditioned and dependent on many additional factors that should not be put aside. One of these factors could be a problem of an 'ergonomy' of this construction, for example, with respect to a cost of transition between appropriate fragments of encoding automata. The word 'cost' may be also interpreted in a purely theoretic sense as an 'ergonomy' of needed moves and steps of the plan controller construction [15].

[15] The issue of a role of cost functions and costs of transition was taken by Prof. P. Sluzak during AAIA-session of the FedCSIS'16-conference. In this moment I would like to thank to him for stimulating remarks and to Prof. Halina Kwasnicka who underlined an importance of remarks of P. Pluzak for materialization of my construction ideas.

Chapter 6

Towards a Synthesis

The foundation on which the whole corpus of investigations of earlier four chapters has been constructed were the following two approaches to modeling fuzzy temporal constraints and preferences:

1. the interval-based approach and

2. the logical approach in terms of Preferential Halpern-Shoham logic.

It arises a natural question whether a more synthetic depiction might be elaborated. We venture to positively answer this question. The potential solutions may contain different proportion of these approaches and may be divided into two classes:

1(L = A) these two approaches play the same role in the new synergy approach,

2(A ¿ L) the convolution-based approach dominates and it is supported by the logical approach,

3(L ¿ A) the logical approach dominates and it is supported by the convolution-based one.

The holistic approach – which we intend to propose now – will belong to the third-type. It will refer to the problem of the plan controller construction. In fact, it seems that this controller construction in a logic-based way requires some complementation. In particular, it is not clear how to primary detect different deviations in the robot task performing.

We only decided to announce that it is a role of a a control function *u(t)* which traces a trajectory function *x(t)*. It seems that these functions may be more specified and normalized by integrals. In this way, the logical approach might be complemented by the integral-based approach.

6.1 Integral-based Approach as a Support of the Logical Approach

To approximate how the integral-approach supports the construction of the plan controller and some naturality of this analytic-support let us observe the following facts.

1. The robot environment E may be considered as a *metric vector space*. In fact, each potential move of the robot in E forms a vector in tis space. In addition, this polygonal E is assumed to be a subspace of metric space ($\mathbb{R}$, d), where d is Euclidean.

2. Even more – E may be seen as a unique *Hilbert space*[1] if we agree to interpret E as a field of all possible trajectories and control functions. In fact – without losing of generality – one can assume that each

[1] An exact definition of Hilbert space is given in Appendix.

such a robot trajectory $x(t)$ and each control function $u(t)$ is a Lebesgue integrable function defined over an interval $[a, b]$. All such Lebesgue-integrable functions form a Hilbert space $\Big(\mathbf{L}^p[a, b],\ \| \bullet \|\Big)^2$ with norms $\| \bullet \|$ defined as follows:

$$\|x\| = |x(t)| + \Big(\int_{[a,b]} |x(t)|dt\Big)^{\frac{1}{p}}, 0 < p \leq 1, \tag{6.1}$$

and

$$\|u\| = |u(t)| + \Big(\int_{[a,b]} |u(t)|dt\Big)^{\frac{1}{p}}, 0 < p \leq 1. \tag{6.2}$$

In this framework, one can identify the robot environment E with a Hilbert space $\mathbf{L}^p[a, b], \| \bullet \|$ of all such Lebesgue integrable trajectories $x(t)_i$ and control functions $u(t)_i$ that each trajectory $x(t)_i$ has a non-empty intersection with its corresponding control function $u(t)_j$ for $i, j \in I$. Briefly and formally:

$$E = \Big\{x(t)_i, u(t)_j \in \Big(\mathbf{L}^p[a, b], \| \bullet \|\Big) : x(t)_i \cap u(t)_j, \text{ for } i = j, i, j \in I,\ 0 < p \leq 1\Big\}. \tag{6.3}$$

One can generalize this reasoning for a multi-dimensional case of $\mathbb{R}^n$. Assume now that $u(t) \in \mathbb{R}^n$, so $u(t) = u(t_1, \ldots, t_k)$. Since a robot move equations are usually involved in partial derivatives it is comfortable to postulate their differentiability up to α-degree. In addition, one may require the following condition:

- both partial derivatives $D^\alpha u(t)$ and $D^\alpha x(t)$ should belong to the same space.

Formally:

$$D^\alpha u(t) := \frac{\partial^\alpha u(t)}{\partial t_1, \ldots \partial t_n} \in \Big(L^p[a, b], \| \bullet \|\Big), \tag{6.4}$$

and

$$D^\alpha x(t) := \frac{\partial^\alpha x(t)}{\partial t_1 \ldots \partial t_n} \in \Big(L^p[a, b], \| \bullet \|\Big). \tag{6.5}$$

In this case, E might be identified with the so-called *Sobolev's space*. Details of further specifications of

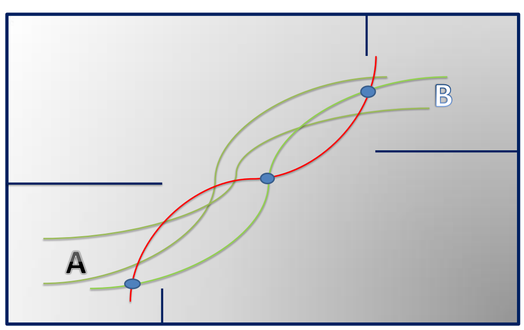


Figure 6.1: Robot polygonal environment (with rooms A and B) as a Hilbert space of Lebesgue integrable trajectories (green lines) and control functions (red line).

[2] Pedantically, one should defined both x and u as functions dependent on space locations x and time-points t. For simplicity, we consider them as only dependent on t. In this way, a space distance between two points of E will be measured by a temporal distance between them. Such a measure is sufficient from our point of view.

Sobolev's space – adopted for E-representation – need not concern us here. Nevertheless, it seems that this interval-based complementation of the purely logical E-specification has at least two advantages:

1. it delivers new*geometric features* of this environment and
2. it delivers new criteria of robot *controllability.*

We describe these advantages in detail.

New geometric features of robot environment. It is not difficult to note that some typical properties of Hilbert space $\mathbf{L}^2[a,b]$ may be interpreted as robot trajectories and the control functions in its motion environment. We put forward them in terms of these functions.

Assume thus that $x(t)$ and $y(t)$ are trajectories defined over $[a,b]$ and Lebesgue integrable over $[a,b]$ in E. Then it holds the following **Schwarz's inequality for trajectories:**

$$\Big(\int_a^b x(t)y(t)d\mu\Big)^2 \leq \int_a^b x(t)^2 \int_a^b y(t)^2 d\mu. \tag{6.6}$$

The similar inequality holds for Lebesgue integrable control functions, say $u(t), v(t)$. It will be called **Schwarz's inequality for control functions:**

$$\Big(\int_a^b u(t)v(t)d\mu\Big)^2 \leq \int_a^b u(t)^2 \int_a^b v(t)^2 d\mu. \tag{6.7}$$

If trajectories $x(t), y(t)$ as above – are also assumed to be non-negative functions and p, q are such that $\frac{1}{p} + \frac{1}{q} = 1$, then it holds the **Hölder inequality for trajectories:**

$$\int_a^b x(t)y(t)d\mu \leq \Big(\int_a^b x(t)^p\Big)^{\frac{1}{p}} \Big(\int_a^b y(t)^q d\mu\Big)^{\frac{1}{q}}. \tag{6.8}$$

The similar inequality holds for the control functions.

Finally, if $x_1(t)\dots x_k(t)$ are Lebesgue integrable non-negative trajectories in E, it holds the following **Minkowski's inequality for trajectories** for $k > 1$:

$$\Big(\int_a^b (x_1(t) + \dots + x_k(t))d\mu\Big)^{\frac{1}{k}} \leq \Big(\int_a^b x_1(t)d\mu\Big)^{\frac{1}{k}} + \dots + \Big(\int_a^b x_k(t)d\mu\Big)^{\frac{1}{k}} d\mu. \tag{6.9}$$

and for $0 < k < 1$:

$$\Big(\int_a^b (x_1(t) + \dots + x_k(t))d\mu\Big)^{\frac{1}{k}} \geq \Big(\int_a^b x_1(t)d\mu\Big)^{\frac{1}{k}} + \dots + \Big(\int_a^b x_k(t)d\mu\Big)^{\frac{1}{k}} d\mu. \tag{6.10}$$

These inequalities may be (more or less) interpreted as follows: a common area determined by a sum of robot trajectories $x_1(t)\dots x_k(t)$ in E is no greater(no smaller) than a sum of each of areas determined by their corresponding trajectories x_i separately. The similar feature holds for control functions satisfying the same properties.

Controllability in new terms. Considering the robot environment E in a more abstract way – as a Sobolev space of trajectories and control inputs – has an additional advantage. Namely, it allows us to adopt the following concepts of controllability and strong controllability – recently elaborated in [140].

1. The system (1) is said to be *locally controllable* at z if for each $\epsilon > 0$ there exists $\rho > 0$ such that for all $v \in B(0,\rho)$ there exists a trajectory x for F, with $\|x - z\| \leq \epsilon$, satisfying $(x(0), x(1) + v) \in S$.

2. It is said to be *strongly locally controllable* [1] at z if there exist $a > 0$ and $\rho > 0$ such that for all u and v in $B(0, \rho)$ there exists a trajectory x for F, with $\|x - z\| \leq a(|u| + |v|)$, satisfying $(x(0) + u, x(1) + v) \in S$.

Here $B(0, \rho)$ denotes the closed ball in H centered at 0 and of radius ρ.

A new algorithm of the controller construction.

It remains to answer to the following intriguing question:'How can we know about the robot plan performing and about the current robot situation in order to encode it in LTL and the appropriate automata'? The answer to this question seems to be clear. A carrier of information about the current robot situations and a real state of its plan performing are just control functions for the robot trajectories – as differntiable functions of the ideal Sobolev's or Hoelder's space.

Each LTL- or automata-based encoding the current situation must be proceeded by an earlier extraction of knowledge about the current robot situation. In a consequence, the general algorithm of the hybrid controller construction from paragraph 5.4 (pp. 127-128) may be enlarged to the following one:

Algorithm: The Hybrid Controller Construction

Procedure: CONTROLLER(E, ϕ)

1. $\triangle \leftarrow Triangulate(E)$
2. $FTS \leftarrow TriangulationToFTS(\triangle)$
3. $\mathcal{A}_{FTS} \leftarrow FTS\ to\ Buchi\ Automaton$
4. $\mathcal{L}TL \cup HS$-representation $\leftarrow$ Detecting Robot Trajectories by Controll Functions
5. $\mathcal{A}_{LTL,HS^D} \leftarrow LTL \cup HS^D to\ Buchi\ Automaton$
6. $\mathcal{A} \leftarrow Product \mathcal{A}_{FTS}, \mathcal{A}_{LTL,HS^D}$
7. **return**: $Controller(\mathcal{A}, \triangle, \phi)$

End procedure

The new 'key' point of this newly extended algorithm is the point 4. It encodes the procedural move from detecting the robot situation (trajectories) to its representing in terms of $LTL \cup HS$. The rest of the algorithm points are as in its original depiction on pages 127-28.

Obviously, this synergy proposal requires deeper analysis and it only forms a kind of an extended outline in the current version. It may be a promising subject for further research.

Chapter 7

Conclusions

7.1 Concluding Remarks

It seems that all expected objectives of the thesis have been satisfied in the framework of depth-analysis of the conceptual tissue of temporal planning with fuzzy constraints and preferences.

In fact, difficulties of earlier approaches for representation of temporal planning with fuzzy temporal constraints and preferences were detected in order to introduce a new (relatively exhaustive) taxonomy of temporal planning problems. It has emerged that such a taxonomy may contain only two classes of problems. The first class may be represented by Multi-Agent Schedule-Planning Problem; the second one – contains problems of Traveling Salesman Problem-type.

In the framework of (strict) contributions of this PhD-work two new approaches for representation and semantic modeling of fuzzy temporal constrains and preferences were proposed. They (at least partially) overcome difficulties of earlier approaches. In fact, a convolution-based approach has a hybrid nature – as it covers both quantitative and qualitative temporal constrains. Similarly, the second logical approach – constructed in terms of Preferential Halpern-Shoham logic and fibred semantics – overcomes some discrepancy between fuzzy temporal constrains and preferences – often separately considered. In fact, fuzziness is just introduced by preferences in this approach. In the framework of the more practical part of contributions – computational and programming-wise aspects of the convolution-based approach were discussed and emphasized in detail with respect to Multi-Agent Schedule-Planning Problem. Finally, an outline of the plan controller construction was promoted on a base of the logical approach in last chapter. Finally, some attempt to reconcile the logical approach with the abstract analysis-based was also proposed in terms of Sobolev's spaces as suitable to represent the robot trajectories and control functions in its idealized motion environment.

Novelty of the PhD-thesis with respect to this area forms a conjunction of the following main results.

A Two paradigmatic problems of temporal planning (*Temporal Traveling Salesman Problem (TTSP)* and *Multi-Agent Schedule-Planning Problem* (MA-SP-P)) were defined and proposed as a subject basis of temporal planning with fuzzy constraints.

B Fuzzy Allen's relations were represented by norms from the appropriate convolutions in the appropriate space of Lebesgue integrable functions.

C A portion of a theory for fuzzy Allen's relations in terms of real analysis and abstract algebra was elaborated.

C A hybrid approach to fuzzy temporal constraints was eleborated on a base of fuzzy Allen's relations in terms of convolutions. (In the context of MA-SP-P.)

E Two planning methods (STRIPS and Davis-Putnam procedure) were extended by temporal and preferential components.

F A hybrid approach to fuzzy temporal constraints with fuzziness introduced by preferences was elaborated in terms of Preferential Halpern-Shoham logic and its fibred models. (In the context of TTSP.)

G An outline of a construction of the plan controller was proposed on a base of the logical approach to fuzzy temporal constraints.

H An attempt of a synthesis of these two approaches was put forward in the context of the plan controller construction. This construction is carried out in 'logical terms', but it is complemented by analytic elements (trajectories of an agent's move are represented by the appropriate functions in Sobolev's spaces.)

From a methodological point of view, this PhD-thesis showed a unique heterogeneity. It was a natural consequence of many-dimensionality of the thesis considerations. A variety of used methods contains – for example: computational methods of measure theory and real analysis (particularly – with respect to convolution computing), formalization in terms of LTL and HS-logic, methods of construction of Büchi automata from LTL-formulas – due to ideas of M. Vardi. These methods are also supported by some methods of logical programming – in a declarative paradigm – such as: PROLOG. In addition, some more sophisticated method such as: the *Pavelka-Hajek's style completeness proof method* or *satisfiability checking via* reduction to the *Quantified Boolean Formula-satisfiability problem.*

Generally speaking, a methodology of the whole thesis formed (in author's belief) a synergy interplay between methods of analysis (in particular: of a comparative analysis) and synthesis. This synergy effect consists in a fact that both types of methods – together taken – allowed us to elucidate difficulties of earlier approaches, to find a more holistic conceptual depiction of temporal constraints and preferences and gave a more complementary picture of these issues.

7.2 Directions of Future Research

Indicating directions of a further research on temporal planning with fuzzy constrains and preferences may be relatively difficult issue for several reasons. At first, this area has already been intensively penetrated in many possible directions and many pretty general results have already been formulated. The next, many logical research on temporal logic system – such as research on subsystems of Halpern-Shoham logic – seems to made entrance to a final phase. This fact is sometimes reflected in titles of recent papers from this area beginning with such phrases as: "(Almost) last results in...". From this perspective, research on temporal planning with fuzzy temporal constrains and preferences seems to be in a period of "Autumn of Middle Ages".

Nevertheless, all of these well-known research paths, planning paradigms and approaches to temporal planning still wait for new combinations and more holistic approaches on a base of them. Some introductory example how to overcome a dichotomy between qualitative and quantitative temporal constraints was proposed in this thesis. It seems also that further research on temporal planning may take into account new components of rational behavior and intelligence such as : games, choices etc. – together combined with fuzzy constraints and preferences. Similarly, the well-known system of temporal logic such as Halpern-Shoham logic – penetrated from internal perspective – may be naturally combined with other temporal systems – such as LTL or μ-calculus – to be more effectively exploited in description of actions and timing. Halpern-Shoham logic seems to also suitable to be combined with purely modal logic systems. Deontic and dynamic Halpern-Shoham logic seem to constitute the most natural extensions of this system.

From a metalogical and theoretic point of view – temporal planning with fuzzy constraints and preferences seem to wait for a new type of combined semantics – similar to the Gabbay's fibred semantics – capable of covering different components. It is possible that the so-called reactive Kripke semantics – recently invented

(2013) by this author – appears to be the appropriate in many different contexts of temporal planning. It seems also that metalogical research on fuzzy logic systems for different concepts of real analysis and measure theory such as: Choquet's integrals, convolutions – possibly useful in AI and temporal planning in particular – form a promising research direction for a future. Some exemplification of fuzzy logic system for integrals and convolutions was presented in Appendix.

Finally, a tool side of temporal planning may be developed towards some fuzzy, temporal and preferential extensions of such languages of the declarative paradigm as PROLOG or *Answer Set Programming*.

7.2.1 List of the Papers of the Author of the Thesis with a Referrence to its Content.

2017:

Krystian Jobczyk, Antoni Ligeza: *STRIPS in Some Temporal-Preferential Extension.* ICAISC (1) 2017: 241-252 (**Chapter 2: section 2.4**)

Krystian Jobczyk, Antoni Ligeza: *Dynamic Epistemic Preferential Logic of Action.* ICAISC (2) 2017: 243-254 (**Chapter 4, Annex 1**)

Krystian Jobczyk, Antoni Ligeza: *Multi-valued Extension of Davis-Putnam Procedure.* ICAISC (2) 2017: 454-465 (**Chapter 2: section 2.4**)

Krystian Jobczyk, Antoni Ligeza, Krzysztof Kluza, *Fuzzy Temporal Reasoning in Terms of Possibilistic Temporal Logic – a Perspective of the Logical and Computational Complementation*, (submitted to:) WIRE Data Mining and Knowledge Discovery (Introduction II: section 4, chapter 2: section 2.2)

2016:

Krystian Jobczyk, Antoni Ligeza: *A General Method of the Hybrid Controller Construction for Temporal Planning with Preferences.* FedCSIS 2016: 61-70 (**Chapter 5**)
Krystian Jobczyk, Antoni Ligeza: *Multi-Valued Preferential Halpern-Shoham logic for relations of Allen and preferences.* FUZZ-IEEE 2016: 217-224(**Chapter 4**)

Krystian Adam Jobczyk, Antoni Ligeza: *Towards a new convolution-based approach to the specification of STPU-solutions.* FUZZ-IEEE 2016: 782-789 (**Appendix A**)

Krystian Jobczyk, Antoni Ligeza: *Why Systems of Temporal Logic Are Sometimes (Un)useful?* ICAISC (2) 2016: 306-316 (**Introduction II: section 5, chapter 5**)

Krystian Jobczyk, Antoni Ligeza, Krzysztof Kluza: *Selected Temporal Logic Systems: An Attempt at Engineering Evaluation.* ICAISC (1) 2016: 219-229

Krystian Jobczyk, Antoni Ligeza, Krzysztof Kluza: *New Integral Approach to the Specification of STPU-Solutions.* ICAISC (2) 2016: 317-328 (**Annex 1, Introduction II: section 4**,

2015: Krystian Jobczyk, Antoni Ligeza, Maroua Bouzid, Jerzy Karczmarczuk: *Comparative Approach to the Multi-Valued Logic Construction for Preferences.* ICAISC (1) 2015: 172-183 (**Introduction II: section: 3.6, chapter 4, Annex 3**)

Krystian Jobczyk, Antoni Ligeza: *Temporal planning in terms of a fuzzy integral logic (FLI) versus temporal planning in PDDL*. INISTA 2015: 1-8 (**Introduction I: section 5**)

Krystian Jobczyk, Antoni Ligeza, Jerzy Karczmarczuk: *Fuzzy-temporal approach to the handling of temporal interval relations and preferences*. INISTA 2015: 1-8
(**Introduction II: section: 3.6, chapter 4, Annex 3**)
2014: Krystian Jobczyk, Maroua Bouzid, Antoni Ligeza, Jerzy Karczmarczuk: *Fuzzy Logic for Preferences expressible by convolutions*. ECAI 2014: 1041-1042 (**Annex 1**)

Appendix A

Annexe 1 – Fuzzy Logic of Integrals for Allen's relations

In this appendix Fuzzy Convolution Logic (FLI) for Allen's relations will be introduced both syntactically and semantically. Furthermore, some metalogical features of this system such as a completeness with respect to infinite Kripke-based models and semi-completeness with respect to finite models will be investigated.

A.0.2 Requirements of the Construction and Notational Remarks.

Since the required FCL-FAIR system is conceived to represent Allen's relations in the convolution-based Ohlbach's depiction, it should be capable of representing convolutions and their typical algebraic features. In fact, we will consider two types of convolutions and integrals; at first – in the syntax of FCL-FAIR (pedantically: as integral and convolution symbols) and secondly – in the FCL-FAIR semantics (as interpretations of integral and convolution symbols).

1. Because of simple Lebesgue integrals – we need the following symbols:
   - *functional symbols of 1-type*: $\phi_t, \phi_{x-t}, \chi_t, \chi_{x-t}, \ldots$ – for Lebesgue integrable functions (in general), say $f, (t)f(x-t), g(t), g(x-t)$ (resp.) ,
   - *integral symbols*: $\int_{-\infty}^{\infty} \phi_t dt$, $\int_{-\infty}^{\infty} \phi_{x-t} dt, \ldots$
2. Because of Allen's interval-interval relations – obtained from the point-interval relations – we need the following symbols:
   - *predicates*:$\widehat{B}_t^i$, $\hat{D}_t^i$, $\widehat{M}_t^i \ldots$ – for the point-interval relations: $B_t(j), D_t(j), M_t(j) \ldots$ [1] and
   - *functional symbols of 2-type*: $\phi_x^{f^{i(x)}}, \chi_t^{j_g} \ldots$ – for syntactic representations of functions $f(x), g(t)$ (resp.) characterizing intervals i, j (resp.) etc.
   - *convolution symbols* for them: $\displaystyle\int_{t_0}^{t_1} \psi_t^i \widehat{R}_t^j dt$ – for convolution representation the Allen's interval-interval relations[2] and $\widehat{R}_t^j \in \{\widehat{B}_t^i, \widehat{D}_t^i, \widehat{M}_t^i \ldots\}$.

We also need 2 constant symbols for representing objects called fuzzy intervals: i, j and symbols for representing rational numbers as normalized values of convolutions: $\widehat{r}_1, \widehat{r}_2, \ldots$.

[1] There are the abbreviations for the point-intervals: 'before', 'during', 'meet', etc.

[2] We assume that direct names $\hat{B}(i,j)$, $\widehat{D(i,j)}$, $\widehat{M(i,j)} \ldots$ for the interval-interval relations does not belong to a language of FCL-FAIR, but to its meta-language, which – however – will be not defined here.

A.0.3 Syntax.

Language $\mathcal{L}$(FCL-FAIR) is taken from language of Łukasiewicz propositional calculus extended to the language of *Propositional Pavelka Logic* with the following connectives and constants: $\rightarrow, \neg, \iff, \wedge$ (weak conjunction), $\otimes$ (strong conjunction), $\vee$(weak disjunction), $\oplus$ (strong disjunction), propositional constants 0 and 1 and new constants: $\hat{r}_1, \hat{r}_2, \hat{r}_3 \ldots$ – see:[141]. The next, we enrich this language by the appropriate integrals and convolutions quantifiers and their corresponding functional symbols and predicates. It leads to the following definitions.

Definition 40 *The alphabet of FCL-FAIR consists of*[3]*:*

- *propositional variables:* $x, y, t \ldots$
- *functional symbols of 1-type:* $\phi, \chi, \phi_t, \phi_{x-t}, \chi_t, \chi_{x-t}, \ldots$
- *functional symbols of 2-type:* $\phi_x^{f^{i(x)}}, \chi_t^{j_g} \ldots$
- *predicates:* $\hat{B}_t^i$, $\hat{D}_t^i$, $\hat{M}_t^i$, $\hat{S}_t^i$, $\hat{F}_t^i \ldots$.
- *rational constant names:* $\widehat{r_1}, \widehat{r_2}, \ldots \widehat{0}, \widehat{1}$,
- *quantifiers:* $| \int ()dt, \int\int ()dxdt, , \int_{t_0}^{t_1} ()dt \ldots$
- *operations:* $\rightarrow, \neg, \vee, \wedge, \bullet, \oplus, \otimes, =$.

Definition 41 *The class of well-formed formulas* FOR *of $\mathcal{L}$(FCL-FAIR) consists of:*

1. *propositional variables and rational constants, functional symbols (of both types) and predicates, as atomic formulas.*
2. *The next - formulas obtained from functional symbols of 1-type by operations* $\neg, \vee, \wedge, \rightarrow, \oplus, \otimes, \int_{-\infty}^{\infty} ()dt$,
3. *the formulas obtained from either functional symbols of 2-type or functional symbols of 2-type and predicates by operations* $\neg, \vee, \wedge, \rightarrow, \oplus, \otimes, \bullet, \int_{t_0}^{t_1} ()dt$.
4. *Finally, formulas obtained from* $\phi_i \in$ FOR *and rational numbers by operations* $\neg, \vee, \wedge, \rightarrow, \oplus, \otimes, \bullet$ *belong to* FOR *as well.*

These classes of formulas exhaust the list of FOR *of $\mathcal{L}$(FCL-FAIR).*

Axiomatic system. The required system FCL-FAIR arises in $\mathcal{L}$(FCL-FAIR) by assuming the following axioms – partially considered by Hájek in [79]:

$$\int_{-\infty}^{\infty} (\neg\phi)dt = \neg \int_{-\infty}^{\infty} \phi dt,$$

$$\int_{-\infty}^{\infty} (\phi \rightarrow \chi)dx \rightarrow (\int_{-\infty}^{\infty} \phi dx \rightarrow \int_{-\infty}^{\infty} \chi dt)$$

$$\int_{-\infty}^{\infty} (\phi \otimes \chi)dt = ((\int_{-\infty}^{\infty} \phi dx \rightarrow \int_{-\infty}^{\infty} (\phi \wedge \chi)dt) \rightarrow \int_{-\infty}^{\infty} \chi dt))$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi dtdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi dydt\,^{4}(\text{Fubini theorem})$$

[3]This system is a (slightly modified) system introduced in [104].

[4]If both sides are defined.

and new axioms defining main algebraic properties of convolutions [5]:

$$\int_{t_0}^{t_1} \phi_t \bullet \chi_{x-t} dt = \int_{t_0}^{t_1} \phi_{x-t} \bullet \chi_t dt,$$
$$\widehat{r} \int_{t_0}^{t_1} \phi_t \bullet \chi_{x-t} dt = \int_{t_0}^{t_1} (\widehat{r}\phi_t) \bullet \chi_{x-t} dt, (\text{r}-constant) \text{ (associativity)},^{6}$$
$$\int_{t_0}^{t_1} \phi_t \bullet (\chi_{x-t} \oplus \psi_t) dt = \int_{t_0}^{t_1} \phi_t \bullet \chi_{x-t} dt \oplus \int_{t_0}^{t_1} \phi_t \bullet \psi_{x-t} dt \text{ (distributivity).}$$

As inference rules we assume: *Modus Ponens, substitution,* generalization rule for $\int$-symbol and two new specific rules:

$\frac{\phi}{\int_{-\infty}^{\infty} \phi dx}, \frac{\phi \to \chi}{\int_{-\infty}^{\infty} \phi dx \to \int_{-\infty}^{\infty} \chi dx}$ and the same rules for convolution integrals.

Definition 42 *FCL-FAIR is the smallest theory in $\mathcal{L}$(FCL-FAIR) with FOR – as its set of well-formed formulas, satisfying all these axioms and closed on inference rules as above.*

However, such a system describes more than convolution-depicted Allen's relations (and simple integrals). Observe that, if admit only formulas obtained from functional symbols of 2-type and predicates in the point 3 of FOR-definition and we formulate convolution axioms for such formulas, we obtain a restricted system, say FCL-FAIR^{Allen} – the appropriate for Allen's convolutions and only for them. Obviously, FCL-FAIR^{Allen} is a subsystem of FCL-FAIR.

A.0.4 Semantics.

In this section, a semantic interpretation of 'syntactic' integrals and convolutions will be given. We also give a semantic interpretation of all syntactic symbols. Begin with a definition of 'semantic integrals'. Assume, thus, that a non-empty set M and an algebra Alg of functions from $M \neq \emptyset$ to [0.1] containing each rational function $r \in [0.1]$ and closed on $\Rightarrow$ (see: [79], p. 240)) are given. It allows us to define 'semantic integrals'.

Definition 43 *('Semantic integrals') Each mapping $I : f \in Alg \mapsto If(x) \in [0,1]$satisfying properties expressed by FCL-FAIR axiomatic system is to be called a semantic integral.*

It means that for each $f, g \in Alg$ it holds:

$$I(1-f)dx = 1 - Ifdx,\ I(f \Rightarrow g)dx \leq (Ifdx \Rightarrow Igdx) \tag{A.1}$$

$$I(Ifdx)dy = I(Ifdy)dx \tag{A.2}$$

$$I(f(t)g(x-t)dt = Ig(t)f(x-t)dt \tag{A.3}$$

$$rIf(t)g(x-t)dt = I(rf(t))g(x-t)dt \tag{A.4}$$

All such I-mappings satisfying properties expressed by above axioms are to be called *semantic convolutions.*

Interpretation. We give an interpretation function for all entities of FCL-FAIR-syntax. Let assume, therefore, that $Int = (\triangle, \|\phi\|)$ with $\triangle \neq \emptyset$ and a (classical fuzzy) truth-value interpretation–function: $\|\|$ defined for the propositions of Łukasiewicz logic as follows: $\|\neg(\psi)\| = 1 - x$, $\| \to (\phi, \psi)\| = \min\{1, 1-x+y\}$,

[5] We only present the axioms for convolution in the infinite domain. The axioms in other cases are introduced in the same way.

[6] That is w.r.t. the scalar multiplication.

$\|\wedge(\phi,\psi)\| = \min\{x,y\}$, $\|\wedge(\phi,\psi)\| = \max\{x,y\}$, $\|\otimes(\phi,\psi)\| = \max\{0, x+y-1\}$, and $\|\oplus(\phi,\psi)\| = \min\{1, x+y\}$ for any $x,y \in$ MV-algebra [0,1]. In addition, $\|\widehat{r_i}\| = r_i$, for each $i \in \mathcal{N}$.

We inductively expand now this interpretation for new elements of the grammar $\mathcal{L}$(FCL-FAIR) as below.

syntax ($\phi \in \mathcal{L}(FCL\text{-}FAIR)$)	**fuzzy semantics** ($\|\phi\|_{FCL-FAIR}$)
$\phi_t, \phi_{x-t}, \chi_t \dots$	Lebesgue integrable functions $f(t), f(x-t), g(t), \dots$
$\widehat{B}^i_t$, $\hat{D}^i_t$, $\widehat{M}^i_t \dots$	relations: $B_t(j), D_t(j), M_t(j) \dots$
$\int_0^\infty \phi dt$	$I_0^\infty f dt$
$\int_{t_0}^{t_1} \phi_t \bullet \chi_{x-t} dt$	$I_{t_0}^{t_1} g(t) \star f(x-t))dt$ ($\star$ -conv. operator)
$\hat{r} \int_0^\infty f(t)dt$	$\|\hat{r}\| \star \|I_0^\infty f\|dt = rI_0^\infty f dt$
$\dfrac{\int_{t_0}^{t_1} \phi_t^{f^{i(x)}} \bullet \widehat{R}^j_{x-t} dt}{\widehat{N}}$	$I_{t_0}^{t_1} \frac{i(t)\star R(j)(x-t)dt}{N}$ for $R(j) \in \{B(j), M(j), D(j) \dots\}$

Definition 44 ***(FCL-FAIR Model.)*** *For a given function algebra Alg, Lebesgue integrable functions $f_1 \dots f_k, g_1 \dots g_l \in Alg$ (for $k, l \in \mathcal{N}$) we define the FCL-FAIR-model M as a n-tuple of the form:*

$$M = \langle |M|, \{r_0, r_1, ..\}, f_1 \dots f_k, g_1 \dots, g_l, I_0^\infty f^{i(x)} dt, I_{t_0}^{t_1} f^{i(x)} \star g_j dt \rangle,$$

such that:

- *$|M|$ is a countable (or finite) set,*
- *$\{r_0, r_1, \dots\}$ is a set of rational numbers belonging to $|M|$,*
- *for $i \in 1, \dots, k, j \in \{1, \dots, l\}$: $I_0^\infty f^{i(x)} g_j dt$ are convolutions of Lebesgue integrable functions $f^{i(x)}$ and g_j.*

If the model $M-$ as in the above def. 8 – is given, but it only contains convolutions of the form:

$$I_{t_0}^{t_1} f \star R\, dt, \tag{A.5}$$

where f is a characteristic function of some fuzzy interval and R, then such a model forms a model for FCL-FAIR^{Allen}-system.

If a model M is given, we write $\|\phi\|_{M,v}$ for a denotation of the *truth value under evaluation v* for each formula of $\mathcal{L}(FCL-FAIR)$ as a function: $\mathcal{L}(FCL-FAIR) \to [0,1]$. If M is a model and v is a valuation, then: $\|\hat{r}\|_{M,v} = r, \|x\|_{M,v} = a \in [0,1]$ for a variable x and for a predicate $Pred(t_1, \dots, t_k)$ it holds: $\|Pred(t_1, \dots t_k)\|_{M,v} = Rel(\|t_1\|_{M,v}, \dots \|t_k\|_{M,v})$ for Rel interpreting $Pred$ in a model M.

Example 2: Consider a sentence ϕ: 'A fuzzy interval $i(x)$ (almost) meets interval $j(x)$' and assume that they are based on finite supports (We can assume without losing of generality that $i(x), j(x) \subseteq [a,b]$ for some fuzzy interval based on $[a,b]$ – our intagrability limitation). Since *meet(i,j)*-relation in the convolution-based depiction is defined for such intervals as follows (see: [85, 86]):

$$\int_a^b \frac{Fin(i)(x)St(j)(x)dx}{N(Fin(i)(x), St(j)(x))}, \tag{A.6}$$

where $Fin(i)(x)$ and $St(j)(x)$ are the appropriate functions that cut points from intervals i, j (resp.) It we put $St(i)(x) = x$ and $Fin(j)(x) = -x^2$, then ϕ may be modeled by (at most denumerable) model:

$$\mathcal{M} = \langle Q \cap [a, b], i(x), j(x), -x^2, x, \int_a^b \frac{-x^3 dx}{N(x, -x^2)} \rangle \text{ and } Q \text{ denotes a set of rational numbers.}$$

A.1 The Pavelka-Hajek-style Completeness and Some Metalogical Properties of FCL-FAIR

This section is aimed as proving the completeness theorem for FCL-FAIR system with respect to the proposed models. We also justify undecidability of the set of tautologies of this system. This completeness proof is based on the Pavelka-Hajek completeness proof style from[79]. This methods works as follows. For a given theory T in a given propositional language $\mathcal{L}$, an MV-algebra [0,1] and a model M for T we introduce the *Truth degree* of ϕ over T as follows:

$$\|\phi\|_T = inf\{\|\phi\|_M | M \models T \text{ over } \mathrm{MV-algebra}\ [0,1]\} \tag{A.7}$$

and the so-called *Provability degree* defined as follows:

$$|\phi|_T = sup\{r \text{ rational } | T \vdash (\widehat{r} \rightarrow \phi)\} \tag{A.8}$$

Completeness in such terms is equivalent to the condition: $\|\phi\|_T = |\phi|_T$ for each ϕ of a fixed language of theory T.

Examples 3: If T is complete: its tautologies of T have both truth and provability degree 1; the contra-tautologies – 0.

In the current proof fact that such an equality holds for a case of "simple" integrals is exploited. Namely, $|\int \phi \mathrm{d}t| = \|\int \phi \mathrm{d}t\|$ for $\phi \in \mathcal{L}(RPL\forall)$ – as shown in [79].

Completeness theorem for *FCL-FAIR*: *For each theory T over $\mathcal{L}$(FCL-FAIR) and for each formula $\phi \in \mathcal{L}$(FCL-FAIR) it holds: $|\phi|_T = \|\phi\|_T$.*

Proof: It has been shown that for each theory T over $RPL\forall$ and for each formula ϕ it holds: $|\phi|_T = \|\phi\|_T$. As mentioned – the case of simple integral-formulas $\int \phi \mathrm{d}x$ was proved in [79]. It remains to consider more complicated case of convolution formulas for $\mathcal{L}$(FCL-FAIR). For clarity of the proof presentation, we omit the appropriate integration limits in convolution symbols.

Let us assume now that: $\phi = \int \psi_t \bullet \chi_{x-t} \mathrm{d}t$ for some ψ_t, $\chi_{x-t} \in \mathcal{L}$(FCL-FAIR). We show now that for such ϕ it holds: $\|\phi\|_T \leq |\phi|_T$. Indeed, if T proves $\widehat{r} \rightarrow \phi$, then $\|\widehat{r} \rightarrow \phi\|_M$ in a model M for this formula, then $r \leq \|\phi\|_M$. However, $\|\phi\|_T = inf\{\|\phi\|_M : \widehat{r} \rightarrow \phi$ is true in the model M$\}$. Hence, $\|\phi\|_T = r$ and then $r \leq sup\{r|$ T proves $\widehat{r} \rightarrow \phi\} = |\phi|_T$, which justifies the soundness. It remains to justify the inequality: $|\phi|_T \leq \|\phi\|_T$. For this purpose let assume that there is such a rational r that $|\int \phi_t \bullet \chi_{x-t} \mathrm{d}t| < r$, what means that our FCL-FAIR does not prove $\widehat{r} \rightarrow \int \phi_t \bullet \chi_{x-t} \mathrm{d}t$, hence $FCL-FAIR \bigcup (\int \phi_t \bullet \chi_{x-t} \mathrm{d}t) \rightarrow \widehat{r}$ is consistent and has a model, say M. Then $\|\int \phi_t \bullet \chi_{x-t} \mathrm{d}t \rightarrow \widehat{r}\|_M = 1$ and $|\int \phi_t \bullet \chi_{x-t} \mathrm{d}t) \rightarrow \widehat{r}|_M \leq r$ and hence $|\int \phi_t \bullet \chi_{x-t} \mathrm{d}t \rightarrow \widehat{r}|_T \leq \|\int \phi_t \bullet \chi_{x-t} \mathrm{d}t) \rightarrow \widehat{r}\|_T$, what finishes our proof in this case.

As above, soundness is easy to confirm, so it remains to sketch how completeness is proved. $|\int \phi_t(\chi_{x-t} \oplus \psi_{x-t})\mathrm{d}t|_T \leq \| \int \phi_t(\chi_{x-t} \oplus \psi_{x-t})\mathrm{d}t\|_T$ For this purpose let assume that there are such rationals r_1 and r_2 that $|\int \phi_t \bullet \chi_{x-t}|_T < r_1$ and $|\int \phi_t \bullet \psi_{x-t}|_T < r_2$. One can chose now such a rational $r(r_1, r_2)$ that $|\int \phi_t \bullet (\chi_{x-t} \oplus \psi_{x-t})\mathrm{d}t|_T < r$, what implies the fact that theory T does not prove $\widehat{r} \to \int \phi_t \bullet (\chi_{x-t} \oplus \psi_{x-t})\mathrm{d}t$, hence $T \bigcup (\int \phi_t \bullet (\chi_{x-t} \oplus \psi_{x-t})\mathrm{d}t) \to \widehat{r}$ is consistent and has a model, say $\mathbf{M}$.[7] Then we get $\| \int \phi_t \bullet (\chi_{x-t} \oplus \psi_{x-t})\mathrm{d}t \to \widehat{r}\|_M = 1$, therefore $\| \int \phi_t \bullet (\chi_{x-t} \oplus \psi_{x-t})\mathrm{d}t\|_M \leq r$ and hence $\| \int \phi_t \bullet (\chi_{x-t} \oplus \psi_{x-t})\mathrm{d}t\|_T \leq r$. It finishes the completeness proof for FLC-FAIR.

It is easy to see that the same completeness feature holds for restricted system FCL-FAIR^{Allen} with respect to FCL-FAIR^{Allen} models. We formulate this theorem as follows.

Theorem 12 *For each theory T over $\mathcal{L}$(FCL-FAIR)^{Allen} and for each formula $\phi \in \mathcal{L}$(FCL-FAIR) it holds:*

$$|\phi|_T = \|\phi\|_{\mathrm{T}}, \tag{A.9}$$

where $\|\|$ is defined for FCL-FAIR^{Allen} models.

Proof: Reasoning for $\dfrac{\int \phi_t^j \widehat{R}_t^j dt}{\widehat{N}}$, where $R_t^j \in \{B_t^j, M_t^j, D_t^j, \ldots\}$ and a normalization factor N forms a subcase of reasoning for convolution formulas. In fact, N ensures that $\|\frac{\phi_t^j \widehat{R}_t^j dt}{\widehat{N}}\| < 1$, hence $r \in [0,1]$ that $\|\dfrac{\int \phi_t^j \widehat{R}_t^j}{\widehat{N}}\| < r$ what reduces this case to the earlier case, considered in completeness theorem.

A.2 FCL-FAIR in Finite Models

It arises a natural question what about completeness of FCL-FAIR with respect to finite models?The answer of this question is not completely clear for the author of the paper. Anyhow, one can try to formulate some premises for a negative answer of this question.

Obviously, FCL-FAIR is still sound. In fact, each provable formula of $\mathcal{L}$(FCL-FAIR) is a 1-tautology (takes Truth value 1) – independently of a size of its model. Unfortunately, one can suspect that completeness does not hold – as it does not hold for its basis Hajek's system – see:[79], pp. 244-45. Nevertheless, one can formulate the following conjuncture.

Conjecture 1 *The set of all 1-tautologies of FCL-FAIR (all formulas true in all FCL-FAIR-models) is Π_1 (i.e.it has a quantifier form $\exists F$, where F is quantifier free).*

It seems that this conjuncture may be proved as the corresponding theorem for its basis system in [79]. Namely, having two finite models, say $\mathbf{M}$ and $\mathbf{M}_1$ and a distance d($\mathbf{M}$, $\mathbf{M}_1$) between them – as defined in [79], p. 245, one could formulate the following lemma:

Lemma 4 *There is a function $U(\phi, \epsilon, n)$, where formula $\phi \in FCL-FAIR, \epsilon > 0$ and natural $n > 0$, such that if $\boldsymbol{M}$ and $\boldsymbol{M}_1$ are two models with the same domain $|M|$ of a cardinality n and d($\boldsymbol{M}$, $\boldsymbol{M}_1$) $< U(\phi, \epsilon, n)$ than $|\|\phi\|_M - \|\phi\|_{M_1}| < \epsilon$.*

[7] This fact follows from the theorem: if *FCL-FAIR* is consistent, it has a model over standard MV-[0,1]algebra. This theorem could be obtained as an extension of the similar theorem for *RPL*$\forall$.

Since this proof was carried out in [79] for simple integrals, it seems that it may be repeated without difficulties for convolutions – as no their unique properties are needed there.

Having such a lemma one could verify now – as in [79] that if ϕ, $\mathbf{M}$, v are such that $\|\phi\|_{M,v} = 1$ and $\mathbf{M}$ is finite, then there is a $\mathbf{M}^{'}$ rational valued with the same domain such that $\|\chi\|_{M',v} = 1$. It shows that being 1-tautology (for finite models) is expressible by π_1-formula. Nevertheless, this research requires a deeper analysis.

Example 4: Consider the same sentence ϕ: 'A fuzzy interval $i(x)$ (almost) meets interval $j(x)$' and assume that they are based on finite supports (We can assume without losing of generality that $i(x), j(x) \subseteq [a, b]$ for some fuzzy interval based on $[a, b]$ – our integrability limitation). Since – as mentioned – *meet(i,j)*-relation in the convolution-based depiction is defined for such intervals as follows (see: [85, 86]):

$$\int_a^b \frac{Fin(i)(x)St(j)(x)dx}{N(Fin(i)(x), St(j)(x))}, \tag{A.10}$$

where $Fin(i)(x)$ and $St(j)(x)$ are the appropriate functions that cut points from intervals i, j (resp.) It we put $St(i)(x) = x$ and $Fin(j)(x) = -x^2$, then ϕ may be modeled by finite model:

$$\mathcal{M} = \langle \{r_1, r_2, \ldots, r_k\} \cap [a, b], i(x), j(x), -x^2, x, \int_a^b \frac{-x^3 dx}{N(x, -x^2)} \rangle$$

for such rational $r_1, r_2 \ldots, r_k$ that $\{r_1, r_2, \ldots, r_k, \ldots\} \cap [a, b]$ is non empty.

Appendix A

Annexe 2 – Fuzzy Allen's relations in a convolution-based depiction in algorithms.

This appendix contains a couple of algorithmically-presented fuzzy Allen's relations in a convolution-based depiction.

Algorithm: Compute Fuzzy Before(i,j)-relation

Input: fuzzy intervals i, j
Output: fuzzy relation $before(i, j)$
begin
| **If** $i = \emptyset$ OR i is positive infinite OR $j = \emptyset$
| **than return** 0
| **If** i is negative infinite
| | **than compute** $|i \cap_{min} j|$
| | **If** $|i \cap_{min} j| = \emptyset$
| | than return: $\frac{\int (i \cap_{min} j)(x) B(j)(x) dx}{|i \cap_{min} j|} D(i, j)$
| | **If** $i \cap_{min} j \neq \emptyset$
| | than return 1
| **If** $i, j \neq \emptyset$ and neither i, nor j are infinite
| **than return** $\frac{\int (i(x) B(j)(x) dx}{|i|}$.
end

The similar algorithm for a fuzzy start-relation looks as follows:

Algorithm: Compute Fuzzy Starts(i,j)-relation

Input: fuzzy intervals i, j
Output: fuzzy relation $starts(i, j)$
begin
| **If** $i, j = \emptyset$ OR i is neg.inf. OR j is neg. infinite
| **than return** 0
| **If** both i and j are negative infinite
| **than return**: $D(i, j)$
| **If** $i, j \neq \emptyset$ and both are finite OR both are pos. inf.
| **than return** $\frac{\int (F_1(i)(x)F_2(j)(x)dxD(i,j)}{N(F_1(i),F_2(j))}$.
end

The almost identical algorithm can be applied to the relation '$finish(i, j)$'. The only difference consists in a exchanging of negative infinite intervals for the positive infinite.

Algorithm: Compute Fuzzy Finishes(i,j)-relation

Input: fuzzy intervals i, j
Output: fuzzy relation $finishes(i, j)$
begin
| **If** $i, j = \emptyset$ OR i is pos.inf. OR j is pos. infinite
| **than return** 0
| **If** both i and j are positive infinite
| **than return**: $D(i, j)$
| **If** $i, j \neq \emptyset$ and both are finite OR both are neg. inf.
| **than return** $\frac{\int (F_1(i)(x)F_2(j)(x)dxD(i,j)}{N(F_1(i),F_2(j))}$.
end

The algorithm for a during relation is more complicated and it requires some comments. Although the main idea of this relation definition in terms of integrals is intuitive and it is based on the average the point-interval *during*-relation $D_p(j)$ (a point p is during j), the case of the infinite interval i appears to be slightly problematic one. In fact, it was shown that in this case we must cut this intervals in order to find all common point of i and j. In oder to consider a possible broad set of such points we take the first coordinates of i and j, say i^{fK} and j^{fK} and a *minimum* of them, $min\{i^{fK}, j^{fK}\}$. For the last coordinates for these intervals, say i^{lK} and j^{lK}, we take their *maximum*: $max\{i^{lK}, j^{lK}\}$. We prune the interval $i(x)$ in these two points: $min\{i^{fK}, j^{fK}\}$ and $max\{i^{lK}, j^{lK}\}$. In result we obtain a restricted interval i_1 as an effect of the *cut*-operation. More precisely, $i_1(x) = cut_{min\{i^{fK},j^{fK}\},max\{i^{lK},j^{lK}\}} i(x)$.

The algorithm in this case looks as follows:

Algorithm: Compute Fuzzy During(i,j)-relation

Input: fuzzy intervals i, j
Output: fuzzy relation $during(i, j)$
begin
| **If** $j = \emptyset$ OR i is infinite OR $i(\neg\infty) > j(\neg\infty)$ OR $i(+\infty) > j(+\infty)$
| **than return** 0
| **If** $i = \emptyset$
| **than return** 1
| **If** i is infinite
| | **than compute** $i_1(x) = cut_{min\{i^{fK}, j^{fK}\}, max\{i^{lK}, j^{lK}\}} i(x), |i_1|$
| | **If** $i_1 = \emptyset$
| | than return: 0
| | **If** $i_1 \neq \emptyset$
| | than return: $\int \frac{i_1 D_p(j)(x)}{|i_1(x)|} dx$
| **If** otherwise
| **than return** $\frac{\int (i(x) D_p(j)(x) dx}{|i|}$.
end

Algorithms for $equals(i, j)$ is similar.

Algorithm: Compute Fuzzy Equals(i,j)-relation

Input: fuzzy intervals i, j
Output: fuzzy relation $equals(i, j)$
begin
1. **compute: during(i,j)**
|**begin**
| | **If** $j = \emptyset$ OR i is infinite OR $i(\neg\infty) > j(\neg\infty)$ OR $i(+\infty) > j(+\infty)$
| | **than return** 0
| | **If** $i = \emptyset$
| | **than return** 1
| | **If** i is infinite
| | **than compute** $i_1(x) = cut_{min\{i^{fK}, j^{fK}\}, max\{i^{lK}, j^{lK}\}} i(x), |i_1|$
| | | **If** $i_1 = \emptyset$
| | | than return: 0
| | | **If** $i_1 \neq \emptyset$
| | | than return: $\int \frac{i_1 D_p(j)(x)}{|i_1(x)|} dx$
| | **If** otherwise
| | **than return** $\frac{\int (i(x) D_p(j)(x) dx}{|i|}$.
|**end**
2. **compute: during(j,i)**
|
3. **If** $during(i, j)$ and $during(j, i)$ are given
| **than return** $equals(i, j) = during(i, j) during(j, i)$
end

Assuming that relations $D(i, j)$ and $D(j, i)$ are given in the algorithm input, it can take very simple form:

Algorithm: Compute Fuzzy Equals(i,j)-relation

Input: fuzzy intervals i, j, $D(i, j)$, $D(j, i)$
Output: fuzzy relation $equals(i, j)$
begin
| **return** $E(i, j) = D(i, j)D(j, i)$
end

Appendix A

Annex 3 – The STPU Solution in the Integral-based Specification

This appendix provides an issue of Simple Temporal Problem under Uncertainty (extending STP) and a problem of its specification in terms of Ohlbach's integrals.

A.0.1 Simple Temporal Problems with Uncertainty (STPUs)

In [18] STP was extended by considering contingent events, whose occurrence is somehow controlled by exogenous factors sometimes referred to as 'Nature'. In this way we introduce the so-called *Simple temporal Problem under Uncertainty* (STPU). As STPU forms a further generalization of STP, it preserves some important point of the STP-based conceptualization. Namely:

a) as in STPs, activity durations are modeled by (compact) intervals,

b) the start times of all activities are controlled by the agent (this brings no loss of generality).

By a contrast, the end times (end time points) are divided into two classes: *required (requirements)* ('free' in [18]) and the *contingent* ones. It immediately introduces a distinction between STPU-variables for the *executable, required* and *contingent* time points. The first ones – are associated to performing agent, the second ones – are assigned by 'external' worlds (factors).

Definition 45 ***(STPU.)*** *A Simple Temporal Problem with Uncertainty (STPU) is such a 4-tuple* $N = \langle X_{exec}, X_{cont}, R_{eqr}, R_{cont} \rangle$ *that:*

- $X_{exec} = \langle e_1 \ldots, e_k \rangle$*: is the set of executable time-points,*
- $X_{cont} = \langle c_1 \ldots, c_k \rangle$*: is the set of contingent time-points,*
- R_{req} *- set of required constraints ,*
- R_{cont} *- set of contingent requirements .*

Example 36 *Consider the following scenario connected to the following two activities:* `preparing a documentation` *and* `delivering this documentation to the office`*. Assume that you can control their star points, but you cannot completely control their finishing. Assume also that you want to deliver a documentation relatively immediately after its preparing. Again, you can control a beginning of its delivering to the office, but you do not know where it will be finished. The contingent and required time points and possible constraints of this scenario are depicted in Fig. A.1.*

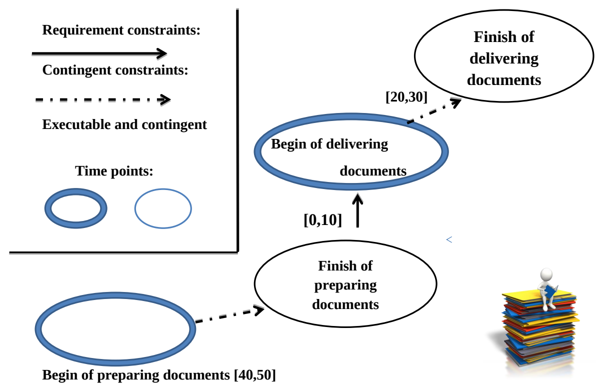


Figure A.1: An example of STPU

A.0.2 STPPU, or Simple Temporal Problem under Uncertainty with Preferences

Whereas STPU extends STP towards uncertainty and STPP – towards preferential rationality, the STPPU covers both of these components together integrated. STPPU may be viewed as an STPP, where time-points are partitioned into two classes: requirement and contingent ones – just as in an STPU. These facts are reflected by the following definition of STPPU.

Definition 46 *(**STPPU**.) Assume that STPU is defined and X_{exec} are constrained by an interval I^{Exec} and X_{cont} are constrainted by and interval I^{cont}. Assume also that:*

a) *a semi-ring $\mathcal{A} = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ is given*

b) *preferences $f^{Exec} : I^{Exec} \to A$ and*

c) *preferences $f^{Cont} : I^{cont} \to A$ are given.*

Then a Simple Temporal Problem with Uncertainty and Preferences is a 5-tuple
$N = \langle X_{exec}, X_{cont}, R_{eq}, R_{cont}, \{f_{Exec}, f_{Cont}\} \rangle$ such that:

- $X_{exec} = \langle e_1 \ldots, e_k \rangle$*: is the set of executable time-points,*
- $X_{cont} = \langle c_1 \ldots, c_k \rangle$*: is the set of contingent time-points,*
- R_{req} *- set of required constraints,*
- R_{cont} *- set of contingent requirements,*
- $\{f_{Exec}, f_{Cont}\}$ *is a set of preferences defined as above.*

Example 37 *Consider the observation satellite task with respect to the Amazon forest with details as depicted in Figure A.2 with some executable time points: `SO` (start observation), `EO` (end observation), `SC` (start clouds) and a contingent time point EC (end clouds), what introduced some uncertainty to the satellite observation. Nevertheless, observations should be continued in the clouding period, but with a smaller preference equal to 0,5. All these facts allow us to distinguish a contingent constraint `[SC-EC]` and required*

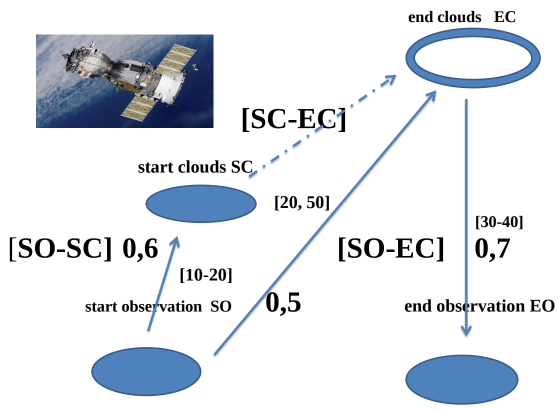


Figure A.2: An observation task of a satellite as an example of STPPU

constraints: `[SO-SC]`, `[SO-EC]`, `[EC-EO]`. *The associated preferences are associated to these constraints as depicted in Figure A.2. If we assume that preferences are defined over a semi-ring with a t-norm* min *as a measure of a global preference of the system, then we obtain* $\min\{0,6,0,5,0,7\}$ = *0,5.*

A.0.3 Integrals and Specification of Solution of STPU's

This section constitutes a continuation of a discussion on computational aspects of temporal planning with different constraints. Recently, two planning methods (STRIPS and Davis-Putnam procedure) were temporally and preferentially extended. Essentially, Allen's relations were exploited there as temporal constraints. Now, we intend to adopt Allen's relations in their convolution depiction to STPU in order to specify it solution.

It seems that the majority of approaches to the STP's and their extensions – seems to show some unilateralism. In fact, they are often based (more or less) on some common-sense observations and intuitions in solving of planning and scheduling w. r. t. the STP-problems. Although some intuitions and common-sense approaches allow us to obtain a correct STP- or STPU-solutions, they do not form the appropriate "machinery" to emphasize a graduality of adequacy of solutions.

Indeed, we often should consider the whole class of different solutions (more or less adequate from our point of view) in STPU-problems of planning and scheduling than a single, "local" one. We show how the proposed convolution-based approach to the representation of the Allen's relations between intervals is capable of satisfying such a sophisticated requirement. For that reason let us consider the following problem of the Multi-Agent Schedule-Planning Problem.

Sub-problem 1 *Consider an agent $n \in N$ performing a sequence of actions $\{a_1, a_2, \ldots, a_i, a_{i+1} \ldots a_k\}$ such that temporal constraints imposed on actions $\{a_1, a_2, \ldots, a_i\}$ form some fuzzy interval $i(x)$ with a support between [0,20] time units and temporal constraints imposed on actions $\{a_{i+1}, \ldots, a_k\}$ form some fuzzy interval $j(x)$ with a support between [50, 80] time units. Performing the action sequence $\{a_1, a_2, \ldots, a_i, a_{i+1} \ldots a_k\}$ requires at least 60 time units. When to begin its performing in order to finish this task effectively?*

We associate this main problem to the following (sub)problems supporting its solution in terms of the Allen's 'before'-relation for its convolution-based representation.

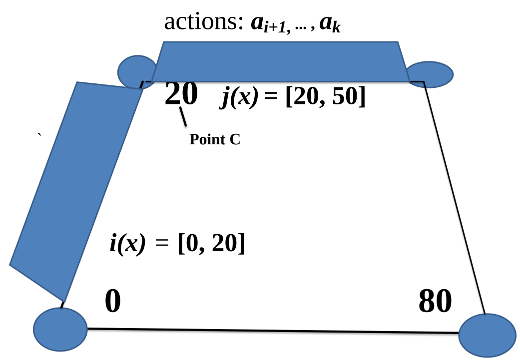


Figure A.3: STPU for the sub-problem of the Multi-Agent Schedulo-Planning Problem

Question 1 *Does the Allen relation 'before' between intervals i(x) and j(x) as above take a single or many values in the integral-based depiction? If many, show which values from [0,1]-interval can be taken by this relation in its integral-based depiction (if this 'before'-relation has a linear component?*

Question 2 *If the 'before'-relation can take values from [0,1], decide for which real parameters C >0 this relation takes values no smaller than 0,7?*

Methodological remarks and formal depiction of the problem

Let us observe that our initial observation task problem can be rendered in terms of the Allen "before"-relation between an interval $i(x)$ (of the observation beginning) and an interval $j(x)$ (for a gradual sky cloudiness). Let us consider that i and j are just such ones. Generally:

$$before(i,j)(x) = \frac{\displaystyle\int i(x)B(j)(x)dx}{max_a \displaystyle\int i(x-a)j(x)dx} \tag{A.1}$$

For simplicity, we are willing to adopt the following assumptions:

Assum1 all considered functions, i.e. $f^{i(x)}$ (the function characterizing fuzzy interval i), the function *B(j)* – representing the point-interval 'before'-relation will be dependent on the same argument x[1],

Assum2 both functions: $f^{i(x)}$ and *B(j)* are linear , i.e.

$\begin{cases} i(x) = & Ax\,, A<0\,, \\ j(x) = & Bx\,, B>0 \end{cases}$ (as depicted in Fig. A.4)

Assume that:

- a fuzzy interval *i(x)* represents the beginning of task performing and
- the second fuzzy interval *j(x)* represent the time

Let us observe that the task performing problem may be rendered in terms of the Allen "before"-relation between an interval *i(x)* (of the observation beginning) and an interval *j(x)* (for a time break of the task performing). Formally:

[1]Obviously, x still plays a role of temporal argument.

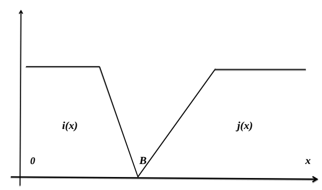


Figure A.4: Functions Ax and Bx (partially) defining the fuzzy intervals i and j

In order to find a simpler form of this function, let us compute:

$$\frac{AB\int x^2dx}{AB\int(x-a)xdx} = \frac{\int x^2dx}{\int(x^2-ax)dx} = \frac{x^3}{3[\frac{x^3}{3}-\frac{ax^2}{2}]} = \tag{A.2}$$

$$= \frac{x^3}{3[\frac{2x^3-3ax^2}{6}]} = \frac{2x^3}{2x^3-3ax^2} \tag{A.3}$$

for some done $a \in R$.

In order to solve the problem one need to investigate the function $f(x) = \frac{2x^3}{2x^3-3ax^2}$ and to find its possible values from [0,1]-interval.

Solving of the main sub-problem

Let us investigate analytic features of the function $\frac{2x^3}{2x^3-3ax^2}$ representing *before(i, j)(x)*-relation for fuzzy intervals *i(x)* and *j(x)* given by linear functions.

Domain and limits of *f(x)*. $2x^3-3ax^2 \neq 0 \iff x \neq 0$ or $x \neq 3/2a$, so $x \in \mathbb{R}/\{0, 3/2a\}$.
$\lim_{x\to-\infty}\frac{3x^3}{2x^3-3ax^2} = [\frac{-\infty}{-\infty}] = 1,\ \lim_{x\to\infty}\frac{3x^3}{2x^3-3ax^2} = [\frac{\infty}{\infty}] = 1,$
$\lim_{x\to(\frac{3}{2}a)^-}\frac{2x^3}{2x^3-3ax^2} = [\frac{2(\frac{3a}{2})^3}{0^-}] = -\infty,\ \lim_{x\to(\frac{3}{2}a)^+}\frac{2x^3}{2x^3-3ax^2} = [\frac{2(\frac{3a}{2})^3}{0^+}] = \infty.$
$\lim_{x\to(\frac{0}{)}^-}\frac{2x^3}{2x^3-3ax^2} = [\frac{0}{0^-}] = 0,\ \lim_{x\to(\frac{0}{)}^+}\frac{2x^3}{2x^3-3ax^2} = [\frac{0}{0^+}] = 0.$

It allows us to visualize the graph of this function as depicted on a fig.4: Therefore, $0 \leq \frac{2x^3}{2x^3-3ax^2} \leq 1$ for $x \in (-\infty, 0)$. Let us note that the above solution holds by assumption that intervals $i(x)$ and $j(x)$ meets in a point $x = 0$ as on the picture.

Since x represents time point (always > 0) we are only interested in this part of the function diagram, which is defined for positive x-values. Therefore, let us consider a function $F(x-B) = \frac{2(x-B)^3}{2(x-B)^3-3a(x-B)^2}$. Obviously, its graph stems from the earlier graph of $\frac{2x^3}{2x^3-3ax^2}$ (Figure A.5) *via* translation by a vector (0, B). It looks as depicted on Fig. A.6:

Having an outline of the $F(x-B)$-diagram we are interested in the part of this graph between $x_0 = 0$ and $x_1 =$ B. On the graph representation one can see that the function $F(x-B)$ is not defined for $x = B$, but $\lim_{B\to 0} F(x-B) = 0$. On can easily compute that $F(0-B) = \frac{-2B^3}{-2B^3-3B^2a} = \frac{2B^3}{-B^3+3B^2a} < 1$. It can be visualized as follows: Therefore, the investigated function takes the values from the interval $I = (0; \frac{2B^3}{2B^3+3B^2a})$.

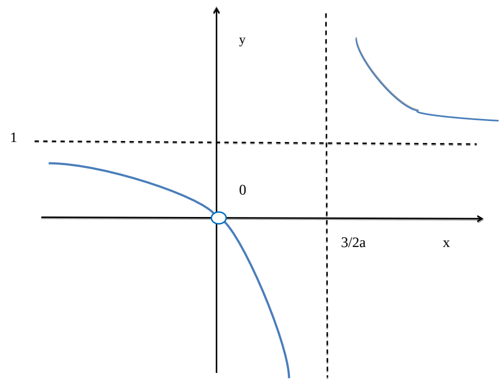


Figure A.5: An outline of the function $\frac{2x^3}{2x^3-3ax^2}$

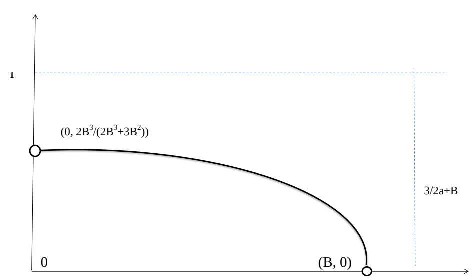


Figure A.6: The fragment of a graph of the function $F(x-B)$, which we are interested in – as a visual representation of our problem solution.

It is easy to see that the intuitive, "local" solution of the initial planning problem to begin the satellite observation possibly early (in a start point 0) was, somehow, justified, but is has also been expanded to the whole classes of solutions.

Example 38 *For B = 1 we obtain an interval $I_1 = (0, \frac{2}{2+3a})$ for done $a \in R$.*

Answer to the questions 1 and 2.

Thanks the arrangements, presented above, we are equipped by a "machinery" to face also the questions 1 and 2 associated to the main subproblem.

Question 3 *Does the Allen relation "before" take a single or many values in the integral-based depiction? If many, show which values from [0,1]-interval can be taken by this relation in its integral-based depiction (if this "before"-relation has a linear component?*

In order to answer the above question – recall that:

$$before(i,j) = \frac{\displaystyle\int i(x)B(j)(x)dx}{max_a \displaystyle\int i(x-a)B(j)(x)dx},$$

for some point-interval relation $B(j)(x)$. Meanwhile, we have just shown that – for linear functions defining the fuzzy intervals – this general definition can be given as:

$$before(i,j) = \frac{2x^3}{2x^3 - 3ax^2}$$

and $0 \leq before(i,j)(x) \leq 1$ holds for $x \in (0; \frac{2C^3}{2C^3+3C^2a})$. Nevertheless, by our assumption $C = 20$ (min) we obtain that: $x \in (0; \frac{2\bullet 20^3}{2\bullet 20^3+3\bullet 20^2a}) = (0; \frac{2\bullet 8000}{2\bullet 8000+1200a}$. Assuming for simplicity $a = 1$ we can get $x \in (0; \frac{16000}{17200}) = (0; 0,9302)$.

Therefore, the considered function takes values from the interval $(0; 0,9302)$.

Problem 2: *For which parameters $C > 0$ the $before(i,j)(x)$-relation takes values no smaller than 0,7?*

Solution: $0,7 \leq \frac{2C^3}{2C^3+3C^2}$. It is equivalent to $0 \leq \frac{2C^3}{2C^3+3C^2} - \frac{0,7\bullet(2C^3+2C^2)}{2C^3+3C^2} = \frac{1,3(C^3-2,1C^2)}{2C^3+3C^2}$. Let's consider the equation $\frac{1,3(C^3-2,1C^2)}{2C^3+3C^2} = 0$ (for $C \neq 0$). It leads to the equation $1,3(C^3-2,1C^2 = 0 \iff C^2(1,3C-2,1) = 0 \iff 1,3C = 2,1 \iff C = \frac{2,1}{1.3}$. Therefore, our inequality holds for $C \in (-\infty; 0) \bigcup (\frac{21}{13}; \infty)$. Because we are only interested in $C > 0$, so the only solution is given by an interval $(\frac{21}{13}; \infty)$.

Appendix A

Annexe 4 – Metalogical properties of Preferential Halpern-Shoham Logic (PHS)

In this appendix a fibred semantics will be deeply presented and some metalogical properties of Preferential Halpern-Shoham Logic such as:

- satisfiability problem for teh Preferential component of PHS,
- model-checking problem for PHS^L

will be investigated. Its presentation will be prefaced, however, by a detailed explanation mutal relationships between Kripke frames and transition systems such as IBIS –defined in chapter 3 of 'Contributions'.

In order to grasp some difference between these concepts we make use of a concept of *unraveling*. An n-frame $\mathcal{F} = \langle W, R_1, \ldots, R_n \rangle$ is called rooted if there is a $w_0 \in W$ such that $W = \{w \in W | w_0 R w\}$ where $R = \bigcup_{1 \leq i \leq n} R_i$. Such a w_0 is called a root of $\mathcal{F}$. Given a rooted n-frame $\mathcal{F}$ with root w_0, we can construct another n-frame $\mathcal{U}nr = \langle V, T_1 \ldots, T_n \rangle\rangle$ by taking V to be the set consisting of $\langle w_0 \rangle$ and all the tuples $\langle w_0, R_{i_1}, w_1, \ldots, R_{i_k}, w_k \rangle, k > 0$ of points in W and accessibility relations $R_{i_j} \in \{R_1, \ldots R_n\}$ such that $w_j R_{i_{j+1}} w_{j+1}$ whenever $j < k$ and, for any two points $\langle w_0, \ldots, w_k \rangle$ and x in V, it holds:

$$\langle w_0, \ldots w_k \rangle T_j x \iff \exists w \in W (x = \langle w_0, \ldots w_k, R_j, w \rangle). \tag{A.1}$$

The frame $\mathcal{U}nr$ is called just the *unraveling* of $\mathcal{F}$. An idea of the above construction is presented on a picture-fig.1. The Kripke frames – defined in the compact algebraic way as the *IBIS* above – can be unraveled in the similar way. It allows us to introduce a more compact definition of the *IBIS*-system. This reasoning justifies the following definition 31 from chapter 3.

Definition 47 *An IBIS-system is an unraveling of the generalized Kripke frame.*

A.1 A detailed presentation of fibred semantics

In order to explain this mechanism, let us take a mixed formula of the form $\psi = \langle \text{Pref} \rangle_i^\alpha \langle L \rangle \phi$, where ϕ is an atom in $\mathcal{L}(\text{HS}^L)$. Let us consider ψ as a formula of a language $\mathcal{L}^{Pref}$ (since its outer operator is $\langle \text{Pref} \rangle_i^\alpha$). From the point of view of $\mathcal{L}^{Pref}$ this formula has a form $\langle \text{Pref} \rangle_i^\alpha p$, where $p = [L]\psi$ is an atomic formula in $\mathcal{L}^{Pref}$. ($\mathcal{L}^{Pref}$ does not recognize $[L]\psi$.)

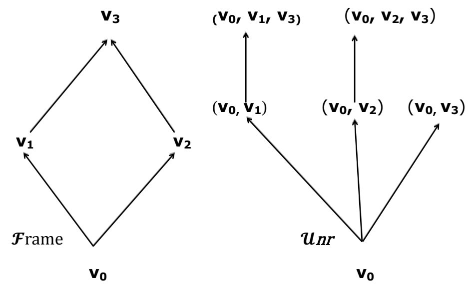


Figure A.1: $\mathcal{U}$nr is the unraveling of $\mathcal{F}$

In order to elaborate a model for $\langle \text{Pref} \rangle_i^\alpha p$ we take any model (based on a Kripke frame) of a language $\mathcal{L}^{Pref}$, say $\mathcal{M}_1 = \langle S^1, \precsim_i^\alpha, h_1 \rangle$, where S^1 is non-empty, $\precsim_i^\alpha$ is defined as earlier and h_1 is a valuation. We also take an interval $I \in S^1$ and we check whether $\mathcal{M}_1, I \models \langle \text{Pref} \rangle_i^\alpha p$.

According to the satisfaction definition 4 we have $\mathcal{M}_1, I \models \langle \text{Pref} \rangle_i^\alpha p \iff \exists I_1 (I \precsim_i^\alpha I_1 \text{ and } I_1 \models p)$. We have to check whether $\mathcal{M}_1, I_1 \models p$, or $\mathcal{M}_1, I_1 \models [L]\psi$. Unfortunately, $[L]\psi$ does not belong to $\mathcal{L}^{Pref}$, thus it cannot be evaluated in this model. In order to make it possible, we associate with such an interval I_1, a new model for $\mathcal{L}(\text{HS}^L)$:

$$\mathcal{M}_2^{I_1} = \langle S^{I_1}, R^L, h_2^{I_1} \rangle,$$

where R^L and $h_2^{I_1}$ denote a relation '**L**ater' between intervals and a valuation (*resp.*). In order to evaluate $[L]\psi$ at the associated $\mathcal{M}_2$ we introduce such a new function – called a*fibring function* **F** – that $\mathbf{F}(I_1) = \mathcal{M}_2^{I_1}$ and it holds the following equality:

$$\mathcal{M}_1, I_1 \models [L]\psi \iff \mathcal{M}_2^{I_1}, I_2 \models [L]\psi$$

for some interval I_2 of $\mathcal{M}_2$.

Because a model $\mathcal{M}_2$ is characterized by the interval I_2, we can identify $\mathbf{F}(I_1)$ with this new interval I_2 of the associated model $\mathcal{M}_2^{I_1}$. It allows us to formulate a new satisfaction condition in the form:

$$\mathcal{M}_1, I_1 \models [L]\psi \iff \mathcal{M}_2^{I_1}, \mathbf{F}(I_1) \models [L]\psi$$

We additionally impose on our fibred function **F** a condition of "switching semantics", i.e. for each $I \in \mathcal{M}_1$, it holds $\mathbf{F}(I) \in \mathcal{M}_2$ and for each $I \in \mathcal{M}_2$: $\mathbf{F}(I) \in \mathcal{M}_1$. We also assume that if $I_1 \neq I_2$, than also $\mathbf{F}(I_1) \neq \mathbf{F}(I_2)$. (**F**-images of two different intervals are different, too.) The same procedure can be repeated for other type of mixed formulas of $\mathcal{L}(\text{PHS}^\text{L})$.

A.2 Some Metalogical Results: Model checking Problem for PHS$^\text{L}$ and Satisfiability

The model checking problem has already been defined in preliminaries. In this section we this problem and its complexity for the newly introduced PHS^L. For this reason, only finite IBIS-based models will be considered. More precisely, we prove now that model checking problem for PHS^L belongs to the class of

PSPACE-problems. In addition we formulate a conjecture that this problem is hard, what would establish PSPACE-completeness of this problem. Anyhow, we leave this conjecture without justification.

Theorem 13 *The model checking problem for* PHS^L *is in PSPACE.*

Proof: It is enough to justify that an alternating model checking algorithm solves this problem in polynomial time. In this case we deal with an algorithm $\mathbf{solve}(\mathcal{M}_1, \mathcal{M}_2, \mathbf{F}, \varphi)$ for a fibred model $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, \mathbf{F})$ and such that both models $\mathcal{M}$ and $\mathcal{M}_1$ are based on the same IBIS system with a domain of cardinality $|S| = n$ and intervals no longer than some fixed K. Assume also that algorithms $\mathbf{solve}(\mathcal{M}_1, P, \phi)$ and $\mathbf{solve}(\mathcal{M}_2, I, \psi)$ are given for $\mathcal{M}_1$ for the $Pref$-component and $\mathcal{M}_2$ for the temporal component of PHS^L. This new algorithm works as follows.

Input: $\mathcal{M}_1$, P, $n = |\mathcal{M}_1|$, α_0, i_0, length of intervals $\leq K$, $\mathbf{solve}(\mathcal{M}_1, P, \phi)$
$\mathcal{M}_2$, I, $|\mathcal{M}_2| = n$, length of intervals $\leq K$,
$\mathbf{solve}(\mathcal{M}_2, I, \psi)$
Output: *Boolean*
We omit a case of atomic formulas. We only consider a case of single and combined modalities.

1. If $\phi = \langle \mathrm{Pref} \rangle_{i_0}^{\alpha_0} \psi$ for some fixed $\alpha_0 \in G$, then compute $\mathbf{solve}(\mathcal{M}_1, P', \psi)$ for such intervals P' that $P \precsim_{i_0}^{\alpha_0} P'$ for an accessibility relation $\precsim_{i_0}^{\alpha_0}$. If $\mathbf{solve}(\mathcal{M}_1, P', \psi)$ returns *False* for at least one of them, then return *False*, else return *True*.

2. If $\phi = \langle L \rangle \psi$ then compute $\mathbf{solve}(\mathcal{M}_2, I', \psi)$ for all such intervals (with a length $\leq K$) I' that I Later I'. If $\mathbf{solve}(\mathcal{M}_2, I', \psi)$ returns *True* for at least one such I', then return *True*, otherwise return *False*.

3. If $\phi = [\mathrm{Pref}]_{i_0}^{\alpha_0} \langle L \rangle \psi$ for the fixed α_0, then compute $\mathbf{solve}(\mathcal{M}_1, P', \varphi)$ for all such intervals P' that $P \precsim_{i_0}^{\alpha_0} P'$ for a given accessibility relation $\precsim_{i_0}^{\alpha_0}$ and $\varphi = \langle L \rangle \psi$. Unfortunately, $\mathbf{solve}(\mathcal{M}_1, P', \varphi)$ can return neither *True*, nor *False*, since $\mathcal{M}_1$ cannot evaluate $\varphi = \langle L \rangle \psi$. This formula can be evaluated in the model $\mathcal{M}_2$, since it recognizes the outer $\langle L \rangle$-operator of φ. Therefore one need change a model for $\mathcal{M}_2$ *via* $\mathbf{F}$-function and compute $\mathbf{solve}(\mathcal{M}_2, I, \varphi)$ for all such I that $\mathbf{F}(I)\, \mathbf{L}ater I$. If $\mathbf{solve}(\mathcal{M}_2, I', \psi)$ returns *True* for at least one such I', then the main algorithm $\mathbf{solve}(\mathcal{M}, \mathbf{F}, \varphi)$ returns *True*, otherwise returns *False*.

This algorithm works similarly for other combined formulas and uses the fibred function $\mathbf{F}$ to "move" between models.

It is easy to deduce that the algorithm needs a space size polynomially dependent on input size. It is enough to consider a boarder case, where the algorithm should 'check' all intervals of l-length (for $l = 1, 2, \ldots, k$ for some fixed k) accessible in the sense of the $\precsim_{i_0}^{\alpha_0}$ or the $\mathbf{L}$ater – relation. In the first step (for $[\mathrm{Pref}]_{i_0}^{\alpha_0}$), this algorithm checks at most $\sum_{i=1}^{k} n^k$; similarly – in case of $L\phi$-operator [1] . Hence, in two steps our algorithms must check at most $\sum_{i=1}^{k} n^k \sum_{i=1}^{k} n^k = \sum_{i=1}^{k} n^{2k} = \sum_{i=1}^{k} |S|^{2k}$ because $|S| = n$. Therefore, the number of intervals that must be checked by the algorithm polynomially depends on $|S|$.

PSPACE-hardness of this problem it is not clear for authors of this paper, but it seems that it can be obtained via an appropriate reduction to the problem of quantifier Boolean formula satisfiability (recognized as PSPACE-complete).

A.3 Satisfiability Problem.

In this conceptual setting, the *satisfiability problem* for a formula ϕ (of a given language $\mathcal{L}$) can be formulated as a problem whether or not ϕ admits an IBIS-system and an interval such that $IBIS, I \models \phi$. We intend to

[1] $\sum_{i=1}^{k} n^k$ is just a sum of all intervals of a length from 1 up to k taken from a n-elemental set S.

justify the PSPACE completeness of such a formulated satisfiability problem for FMPL. The proof uses a polynomial reduction of a satisfiability problem for FMPL to the satisfiability problem for $S4_m$ that is known to be PSPACE-complete, [142]. In this proof we adopt the proof ideas from [67], although the considered situation is more general than this one in [67]. In fact, authors of [67] refer to intervals epistemicaly indistinguishable (by some agent i); we refer to intervals α-similar (to an initial one).

Theorem 14 *(Satisfiability.) The satisfiability problem for FMPL is PSPACE-complete.*

Proof: (Short outline.) This proof uses a polynomial reduction of satisfiability problem for FMPL to satisfiability problem for a (specifically interpreted) $S4_m$ – known to be PSPACE-complete, [142]. For simplicity, we only consider the case of $m = 1$, or a single modal system S4 and FMPL with a single, fixed $\alpha \in G = \{\alpha\} \subset [0, 1]$, because G is a singleton in this case. More precisely, we will consider S4 interpreted semantically by (partially ordered) accessibility relation R only between α-similar intervals (in the sense, earlier defined). Denote such S4-system by $S4^\alpha$ and its relation R by R^α. Let also $\mathcal{A} = \{1, 2, \ldots, k\}$ be a set of agents.

In order to reduce the case of S4 to the case of FMPL, it suffices to show now that

$$\text{IBIS}, I \models_{\text{FMPL}} \phi \iff M \models_{S4^\alpha} f(\phi) \tag{A.2}$$

for some model M and a bijection f that transforms the formulas of FMPL into formulas of $S4^\alpha_m$ replacing each box-operator of $S4^\alpha_m$ by $[\text{Pref}]^\alpha_i \phi$ in polynomial time. The 'heart' of this reasoning is to justify that:

$$I_1 R^\alpha_i I_2 \text{ in M} \iff I_1 \precsim^\alpha_i I_2 \text{ in IBIS}, \tag{A.3}$$

for a fixed α and $i \in \mathcal{A}$. This inductive proof will be omitted.

The algorithm solving the SAT problem for FMPL works as follows. For a given formula ϕ, replace all occurrences of $[\text{Pref}]^\alpha_i$ by the box-operator $[\,]^\alpha_i$ of $S4^\alpha$ and check satisfiability of the resulting formula in this system. It easy to see that a function $f : \mathcal{L}(FMPL) \times \mathcal{A} \mapsto \mathcal{L}(S4^\alpha) \times \mathcal{A}$ with a parameter α such that

$$f([Pref]^\alpha_i \phi) = \Box^\alpha_i \phi \tag{A.4}$$

and f^{-1} are bijective polynomial-time reductions between FMPL and $S4^\alpha$ for each atomic ϕ[2] and each $i \in \mathcal{A}$.

A.4 State of the Art

The logic of Halpern-Shoham (HS) – introduced in [59] forms a multi-modal system suitable to represent the 13 well-known Allen's relations between intervals, [57], and constitutes a concurrent approach to temporal reasoning w. r. t. the Computational Tree Logic (CTL) or the Linear Temporal Logic LTL [143] – the more traditional and pointwise approaches. Although (chronologically) the first attempt to formalize intervals and some typical operations on them was proposed in [144] in terms of the Interval Temporal Logic (ITL) with chop-stars operators, HS – as the first logical system – refers to external relations between intervals of Allen's sort. Majority works referring to HS – such as [63, 66, 64] – elucidated only a variety of meta-logical features of HS and its subsystems.

An idea to investigate extensions of HS was recently proposed in [67]. According to it, this paper will be aimed at proposing some preferential extension of HS – denoted later by PHS. Nevertheless, an idea of combing preferences with Shoham-Halpern logic is not completely new and it seems to be reflected in a construction of the Property Specification Language (PSL) – capable of expressing preferences or wishes w. r. t. a behavior of considered systems by requirements imposed on temporal aspects of actions. It has

[2] We assume – without losing of generality– that both languages are built from the same propositional language, so they have the same atomic formulas.

also emerged that temporal logics prove to be susceptible to be combined with modalities – due to [145] – or to be considered in various fuzzy contexts – investigated in [146, 147]. Slightly against this state of art we intend to combine preferences and temporal relations – expressible in terms of HS – elucidating a fuzzy nature of preferences and rendering them by means of formalism proposed in [117] and discussed in [79]. We also propose an outline of the interval-based fibred semantics for combined formulas of PHS – due to pioneering ideas from [114], sharpened in [115].

Appendix A

Annexe 5 – Proofs of Some Geometric Properties of Robot Environment

A.1 Proof of Some Facts

Schwarz's inequality for trajectories:

$$\Big(\int_a^b x(t)y(t)d\mu\Big)^2 \leq \int_a^b x(t)^2 \int_a^b y(t)^2 d\mu. \tag{A.1}$$

Proof: If $\langle f, g\rangle = 0$, i.e. $\displaystyle\int_a^b f(x)g(x)dx = 0$, the therem is trivial. Thus assume that $f, g \neq 0$ and $\lambda = \langle f, g\rangle/\|g\|^2$. In our case: $\lambda = \dfrac{\int_a^b f(x)\overline{g(x)}d\mu}{\int_a^b |f(x)|^2 dx}$ and $\bar{\lambda} = \dfrac{\overline{\int_a^b f(x)\overline{g(x)}d\mu}}{\int_a^b |f(x)|^2 dx}$, $\langle f, g\rangle = \displaystyle\int_a^b f(x)\overline{g(x)}d\mu$, $\langle g, f\rangle = \displaystyle\int_a^b g(x)\overline{f(x)}d\mu$. Obviously:

$$\|g - \lambda f\|^2 = \|g\|^2 - \lambda\langle f, g\rangle - \bar{\lambda}\langle g, f\rangle + \lambda\bar{\lambda}\|f\|^2. \tag{A.2}$$

$$\text{Therefore, } \|g - \lambda f\|^2 = \int_a^b |g(x)|^2 d\mu - \frac{\int_a^b f(x)\overline{g(x)}d\mu}{\int_a^b |f(x)|^2 d\mu} \int_a^b f(x)\overline{g(x)}d\mu \tag{A.3}$$

$$-\frac{\overline{\int_a^b f(x)\overline{g(x)}d\mu}}{\int_a^b |f(x)|^2 d\mu} \int_a^b \overline{f(x)}g(x)d\mu + \frac{\overline{\int_a^b f(x)\overline{g(x)}d\mu}\int_a^b f(x)\overline{g(x)}d\mu}{\int_a^b |f(x)|^2 d\mu}. \tag{A.4}$$

$$\text{Thus } \|g-\lambda f\|^2 = \int_a^b |g(x)|^2 d\mu \; - \; \frac{\displaystyle\int_a^b \overline{f(x)} g(x) d\mu}{\displaystyle\int_a^b |f(x)|^2 d\mu} \int_a^b f(x)\overline{g(x)} d\mu \tag{A.5}$$

$$- \; \frac{\displaystyle\int_a^b \overline{f(x)} g(x) d\mu}{\displaystyle\int_a^b |f(x)|^2 d\mu} \int_a^b f(x)\overline{g(x)} d\mu \; + \; \frac{\displaystyle\int_a^b \overline{f(x)} g(x) d\mu}{\displaystyle\int_a^b |f(x)|^2 d\mu} \int_a^b f(x)\overline{g(x)} d\mu. \tag{A.6}$$

Finally:

$$0 \leq \|g-\lambda f\|^2 = \int_a^b |g(x)|^2 d\mu \; - \; \frac{\displaystyle\int_a^b \overline{f(x)} g(x) d\mu}{\displaystyle\int_a^b |f(x)|^2 d\mu} \int_a^b f(x)\overline{g(x)} d\mu. \tag{A.7}$$

It implies that:

$$\frac{\displaystyle\int_a^b f(x)\overline{g(x)} d\mu}{\displaystyle\int_a^b |f(x)|^2 d\mu} \int_a^b f(x)\overline{g(x)} d\mu \leq \int_a^b |g(x)|^2 d\mu. \tag{A.8}$$

and

$$\Big(\int_a^b f(x)\overline{g(x)} d\mu\Big)^2 \leq \int_a^b |g(x)|^2 d\mu \int_a^b |f(x)|^2 d\mu. \tag{A.9}$$

It finishes the proof of the Schwarz's inequality for trajectories.
Hölder inequality for trajectories:

$$\int_a^b x(t)y(t)d\mu \leq \Big(\int_a^b x(t)^p\Big)^{\frac{1}{p}} \Big(\int_a^b y(t)^q d\mu\Big)^{\frac{1}{q}}. \tag{A.10}$$

Minkowski inequality for trajectories:

$$\Big(\int_a^b (f_1(x) + \ldots f_n(x)\mathrm{d}\mu\Big)^{\frac{1}{k}} \leq \Big(\int_a^b f_1^k(x)\Big)^{\frac{1}{k}} + \ldots + \int_a^b f_n^k(x)\Big)^{\frac{1}{k}}. \tag{A.11}$$

Proof: Observe that:

$$\int_a^b \Big((f_1(x) + \ldots f_n(x)\Big)^k \mathrm{d}\mu = \int_a^b \Big((f_1(x) + \ldots f_n(x)\Big)^{k-1} \Big(f_1(x) + \ldots f_n(x)\Big)\mathrm{d}\mu \tag{A.12}$$

$$= \int_a^b f_1(x)\Big((f_1(x) + \ldots f_n(x)\Big)^{k-1} \mathrm{d}\mu + \ldots + \int_a^b f_n(x)\Big((f_1(x) + \ldots f_n(x)\Big)^{k-1} \mathrm{d}\mu \tag{A.13}$$

$$\leq \Big(\int_a^b f_1^k(x)\mathrm{d}\mu\Big)^{\frac{1}{k}} \Big(\int_a^b (f_1(x) + \ldots f_n(x))^k \mathrm{d}\mu\Big)^{\frac{k-1}{k}} \tag{A.14}$$

$$+ \ldots + \Big(\int_a^b f_n^k(x)\mathrm{d}\mu\Big)^{\frac{1}{k}} \Big(\int_a^b (f_1(x) + \ldots f_n(x))^k \mathrm{d}\mu\Big)^{\frac{k-1}{k}}. \tag{A.15}$$

This 'global' inequality is a consequence of the 'local' H$\acute{}$older inequalities of the form:

$$\int_a^b f_j \Big(\sum_{j=1}^n f_j(x)\Big)^{k-1} \leq \Big(\int_a^b f_j^k(x)\mathrm{d}\mu\Big)^{\frac{1}{k}} \Big(\int_a^b (f_1(x) + \ldots f_n(x))^k \mathrm{d}\mu\Big)^{\frac{k-1}{k}}, \tag{A.16}$$

for each $j = 1 \ldots, n$. This global inequality finishes the proof. □

Appendix A

Annexe 6 – Planning as Graph-Search – More Advanced Algorithm

This appendix does not introduce any new results. It only present some more advanced methods of planning as graph-search in a more didactic and explanatory way – for some complementary of the thesis considerations.

A.0.1 Graph Planning and its Short Analysis

The last type of planning paradigm relies on graph analysis and is called: 'graph-based planning' or simply: 'graph-planning'.

The graph-based planning is a procedure similar to iterative deepening and it relies on two types of activity:

- exploring a new (graph) search area from the initial one, and
- backtracking from a level P_i for some i[1] including all goal propositions.

The exact method of the backward graph search is the following. We begin with a level P_i which includes the required goal g, i.e $g \in P_i$. We look for a sequence π of actions – leading to g – which are mutually independent (non-mutex). During this backtrack, we define new subgoals of g: g_1, g_2, g_k, for some k, which allow us to find earlier actions leading to them. If finally level 0 is successfully reached, the corresponding sequence Π of actions forms a desired solution plan.

This procedure exploits the action dependence reflected by a concept of *mutex actions* ([3], p.121-123; for intuition, these are mutually exclusive sequences of actions). Dependence of actions is a converse notion with respect to their independence. The last one means – for a pairs of actions a and b – that an action a has nothing to do with preconditions and effects of b and an action b has nothing to do with preconditions and effects of a. Formally:

Definition 48 *Actions* `a` *and* `b` *are* `independent` *if:*

- $\mathsf{effects}(a)^- \cap \{\mathsf{precond}(b) \cup \mathsf{effects}(b)^+\} = \emptyset$,
- $\mathsf{effects}(b)^- \cap \{\mathsf{precond}(a) \cup \mathsf{effects}(a)^+\} = \emptyset$,

Definition 49 *Assume an iterative step i and associate to i some level P_i of a plan graph G and some set of actions A_i. Two actions* `a` *and* `b` *are* `mutex` *in A_i if either*

[1] Each of P_i is a part of a considered graph G admissible at the iterative step i.

- *`a` and `b` are dependent or*
- *if a precondition of `a` is mutex with a precondition of `b`.*

A set of mutex pairs at the level A_i is denoted by μA_i and a set of mutex pairs in P_i is denoted as μP_i.

In order to give a more detailed description of graph planning – in terms of the graph-plan algorithm – due to [3] – we also need two concepts, which the graph-plan algorithm is involved in:

- *extraction of a plan for a given goal g* and
- *expansion of the planning graph.*

Extraction. The role of the extraction procedure is to select actions respecting the final goal at the appropriate search levels.

If a goal g is achieved at the level 0, then a solution plan is empty (since no action is needed to reach g). If g belongs to a set of no-good tuples[2] at level i, then algorithm returns `failure` (at level i) since it means that a plan cannot be performed at this level. For π_i – a plan at the level i use a GP-search and if $\pi_i \neq$ `failure` then algorithm return(π_i) since π_i can be accepted. The extraction algorithm orders to exchange a set of no-good tuples $\triangle(i)$[3] by $\triangle(i) \cup \{g\}$ (at the level i) if GP-search fails to establish g at this level.

This method is reflected by the following algorithm (see: [3], p.127):

```
Extract(G, g, i)
begin
        i = 0 then return (⟨⟩)
        if g ∈ △(i) then return (failure)
        π_i ← GP − search(G, g, ∅, i)
        if π_i ≠ failure then return(π_i)
        △(i) ← △(i) ∪ {g}
        return(failure)
end
```

Expansion of planning graph. The role of the graph expansion in the graph-plan search procedure is to *enlarge the search space* for a use of a required plan detecting. This procedure refers to different components of the planning graph: to actions, their effects, mutex pairs of actions at a given level i. This variety of different graph components is also reflected in the expansion algorithm for a planning graph. In order to present it, let us assume that a planning graph up to level $i-1$ as a sequence of layers of nodes and mutex action pairs is given in the form:

$$G = \langle P_0, A_1, \mu A_1, P_1, \mu P_1, \ldots, A_{i-1}, \mu A_{i-1}, P_{i-1}, \mu P_{i-1} \rangle \tag{A.1}$$

We intend to expand this graph to a graph of the form:

$$G_1 = \langle G, A_i, \mu A_i, P_i, \mu P_i \rangle \tag{A.2}$$

which contains a new level i. In order to perform such an extension it is enough to add to the old graph G the following components:

- new mutually independent actions with preconditions from P_{i-1} ,
  In algorithmic form, it can be expressed as follows:
  $A_i \leftarrow \{a \in A | \texttt{precond(a)} \subseteq P_{i-1}$ and $\texttt{precond(a)}^2 \cap \mu P_{i-1} = \emptyset\}$

[2]There are such tuples of actions, which cannot be considered as a plan sequence. See: [3], p.134.

[3]These no-good tuples are such tuples, which cannot play a role of a plan sequence. $\triangle(i)$ denotes a no-good triple at the iteration level i.

- set of all positive effects of actions from A_{i-1},
  In algorithmic form:
  $P_i \leftarrow \{p | \exists a \in A_i : p \in \ \texttt{effect}^+(a)\}$
- pairs of possitive effects of pairs of (mutex) actions from A^2_{i-1} .
  In algorithmic form:
  $\mu A_i \leftarrow \{(a,b) \in A_i^2, a \neq b | \texttt{effects}^-(a) \cap [\texttt{precond(b)} \cup \texttt{effects}^+(b)] \neq \emptyset$ or
  $\texttt{effects}^-(b) \cap [\texttt{precond(a)} \cup \texttt{effects}^+(a)] \neq \emptyset$ or
  $\exists (p,q) \in \mu P_{i-1} : p \in \texttt{precond(a)}, q \in \texttt{precond(b)}\}$.

This algorithm in a compact form can be presented as follows:

$\texttt{Expand}(G = \langle P_0, A_1, \mu A_1, P_1, \mu P_1, \ldots, A_{i-1}\mu A_{i-1}, P_{i-1}, \mu P_{i-1} \rangle)$

$A_i \leftarrow \{a \in A | \texttt{precond(a)} \subseteq P_{i-1}$ and $\texttt{precond(a)}^2 \cap \mu P_{i-1} = \emptyset$

$P_i \leftarrow \{p | \exists a \in A_i : p \in \texttt{effect}^+(a)\}$

$\mu A_i \leftarrow \{(a,b) \in A_i^2, a \neq b \,| \texttt{effects}^-(a) \cap [\texttt{precond(b)} \cup \texttt{effects}^+(b)] \neq \emptyset$ or

$\texttt{effects}^-(b) \cap [\texttt{precond(a)} \cup \texttt{effects}^+(a)] \neq \emptyset$ or
$\exists (p,q) \in \mu P_{i-1} : p \in \texttt{precond(a)}, q \in \texttt{precond(b)}\}$

$\mu P_i \leftarrow \{(p,q) \in P_i^2, p \neq q | \forall a, b \in A_i, a \neq b :$
$p \in \texttt{effects(a)}, q \in \texttt{effects(b)} \rightarrow (a,b) \in \mu A_i\}$

return $\langle G, A_i, \mu A_i, P_i, \mu P_i \rangle$

The `Graph-plan` algorithm is an iterative, multi-stage reasoning – based on a bi-directional search:

- a forward search for a graph G expansion,
- a backward-search for a search of a Π plan.

At the input of the `graph-plan` we have a tuple $(\mathcal{A}, s_0, g)$, that is a set of actions $\mathcal{A}$, an initial state s_0 and a goal g. In output we get a plan Π leading from s_0 to g.
Graph-plan algorithm. This procedure can be briefly characterized as follows.

- In the initial step $i = 0$ we begin with the initial state s_0 (or with a set S_0 of such initial states [3], p. 126). At this level P_0 the graph G is a singleton $G = \{S_0\}$ and $\Pi = \emptyset$.
  In algorithmic form: $i \leftarrow 0, \Pi \leftarrow \emptyset$
  $G \leftarrow \langle S_0 \rangle$
- In the next step $i \geq 1$ we simultaneously:
  - expand a graph G (adding new actions with preconditions in S_0 to it) in a forward step and ,
  - choose non-mutex actions to Π (extract a plan) that satisfy a goal g in a backward step (if such actions do not exist, an algorithm returns `failure.` ,

  In algorithmic form:
  until $[g \subseteq P_i$ and $g \cap \mu P_i = \emptyset]$ **do**
  $j \leftarrow j + 1,$

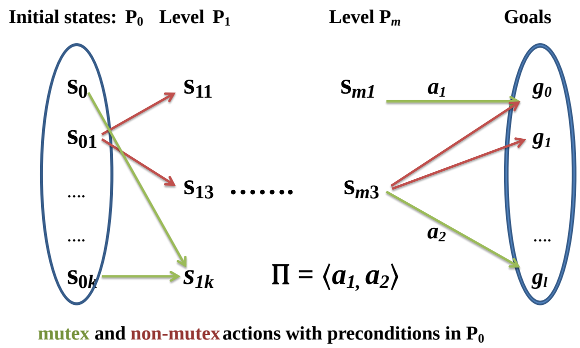


Figure A.1: The second step of the Graph-plan algorithm search

$G \leftarrow$ `Expand(G)`,
$\Pi \leftarrow$ `Extract(G, g, i)`
otherwise, if $[g \not\subseteq P_i$ or $g \cap \mu P_i \neq \emptyset]$ then return (`failure`).

- we repeat this reasoning in the next step $i+2$,
- We proceed until a level 0 will be successfully achieved by backtracking, that is until we find such actions α_1 and α_2 which required a subgoal $g_{i-1} \in P_1$. Then α_1 and α_2 are added to Π.

This procedure was presented in the following algorithm – based on a representation from [3](p. 129) and description from [5][4].

`Graph-plan`$(\mathcal{A}, s_0, g)$
begin
  $i \leftarrow 0, \Pi \leftarrow \emptyset, P_0 \leftarrow s_0$
  $G \leftarrow \langle P_0 \rangle$
  until $[g \subseteq P_i$ and $g \cap \mu P_i = \emptyset]$ or Fixedpoint(G) **do**
    $i \leftarrow i+1$
    $G \leftarrow$ `Expand(G)`
  if $g \not\subseteq P_i$ or $g \cap \mu P_i \neq \emptyset$ then return (`failure`)
  $\Pi \leftarrow$ `Extract(G, g, i)`
end

Example 39 ***(Traveling Plan in Germany)****. Consider an agent `Salesman` traveling between cities in Germany. His (possible) starting points is a chosen city from a set of initial states `Munich, Stuttgart` and his goal g is defined as a set = `Bremen, Hamburg`. His travel must be two-stages and can lead either by Bonn or Dresden, which may be achieved by `Salesman` as depicted on a map. Find the appropriate traveling plan for `Salesman` by means of the `Graph-plan` algorithm.*

Solution. The algorithm input in this case is determined by putting forward: $S_0 = \{$`Munich, Stuttgart`$\}$, g $= \{$`Bremen, Hamburg`$\}$ and a set of actions is the following $A = \{$`Move(Munich, Dresden), Move(Stuttgart,`

[4]The algorithm from [3] also refers to the case, when we find the so-called fixedpoint level of G, that is such the smallest level κ that each level $i > \kappa$ is identical with a level κ – see: [3], p. 125.

Bonn), Move(Bonn, Bremen), Move(Bonn, Hamburg), Move(Dresden, Hamburg)} – as depicted on a fig. 2. The Graph-plan algorithm works in this case as follows.

1. In the first step $i = 0$ we put:

   - $\Pi = \emptyset$ (a plan is empty),
   - $G = S_0 = \{$Munich, Stuttgart$\}$.

   Because g $=\{$Hamburg, Bremen$\} \not\subseteq G = \{$Munich, Stuttgart$\}$ the working algorithm condition is satisfied, so

2. In the next step we take $i = 1$ and we

   - extract the plan by non-mutex actions satisfying g,
   - expand the graph by actions with preconditions in s_0, namely:

   we obtain:

   - $\Pi = \{$Move(Bonn, Bremen), Move(Dresden, Hamburg)$\}$ –as mutually non-mutex actions [5],
   - $G = \{$Munich, Stuttgart, Move(Munich, Dresden), Move(Stuttgart, Bonn)$\}$

   We also define a new goal $g_1 = \{$Bonn, Dresden$\}$. Because $g_1 = \{$Bonn, Dresden$\} \not\subseteq G = \{$Munich, Stuttgart, Move(Munich, Dresden), Move(Stuttgart, Bonn)$\}$, the algorithm working condition is satisfied, thus:

3. In the next step we assume $i = 2$ and

   - extract the plan Π by non-mutex actions satisfying g_1,
   - expand the graph G by actions with preconditions in $S_1 = \{$Bonn, Dresden$\}$, namely:

   we obtain:

   - $\Pi = \{$Move(Bonn, Bremen), Move(Dresden, Hamburg), Move(Stuttgart, Bonn), Move(Munich, Dresden)$\}$ –as mutually non-mutex actions,
   - $G = \{$Munich, Stuttgart, Move(Munich, Dresden), Move(Stuttgart, Bonn)$\}$.

   We also define a new goal $g_2 = \{$Munich, Stuttgart$\}$. Because now $g_2 \subseteq P_0$, we can finish a procedure, as the initial states have been already achieved by a back-tracking. Finally, we obtain two plans:

   $\Pi_1 = \{$Move(Bonn,Bremen), Move(Stuttgard, Bremen)$\}$ and

   $\Pi_2 = \{$Move(Dresden, Hamburg), Move(Munich, Dresden)$\}$.

A.1 Some Formal Properties of STP.

With respect to STPs (in the graph-based depiction) a couple of useful facts are known, which somehow elucidate an operational simplicity of STPs.

Colollary 4 *STPs have the following features:*

- *Any consistent STP is backtrack-free, that is decomposable relative to the constraints in its d-graph (see: [20])*

[5] Actions Move(Bonn, Bremen), Move(Bonn, Hamburg) are mutex as they have the same precondition Bonn.

Figure A.2: The Salesman's traveling between cities in Germany

- *The set of feasible values for variable X_i in a distance graph of a consistent STP is $[-d_{t0}, d_{0t}]$.([20])*
- *The d-graph of an STP can be found by applying* `Floyd-Warshall's All Pairs Shortest algorithm` *to the distance graph with a complexity of $O(n^3)$, where n is the number of variables.*

An idea of the `Floyd-Warshall's (All Pairs Shortest) Algorithm is relatively simple`. The first steps of this algorithm introduction an assumption that a weight a_{ij} should be associated to each edge of a distance graph G. In particular, because of a lack of a distance between i and i itself, we put: $d_{ii} = 0$.

In algorithmic formulation:

for $i := 0$ to n **do** $d_{ii} \leftarrow 0$
for $i, j := 0$ to n **do** $d_{ij} \leftarrow a_{ij}$

Secondly, algorithm recursively (from $k = 1$ to n) introduces a method to measure a distance between edges. In a step $k = 0$, that is for $d(i, j, 0)$, we simply put $d(i, j, 0) = a_{ij}$ as there is no edge between i and j. In for $k \geq 1$, that is for $d(i, j, k)$ we compare $d(i, j)$ with $d(i, k) + d(k, j)$ as there is (at least one) edge between i and j. In this case we take a minimal of values between $d(i, j)$ and $d(i, k) + d(k, j)$.

In algorithmic formulation:
for $k := 0$ to n **do**
    for $k := 0$
        $d_{ij,0} = a_{ij}$,
    for $k :\geq 1$
        $d_{ij,k} \leftarrow min\{d_{ij0}, d_{ik0} + d_{kj0}\}$

`Floyd-Warshall's algorithm`
for $i := 0$ to n **do** $d_{ii} \leftarrow 0$
    for $i, j : 0$ to n **do** $d_{ij} \leftarrow a_{ij}$
for $k := 0$ to n **do**
    for $k := 0$
        $d_{ij,0} = a_{ij}$,
    for $k :\geq 1$
        $d_{ij,k} \leftarrow min\{d_{ij0}, d_{ik0} + d_{kj0}\}$

Appendix A

Annexe 7 – Lindstrøm Characterizing Theorem and its Simplified Proof

This appendix[1] presents the so-called Lindstrøm Characterizing Theorem. This theorem delivers metalogical criteria for an expressive power for formal languages by the reference to the first order logic. It may be also exploited for different temporal planning languages. In the current depiction, we make use a more general category of an *abstract logic*. Anyhow, it might represent different planning languages, such as languages of: a descriptive or an attributive logic or PDDL (PDDL+).

The merit-related impact of this chapter is to propose a simplified proof of the original proof of Lindstrøm theorem from his sophisticated and pioneering paper [43].

A.1 Terminological Background

A.1.1 Languages

EL = *Elementary Logic* (= first order logic).

L = *Abstract Logic* = EL *plus* some generalized quantifiers....

Example 40 *Q_1: 'There are finitely many', Q_2: ' There are uncountable many', Q: 'There is many P's satisfying ϕ', Henkin quantifiers, etc.*

L is EL enriched to be a portion of *Second Order Logic* (We can quantify there not only over individuals, but also over: functions, relations and sets of individuals).

A.1.2 Structures

We deal with structures of the form:

$$\langle A, R_0, R_1, \ldots, R_{m-1} \rangle, \tag{A.1}$$

where R_i is a t_i-ary relation on A. The sequence $t = \langle t_0, \ldots, t_{m-1} \rangle$ is the *type* of structures.

Each structure $\langle A, R_0, R_1, \ldots, R_{m-1}, S_0, \ldots, S_k \rangle$ is an *expansion* of $\langle A, R_0, R_1, \ldots, R_{m-1} \rangle$.

aK will denote a class of structures (of the same type).

[1] This appendix presents the simplified proof of Lindstrøm Characterizing Theorem, which in a modified version was presented in the philosophical PhD-thesis 'Skolem-Loewenheim Theorem and its philosophical utility' (2011).

Example 41 *Assume that K contains structures of the type $t = \langle t_0, \ldots, t_k \rangle$. Then both $\langle A, R_1, \ldots, R_k \rangle \in K$ and $\langle B, S_1, \ldots, S_k \rangle \in K$, but $\langle A, R_1, \ldots, R_m \rangle$ does not, if only $m \neq k$.*

We also distinguish the following classes of structures:

F – the family of the classes of structures, K, consisting of $\langle A, S \rangle$, $S \subseteq A, S$ is finite, nonempty and such that for each $n > 0$ there is a structure with S of cardinality n. (*Semantic definition of finiteness.*)

$\mathbf{F}_\omega$ – the same as F, with infinite A only.

$\mathbf{Mod}_{t,L}(\phi)$ – the class of structures (of type t), in which (L-)formula ϕ is true.

$\mathbf{C}_L$ – the family of all such classes, i.e. the family of all classes of structures, which make true (L-)formulas.

We show later that neither F, nor F_ω can be characterized by a first order formula.

We say that K is L-*characterizable* iff it is characterized by some M in C_L, that is, members of M (class of structures) are just expansions of the members of K.

Example 42 *Assume that the class of structures $K = \{\langle A, R \rangle : A \neq \emptyset, R$ is a relation$\}$ and let $M = \{\langle A, R, S_0, S_1 \rangle$: $A \neq \emptyset, R, S_0, S_1$ are relations$\}$ belong to C_L for some language L. Since members of M, i.e. structures (of the form): $\langle A, R, S_0, S_1 \rangle$ are expansions of members of K, or the structures (of the form): $\langle A, R \rangle$, we can say that K is characterized by M. Due to the above definition – K is also L-`characterizable`.*

A.1.3 Two Further Definitions

- $\mathrm{L} \subseteq \mathrm{L}'$ (L' *enriches* L) iff for every type t and L-formula ϕ there exists L'-formula such that $\mathrm{Mod}_{t,L}(\phi) = \mathrm{Mod}_{t,L'}(\phi')$. (L and L' are 'true' in the same classes of structures of t-type).
- $\mathrm{L} \subseteq_{inf} \mathrm{L}'$ iff the same holds, but $\mathrm{Mod}_{t,L}(\phi) = \mathrm{Mod}_{t,L'}(\phi')$ holds for their *infinite* members. (L and L' are 'true' in the same classes of *infinite* structures of t-type. L and L' agree for infinite structures, but not necessary for the finite ones.)

Example 43 *Consider $\mathcal{L}(Arytm_1)$ and $\mathcal{L}(Arytm_2)$ and let:*

$$\mathrm{Mod}_{t,\mathcal{L}(Arytm_1)}(\phi) = \{\langle \{0, 1, 2 \ldots, 7\}, succ, 0, 1 \rangle, \langle \omega, Succ, 0, 1 \rangle\}. \tag{A.2}$$

and

$$\mathrm{Mod}_{t,\mathcal{L}(Arytm_2)}(\phi) = \{\langle \{0, 1, 2 \ldots, 44\}, succ, 0, 1 \rangle, \langle \omega, Succ, 0, 1 \rangle\}. \tag{A.3}$$

Since $\mathrm{Mod}_{t,\mathcal{L}(Arytm_1)}(\phi) = \mathrm{Mod}_{t,\mathcal{L}(Arytm_2)}(\phi)$ for `infinite` structures only , i.e. for $\mathcal{F} = \langle \omega, succ, 0, 1 \rangle$, it might only holds:

$$\mathcal{L}(Arytm_1) \subseteq_{inf} \mathcal{L}(Arytm_2) \text{ or } \mathcal{L}(Arytm_2) \subseteq_{inf} \mathcal{L}(Arytm_2) \tag{A.4}$$

A.1.4 Theorems

Theorem 15 ***(SL property of L)****: If $K \in C_L$ and K has an infinite member than K has a countable member.*

Example 44 *If $\langle \mathcal{R}, +, \bullet \rangle \in K$, then also $\langle \omega, +, \bullet \rangle \in K$.*

Theorem 16 ***(Compactness Theorem)****: If $\forall n K_n \in C_L$ and*

$$\bigcap \{K_n : n = 0, 1, 2 \ldots\} = \emptyset, \tag{A.5}$$

then

$$\exists \text{ such } m \text{ that } \bigcap_{n<m} K_n = \emptyset. \tag{A.6}$$

Theorem 17 *(**Lindstrøm Theorem** 1969) If $EL \subseteq L$ and L has* `compactness` *and* `SL property`*, then* $L\subseteq$ *EL.*

A.2 Proof of Lindstrøm Theorem

In order to prove Lindstrøm Theorem, one needs to formulate two important lemmas.

The first one will assert that if L (having SL property) enriches EL, then one can characterize F_ω in it. (Informally: such L is capable of rendering the notion of 'countability'.)

The second one asserts that the same L can characterize F. (Informally, L is capable of rendering the notion of 'finiteness'.)

Theorem 18 *If L has SL property and EL$\subseteq_{inf}$ L but not conversely, then some member of F_ω is L-characterizable.*

Outline of the proof. Assume that K_0 is in C_L, or $K_0 = \mathrm{Mod}_{t,L}(\phi)$, but for no M in C_{EL}, K_0 and M have the same infinite members. Then, due to SL property (here we use SL), for no M in C_{EL}, K_0 and M have the same countable members. We can say that K_0 is not EL-definable for countable structures.

Now, let us assume, for simplicity, that K_0-members are $\langle A, R\rangle$, where R is a binary relation, i.e. $R \subseteq A \times A$. We will define now a new class of structures, K_1 in EL.

K_1 is defined in EL as the class of structures with infinite universes, with a distinguished subset of the universe, with a linear order on that subset, which has the first element and each element has an immediate successor. Finally, there are finite correlations of the elements of the universe that establish a *partial isomorphism* of two distinguished relations R_1 and R_2. It is assumed that the correlations can always be made longer.

In order to return to the L-characterizability in terms of K_0 and K_1, let us define a new class of structures K_2 as follows. K_2 consists of structures of the same type as those in K_1, but $\langle A, S, R_1, R_2 \ldots\rangle$ is in K_2 iff:

$$\langle A, R_1\rangle \in K_0 \text{ and – simultaneously – } \langle A, R_2\rangle \notin K_0.$$

Finally, define $K = K_1 \cap K_2$. Since K_0 is in C_L, so is K. In this way, we have proved that K characterizes a member of F_ω. □

Theorem 19 *If L has SL property and EL$\subseteq_{inf}$ L but not conversely, then some member of F is L-characterizable.*

Proof. If L non$\subseteq_{inf}$ EL, then Theorem 16 gives the conclusion. Suppose, therefore, that $L \subseteq_{inf}$ EL, but L non$\subseteq$ EL. (Informally: they agree only for infinite structures.) Formally:

1 There are classes K, M such that:

- K is in C_L, i.e. $K = Mod_{t,L}(\phi)$, (Repeat: C_L = class of structures, where L is true.)
- M is in C_{EL}, i.e. $M = Mod_{t,EL}(\phi)$ and
- K is not in C_{EL}. (All from L $\subseteq_{inf}$ EL).

2 The classes K and M do not have the same members of power n, for arbitrary large natural number n

(From L non $\subseteq$ EL and the fact that L and EL 'agree' for infinite structures).

Now let the class N consists of the structures $\langle A, S, R_0, \ldots, R_{m-1}\rangle$, $S \subseteq A, S$ is non-empty and let $\langle A, S, R_0, \ldots, R_{m-1}\rangle$ be in $(K - M) \cup (M - K)$ (We should reject infinite structures as belonging to the non-empty intersection $K \bigcap M$). Then N is in C_L (as K is in C_L). In fact, if $\langle A, S, R_0, \ldots, R_{m-1}\rangle$ is in $K - M$ (N is a class of structures with finite universes). Then is in C_L.

Note that N may represent the class of all $\langle A, S\rangle \in F$, for S-finite, in such a sense that the extension $\langle A, S, R_0, \ldots, R_{m-1}\rangle$ is in N. Hence, $\langle A, S\rangle \in F$ is representable by L-characterizable class N. Thus, a member of F (the structure $\langle A, S\rangle$ alone) is L-characterizable. □

Theorem 20 *(**Lindstrøm Theorem**). If $EL \subseteq L$ and L has* `compactness` *and* `SL property`*, then $L \subseteq EL$.*

Proof. If EL$\subseteq$ L and L has SL and L non$\subseteq$ EL, then it can be shown that compactness fails for L. By Theorem 1 – there is a class K_0 (of type t) in C$_L$, i.e. $K_0 = Mod_{t,L}(\phi)$, which characterizes a member of F. For $n > 0$ let K_n let K_n be a class of structures $\langle A, S, R_0, R_1 \ldots, R_{m-1}\rangle$ in K_0 such that S has at least n elements. (Formally: $K_n = Mod_{t,L}(\phi \wedge \exists \geq n$ elements).

On one hand, $\bigcap\{K_n : n < m\} = \{\langle A, S, R_0, \ldots\rangle : |S| \geq n\}$, so $\bigcap\{K_n : n < m\} \neq \emptyset$. On the other hand, $\bigcap\{K_n : n = 0, 1, 2 \ldots\} = \emptyset$.

In fact, having established an arbitrary k, we can always find that $\bigcap\{K_n : n < k\} \neq \emptyset$, say K_1^{Fin}. However, we can always skip to $k+1$ to require $\bigcap\{K_n : n < k+1\}$. However, $K_1^{Fin} \neq \bigcap\{K_n : n < k+1\}$, as a class of k-elemental structures may be only such an intersection (K_1^{Fin} contains $k-1$-elemental structures.) Thus, we should take another candidate, say K_2^{Fin}. Nevertheless,, we can skip once again to $k+2$ and we should reject K_2^{Fin} for the same reasons, etc. It is clear that we cannot a non-empty intersection for all K_n for $n = 0, 1, \ldots$. $\square$

Appendix A

Annex 8–The Chosen Mathematical Concepts Used in the Thesis

A.1 Measure theory

. This part constitutes a brief introduction to the concept of measure and product measure.

Let X be a a non-empty set and $\mathcal{A}$ be such a unique family of subsets of X, which contains $\emptyset$ and containing both $\bigcup_{k=1}^{\infty} A_k$ and $\bigcap_{k=1}^{\infty} A_k$ if only each subset $A_1, \ldots A_k$ of X belong to $\mathcal{A}$. In that case, $\mathcal{A}$ is called to be *ρ-algebra*. If $\mathcal{A}$ is given now, we can define a function $\mu : \sum \to [0, \infty]$, which satisfies the following conditions:

1. **Non negativity** : for all $A \in \mathcal{A}$, it holds: $\mu(A) \geq 0$.
2. **Null empty set** : $\mu(\emptyset) = 0$.
3. **Denumerable additivity** : For all pairwise disjoint sets $A_i (i = 1, 2 \ldots)$ in $\mathcal{A}$:

$$\mu(\bigcup_{k=1}^{\infty} A_k) = \sum_{k=1}^{\infty} \mu(A_i). \tag{A.1}$$

Each function μ satisfying conditions 1-3 is called: 'an *(additively denumerable) measure*' – simply – 'a measure' and $(X, \mathcal{A})$ is called: 'a measurable space'. Some commonly exploited examples of measures are: probability measures and the Lebesgue measures. The last one enables of introducing the following definition of the Lebesgue integrals.

Definition 50 *(**Measurable function**). Let (X, ρ_1, μ) and (Y, ρ_2, v) be measurable spaces. We say that a function $f : X \to Y$ is said to be measurable if the preimage of E under f is in ρ_1, for each $E \in \rho_2$. Formally:*

$$f^{-1}(E) = \{x \in x : f(x) \in E\} \in \rho_1, \forall E \in \rho_2. \tag{A.2}$$

A.2 Probability theory

. Assume that a measurable space $(\Omega, \mathcal{F}, P)$ is given, where:

- Ω is a non-empty set,

- $\mathcal{F} \subseteq 2^{\Omega}$ is a usual σ-algebra – a set of substes of Ω, called *events*, such that:
  - $\Omega \in \mathcal{F}$,
  - If $A \in \mathcal{F}$, then also $(\Omega/A) \in \mathcal{F}$,
  - If $A_i \in \mathcal{F}$, for $i = 1, 2, \ldots$, then $\bigcup_{i=1}^{\infty} A_1 \in \mathcal{F}$.
- $P : \mathcal{F} \to [0, 1]$ – *the probability measure* – is:
  - a denumerable additive, i.e. $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$ and
  - $P(\Omega) = 1$.

Then the space $(\Omega, \mathcal{F}, P)$ forms a *probability space* and P is called a *probability* (measure).

Definition 51 ***(Distribution function)****. Assume that a probability measure $P : 2^{\Omega} \to [0, 1]$ is given. The* ***`distribution function`*** *of a real random variable is the function $F : \mathbb{R} \to \mathbb{R}$:*

$$F(x) = P((-\infty, x]), \; x \in \mathbb{R}. \tag{A.3}$$

Definition 52 ***(Density function)****. Asuuming that a distribution function F is given a continuous function of a random variable X, then there exists such a function $f \geq 0$ that for each x it holds:*

$$F(x) = {}_{-\infty}^{x} f(t)\mathrm{d}t = P(-\infty, x]). \tag{A.4}$$

Example 45 *Assuming that $f(x) = 0$, for $x < 0$ and $f(x) = e^{-x}$, for $x \geq 0$ is a density function, one can compute its distribution and probability $P(1 \leq X \leq 3)$ as follows:*

$$F(x) = {}_{-\infty}^{x} f(t)\mathrm{d}t = {}_{-\infty}^{0} 0\mathrm{d}x + \int_{0}^{x} e^{-t}\mathrm{d}t = 1 = e^{x}. \tag{A.5}$$

$$P(1 \leq X \leq 3) = F(3) - F(1) = e^{-1} - e^{-3}. \tag{A.6}$$

Definition 53 ***(Expected value, Mean value)****. If X is a random variable defined over a probability space $(X, \mathcal{F}, P)$, then the* ***`expected value`*** *$E(X)$ of X is given by the Lebesgue integral:*

$$E(X) = \int_{\Omega} X\mathrm{d}P, \tag{A.7}$$

if this integral exists.

In particular case, if a probability denisty function $f(x)$ is given, then E(X) may be computed as:

$$E(X) = \int_{-\infty}^{\infty} xf(x)\mathrm{d}x, \tag{A.8}$$

provided that this integral is finite.

Example 46 *If a density function $f(x) = e^{-x}$ is given, then the expected value $E(X)$ of a random variable X is computed as follows (e^{x} is continuous):*

$$E(X) = \int_{-\infty}^{\infty} xf(x)\mathrm{d}x = \int_{-\infty}^{\infty} xe^{-x}\mathrm{d}x = [-xe^{-x} + \int e^{-x}\mathrm{d}x]_{0}^{\infty} \tag{A.9}$$

$$= [-xe^{-x} - e^{-x}]_{0}^{\infty} = 1. \tag{A.10}$$

A.3 Algebra

In chapter 2 of 'Contributions' a concept of linear(vector) space is recalled. This object is defined as follows.

Definition 54 *A* `vector(linear)` *space is an ordered triple* $(X, \bullet, K)$, *where* X *is an additive Abelian group,* K *is a field and* $\bullet$ *is a function from* $K \times X$ *into* X, *whose values at* (α, x) *is* αx, *such that for* $\alpha, \beta \in K$ *and* $x, y \in X$ *the following conditions hold:*

a $\alpha(x + y) = \alpha x + \alpha y$;

b $(\alpha + \beta)x = \alpha x + \beta x$;

c $\alpha(\beta x) = (\alpha\beta)x$;

d $1x = x$, *where* 1 *forms the multiplicative unit of* K.

Example 47 *Let* K *be a field, let* $n \in \mathbb{N}$ *and* $X = K^n$. *For* $\boldsymbol{x} = (x_1, \ldots, x_n)$ *and* $\boldsymbol{y} = (y_1, \ldots, y_n)$ *in* X *and for* $\alpha in K$ *let us define:*

$$\boldsymbol{x}+\boldsymbol{y} = (x_1 + y_1, \ldots, x_n + y_n), \;\; \alpha\boldsymbol{x} = (\alpha x_1 \ldots, \alpha x_n). \tag{A.11}$$

Then X *forms a vector space over* K.

Example 48 *Assume that* $C(\mathbb{R})$ *is a class of continuous real functions on* $\mathbb{R}$. *Defining the operations* $+$ *and* $\bullet$, *for each* $\alpha \in \mathbb{R}$ *and for all* $x \in \mathbb{R}$ *as follows:*

$$(f + g)(x) = f(x) + g(x), \;\; (\alpha f)(x) = \alpha f(x), \tag{A.12}$$

one can show that $(C(\mathbb{R}), +, \bullet)$ *is a vector space.*

Bibliography

[1] T. L. Dean and R. McDermott, "Spatio-temporal reasoning and linear inequalities," *Artificial Intelligence Laboratory, AIM-875*, vol. Cambridge, 1986.

[2] V. Lifschitz, "On the semantics of strips," *Reasoning about Actions and Plans*, vol. (eds.)Georgeff and Lansky, pp. 1–9, 523–530, 1990.

[3] P. Traverso, M. Ghallab, and D. Nau, "Automated planning: theory and practice," *Elsevier*, vol. 2004, 1997.

[4] A. L. Blum and M. L. Furst, "Fast planning throught planning graph analysis," *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1636–1642, 1995.

[5] ——, "Fast planning throught planning graph analysis," *Artificial Intelligence (IJCAI)*, vol. 90, pp. 281–300, 1997.

[6] A. Blum and W. Langford, "Probabilistic planning in the graphplan framework," *Proceedings of the European Conference on Planning (ECP)*, pp. 319–322, 1999.

[7] S. Kambhampati and X. Yang, "On the role of disjunctive representations and constraint propagation in refinement planning," *Proceedings of the International Conference on Knowledge Representation and Reasoning (KR)*, 1996.

[8] S. Kambhampati, "Planning graph as (dynamic) csp: Exploiting ebl, ddb and other csp techniques in graphplan," *Journal of Artificial Intelligence Search*, vol. 12, pp. 1–34, 2000.

[9] S. Kambhampati and R. Nigenda, "Distinct based goal ordering heuristics for graphplan," *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 315–322, 2000.

[10] M. Flood, "The travelling-salesman's problem," *Operations Research*, vol. 5, pp. 61–75, 1957.

[11] K. Menger, "Das botenproblem," *Ergebnisse eines Mathematisches Kolloquiums*, vol. 2, pp. 11–12, 1932.

[12] I. Heller, "The travelling salesman's problem," *Proceedings of the Second Symposium in Linear Programming*, vol. 1, pp. 27–29, 1955.

[13] B. Cheang, H. Li, A. Lim, and B. Rodrigues, "Nurse rostering problems–a bibliographic survey." *European Journal of Operational Research*, 2003.

[14] E. K. Burke, T. Curtois, R. Qu, and G. V. Berghe, "A scatter search approach to the nurse rostering problem."

[15] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, "Staff scheduling and rostering: A review of applications, methods and models," *European Journal of Operational Research*, vol. 153(1), pp. 3–27, February 2004.

[16] R. Qu and F. He, "A hybrid constraint programming approach for nurse rostering prolems." *Technical report, School of Computer Science*, 2008, Nottingham.

[17] L. Khatib, P. Morris, R. Morris, and F. Rossi, "Temporal reasoning about preferences," in *Proceedings of IJCAI-01*, 2001, pp. 322–327.

[18] T. Vidal and H. Fargier, "Handling contingency in temporal constraints networks: From consistency to controllabilities," *Journal of Experimental and Technical Artificial Intelligence*, vol. 11, no. 1, pp. 23–45, 1999.

[19] F. Rossi, N. Yorke-Smith, and K. Venable, "Temporal reasoning with preferences and uncertainty," in *Proceedings of AAAI,*, vol. 8, 2003, pp. 1385–1386.

[20] R. Dechter, Meiri, and J. Pearl, "On fuzzy temporal constraints networks," *Twmporal constraints network*, vol. 49(1-3), pp. 61–95, 1991.

[21] F. Pednault, "Synthetizing plans that contain actions with contex-dependent effects," *Computational Intelligence*, vol. 4(4), pp. 356–372, 1988.

[22] ——, "Exploring the midle ground between stripsa nd the situation calculus," *Proceedings of the International Conference on Knowledge Representation and Reasoning (KR)*, vol. 4(5), pp. 324–332, 1989.

[23] ——, "Adl-and the state-transition model of action," *Journal of Logic and Compution*, vol. 4(5), pp. 467–512, 1994.

[24] H. Kautz and B. Selman, "Planning as satisfiability," *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 1992.

[25] J. Harisson, *Handbook of practical logic and automated reasoning.* 79-90: Cambridge University Press, 2009.

[26] A. Emerson and E. Clark, "Characterizing correctness properties of parallel programs using fixpoints," *Automata, Languages and Programming*, 1980.

[27] E. Calrk and A. Emerson, "Design ans synthesis of synchronization skeleton usingbranching-time temporal logic," *Logic of Programs*, pp. 52–71, 1981.

[28] E. Clark, A. Emerson, and A. Sistla, "Automatic verification of finite-state concurrent systems using temporal logic specifications," *ACM Transactions on Programming Languages and Systems*, vol. 8(2), p. 244, 1986.

[29] E. Giunchiglia and P. Traverso, "Planning as model checking," *Proceedings of the European Conference on Planning(ECP)*, vol. LNAI, pp. 1–20, 1999.

[30] A. Cimatti, E. Giunchiglia, F. Giunchiglia, and P. Traverso, "Planning via model checking: A decision procedure for ar," *Proceedings of the European Conference on Planning(ECP)*, vol. LNAI 1348, pp. 130–142, 1997.

[31] A. Cassandra, L. Kaelbling, and M. Littman, "Acting optimally in partially observable stochstic domains," *Proceedings of the National Conference on Artificial Inetlligence(AAAI)*, vol. AAAI Press, 1994.

[32] L. Kaelbling, M. Littman, and A. Cassandra, “Partially observable marcov decision processes for artificial intelligence,” *Proceedings of Reasoning with Uncertainty in Robotics*, 1995.

[33] T. Dean, L. Kaelbling, J. Kirman, and A. Nicholson, “Planning with deadlines in stochastic domains,” *Proceedings of the National Conference on Artificial Inetlligence(AAAI)*, vol. AAAI Press, pp. 574–79, 1993.

[34] M. Asuncion, L. Castillo, J. Fernandes-Olivares, and O. Garcia-Perez, “Handling fuzzy temporal constraints in a planning environment,” *Annals of Operational Research*, vol. 155(1), pp. 391–415, November 2007.

[35] A. Pnueli, “The temporal logic of program focs,” pp. 46–57, 1977.

[36] D. McDermott, “The 1998 ai planning system competition,” *AI Magazine*, vol. 21(2), 2000.

[37] M. Fox and D. Long, “Pddl+: Planning with time and metric sources,” *Technical report*, vol. University of Durham, 2001a.

[38] ——, “Pddl2.1: An extension to pddl for expressing temporal planning domains,” *Technical report*, vol. University of Durham, 2001b.

[39] ——, “An extension to pddl for expressing temporal planning domains,” *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.

[40] ——, “The third international planning competition: Temporal and metric planning,” *Preprints of The Sixth International Conference on Artificial Intelligence Planning and Scheduling*, vol. 20, pp. 115–118, 2002.

[41] R. Goldblatt, “First-order definability in modal logic,” *The Journal of Symbolic Logic*, vol. 40, pp. 35–40, 1975.

[42] F. Baader, J. Horrock, and U. Sattler, “Description logics,” in *Handbook of Knowledge Representation*, vol. Elsevier, 2007.

[43] P. Lindstrom, “On extension of elementary logic,” *Theoria*, vol. 35, pp. 1–11, 1969.

[44] P. Fikes and N. Nilsson, “Strips: A new appraoch to the application of theorem proving to problem solving,” *Artificial Intelligence*, vol. 2(3-4), pp. 189–208, 1971.

[45] J. McCarty and J. Hayes, “Some philosophical problems from the standpoint of artificial intelligence,” *Machine Intelligence*, vol. Edinburgh University Press, pp. 469–502, 1969.

[46] H. Putnam and M. Davis, “A computing procedure for quantification theory,” *Journal of the ACM*, vol. 7(3), pp. 201–215, 1960.

[47] C. Green, “Application of theorem proving to problem solving,” *Proceedings of thr International Joint Conference on Artificial Intelligence(IJCAI)*, 1969.

[48] N. J. Nilsson, *Pinciples of Artificial Intelligence.* Tioga Publishing, 1980.

[49] J. Beckford, G. Logemann, and D. Loveland, “A machine program for theorem proving,” *Communications of the ACM*, vol. 5(7), pp. 394–397, 1962.

[50] R. Dechter and I. Rish, “Directional resolution: The davis-putnam procedure, revisited.” *Principles of Knowledge Representation and Reasoning: Proc. of the Fourth International Conference KR*, pp. 134–145, 1994.

[51] C. Ramirez-Atencia, G. Bello-Orgaz, M. Rodriguez Moreno, and D. Camacho, "A hybrid moga-csp for multi-uav mission planning," *GECCO(Companion)*, pp. 1205–1208, 2015.

[52] ——, "Performance evaluation of multi-uav cooperative mission planning models," *ICCI*, vol. 2, pp. 203–212, 2015.

[53] ——, "Gampp: Genetic algorithm for uav mission planning problems," *IDC*, pp. 167–176, 2015.

[54] ——, "Branching to find feasible solutions in unmanned air vehicle mission planning," *IDEAL*, pp. 286–294, 2014.

[55] M. Vilain and H. Kautz, "Constraint propagation algorithms for temporal reasoning," *Proceedings of teh National Conference on Artificial Intelligence(AAAI)*, pp. 377–382, 1986.

[56] A. Gerevini, "Incremental qualitative temporal reasoning: algorithms for the point algebra and ord-horn class," *Artificial Intelligence*, vol. 166, pp. 37–80, 2005.

[57] J. Allen, "Maintaining knowledge about temporal intervals," in *Communications of ACM*, 26(11)1983, pp. 832–843.

[58] B. Nebel and B. H-J., "Reasoning about temporal relations: A maximal tractable subclasses of allen's interval algebra," in *Journal of the ACM 42*, 1995, pp. 43–66.

[59] J. Halpern and Y. Shoham, "A propositional modal logic of time intervals," *Journal of the ACM*, vol. 38, pp. 935–962, 1991.

[60] M. Vilain, H. Kautz, and P. Beek van, "Constraint propagation algorithms for temporal reasoning: A revised raport," *Reading in Qualitative Reasoning about Physical Systems*, vol. Morgan Kaufmann, 1989.

[61] P. Beek and W. Manchak, "The design and experimantal analysis of algorithms for temporal reasoning," *Journal of Artificial Intelligence Research*, vol. 4, pp. 1–18, 1996.

[62] D. Bresolin, D. Della Monica, A. Goranko V., Montanari, and G. Sciavicco, "Undecidablility of the logic of overlap relation over discrete linear orderings," *Proc. of M4M 6: the Workshop on Methods for Modalities*, 2009.

[63] A. Montanari, I. Pratt-Hartmann, and P. Sala, "Decidability of the logic of a reflexive sub-interval relations over finite linear orders," in *TIME*, Monterey, CA, 2010.

[64] A. Montanari, G. Puppies, P. Sala, and G. Sciavicco, "Decidability of the interval temporal logic abb on natural numbers," in *Proceeding of 27th Symposium on Theoretical Aspects of Computer Science STACS*, Monterey, CA, 2010, pp. 597–608.

[65] D. Bresolin, D. Della Monica, A. Goranko V., Montanari, and G. Sciavicco, "Decidable and undecidable fragments of halpern and shoham's interval temporal logic: Towards a complete classification," *Proc. of the 15th Int. Conf. on Logic for Programming*, vol. 5330 of LNCS, Springer, pp. 590–604, 2008.

[66] V. Goronko, A. Montanari, and G. Sciavicco, "A road map of interval temporal logics and duration calculi," in *Journal of Applied Non-Classical Logics*, Monterey, CA, 2004(14), pp. 9–54.

[67] A. Lomuscio and J. Michaliszyn, "An epistemic halpern-shoham logic," in *Proceedings of IJCAI-2013*, 2013, pp. 1010–1016.

[68] Valdes-Perez, "Temporal data base management," *Artificial Intelligence*, vol. 32, pp. 1–55, 1987.

[69] R. Shostak, "Deciding linear inequalities by computing loop residues," *Journal of ACM*, vol. 28(4), pp. 769–779, 1981.

[70] C. Leierson and J. Saxe, "A mixed-integer linear programming problem which is efficiently solvable," *Proceedings 21st Annual Alerton Conference on Communications, Control, and Computing*, pp. 204–213, 1983.

[71] W. Armstrong, "Determinates of the utility function," *Economic Journal*, vol. 49, 1939.

[72] ——, "Uncertainty and the utility function," *Economic Journal*, vol. 58, pp. 1–10, 1949.

[73] F. Ramsey, "A mathematical theory and saving," *Economic Journal*, vol. 38, pp. 543–59, 1928.

[74] S. O. Hansson, "Preference logic," in *Handbook of Philosophical Logic*, ser. Handbook of Philosophical Logic, D. Gabbay and F. Guenthner, Eds. Springer Netherland, 2002, vol. 4, pp. 319–393.

[75] D. Davidson, "Hempel on explaining action," *Essays on Actiona and Events*, pp. 261–275, 1976.

[76] K. Jobczyk, B. M., A. Ligeza, and J. Karczmarczuk, "Comparative approach to the multi-valued logic construction for preferences," *Proceeding of ICAISC'15*, vol. LNAI, 2015.

[77] P. Hajek and D. Hramancova, "A qualitative fuzzy possibilistic logic," *International Journal of Approximate Reasoning*, vol. 12, pp. 1–19, 1995.

[78] L. Godo, P. Hajek, and F. Esteva, "Fuzzy logic and probability," *CoRR*, vol. abs/11302.4953, pp. 237–244, 2013.

[79] P. Hajek, *Metamathematics of Fuzzy Logic.* Dordrecht: Kluwer Academic Publishers, 1998.

[80] J. van Benthem and A. Gheerbrant, "Game: Solution, epistemic dynamics, and fixed-point logics," *Fundamenta Informaticae*, vol. 100, pp. 19–41, 2010.

[81] J. van Benthem, "Dynamic logic for belief revision," *Journal of Applied Non-Classical Logics*, vol. 17(2), pp. 119–155, 2007.

[82] J. von Wright, *The Logic of Preference.* Edinburgh, 1963.

[83] J. Y. Halpern, "Defining relative likehood in partially-ordered preferential structure," *Journal of Artificial Intelligence Research*, vol. 7, pp. 1–24, 1997.

[84] K. Venable and N. Yorke-Smith, "Simple temporal problems with preferences and uncertainty," in *Proceedings of AAAI,*, 2003, pp. 1385–1386.

[85] H.-J. Ohlbach, "Fuzzy time intervals and relations-the futire library," Inst. f. Informatik, LMU Munich, Research Report PMS-04/04, 2004.

[86] H. Ohlbach, "Relations between time intervals," in *11th Internal Symposium on TEMPORAL REPRESENTATION AND REASONING*, vol. 7, 2004, pp. 47–50.

[87] S. Shockert, M. Cock, and J. Kerre, "Efficient algorithms for fuzzy qualitative temporal reasoning," *IEEE Transition on fuzzy systems*, 2000.

[88] ——, "Imprecise temporal interval relations," *Proceedings of the 6th Intervnational Workshop on Fuzzy Logic and Application*, pp. 108–113, 2006.

[89] S. Shockert and M. Cock, "Temporal reasoning about fuzzy intervals," *Artificial Intelligence*, vol. 172(8-9), pp. 1158–1193, 2008.

[90] S. Shockert, M. Cock, and C. Cornelis, "Relatedness of fuzzy sets," *Journal of Intelligent and Fuzzy Systems*, vol. 16(4), pp. 297–303, 2005.

[91] S. Shockert, M. Cock, and E. Kerre, "Fuzzyfying allen's temporal interval relations," *IEEE Transiotion on Fuzzy System*, vol. 16(2), pp. 517–533, 2008.

[92] Y. Liao and C. Wong, "An algorithm to compact a vlsi symbolic layout with mixed constraints," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 2(2), pp. 62–69, 1983.

[93] G. Malik and T. Binford, "Reasonig in time and space," *Proceedings of IJCAI*, pp. 343–45, 1983.

[94] R. Jeffrey, *The Logic of Decision.* Edinburgh, 1971-89.

[95] C. Papadimitriou, "The euclidean traveling salesman problem is np-complete," *Theoretical Computer Science*, vol. 4(3), pp. 237–244, 1977.

[96] J.-P. Metivier, P. Boizumault, and S. Loudni, "Solving nurse rostering problems using soft global constraints." *In Proceedings of the 15th international conference on Principles and practice of constraint programming, CP09*, pp. 73–87, 2009.

[97] T. Lepegue, D. Prot, and O. Bellenguez-Morineau, "A tour scheduling problem with fixed jobs: use of constraint programming," *Practice and Theory of Automated Timetabling*, vol. August, pp. 29–31, 2012.

[98] W. Rudin, *Real and complex analysis.* McGraw-Hill, 1966.

[99] E. Hewitt and K. Stromberg, *Real and abstract analysis.* Stanford, Stanford University Press: Springer-Verlag, 1965.

[100] K. Jobczyk and A. Ligeza, "Strips in some temporal-preferential extension," *Proceedings of ICAISC2017 (FuzzIEEE)*, vol. 241-252, p. to appear, 2017.

[101] ——, "Multi-valued extension of putnam-davis procedure," *Proceedings of ICAISC2017 (FuzzIEEE)*, vol. 454-465, p. to appear, 2017.

[102] H.-J. Levesque, "A logic of implicit and explicit belief," *Proceedings of the 3rd National Conference on Artificial Intelligence(AAAI)*, pp. 198–202, 1984.

[103] U. Straccia, "A four-valued propositional logic," *Proceedings of IJCAI-1997*, pp. 128–133, 1997.

[104] K. Jobczyk, B. M., A. Ligeza, and J. Karczmarczuk, "Fuzzy integral logic expressible by convolutions," *Proceeding of ECAI'2014*, pp. 1042–1043, 2014.

[105] ——, "Fuzzy logic for representation of temporal verbs and adverbs 'often' and 'many times'," *Proceeding of LENSL'11 Tokyo*, 2014.

[106] K. Jobczyk and A. Ligeza, "Multi-valued halpern-shoham logic for temporal allen'srrelations and preferences," *Proceedings of the annual international conference of Fuzzy Systems (FuzzIEEE)*, p. to appear, 2016.

[107] ——, "Fuzzy-temporal approach to the handling of temporal interval relations and preferences," *Proceedings of INISTA*, vol. 61-8, p. to appear, 2015.

[108] J. Marcinkowski and J. Michaliszyn, "The last paper on the halpern-shoham interval temporal logic," 2010.

[109] A. Montanari, A. Murano, G. Perelli, and A. Peron, “Checking interval properties of computations,” *Proceeding of TIME*, pp. 59–68, 2014.

[110] D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco, “Interval temporal logics: a journay,” *Bulletin of EATCS*, vol. 105, pp. 73–99, 2011.

[111] B. Wozna and A. Lomuscio, “A logic for knowledge, correctness, and real time,” *CLIMA*, pp. 1–15, 2004.

[112] R. Fagin, J. Halpern, Y. Moses, and M. Vardi, *Reasoning about Knowledge.* Cambridge, 1995.

[113] K. Honda and N. Yoshida, “On reduction-based process semantics,” *Theoretical Computer Science*, vol. 152(1), pp. 437–86, 1995.

[114] D. Gabbay and V. Shehtman, “Product of modal logic, part 1,” *Logic Journal of the IGPL*, vol. 6(1), pp. 73–146, 1998.

[115] A. Kurucz, “Combining modal logic,” *Handbook of Modal Logic (eds.J. van Benthem, P. Blackburn, F. Wolter)*, vol. 3 (Elsevier), pp. 869–924, 2007.

[116] P. Kremer, “Strong completeness of s4 for any dense-in-itself metric space,” *The Review of Symbolic Logic*, vol. 6, pp. 545–570, 2013.

[117] L. Godo, H. Esteva, and R. Rodriquez, “A modal account of similarity-based reasoning,” *International Journal of Approximate Reasoning*, vol. 4, 1997.

[118] K. Jobczyk and A. Ligeza, “A general mathod of the hybrid controller construction for temporal planning with preferences,” *Proceedings of FedCSIS*, vol. 61-70, p. to appear, 2016.

[119] R. Buchi, “On a decision method in restricted second-order arithmetic,” *Stanford University Press*, 1962.

[120] M. Antonniotti and B. Mishra, “Discrete event models+ temporal logic = supervisory controller: Automatic synthesis of locomotion controllers,” *Proceedings of IEEE Intern. Conf. on Robotics and Automation*, 1999.

[121] F. Bacchus and F. Kabanza, “Using temporal logic to express search control knowledge for planning,” *Artificial Intelligence*, vol. 116, 2000.

[122] D. Hristu-Varsakelis, M. Egersted, and S. Krishnaprasad, “On the complexity of the motion description language mdle,” *Proceedings of the 42 IEEE Conference on Decision and Control*, vol. December, pp. 3360–3365, 2003.

[123] M. Vardi and P. Wolper, “An automata-theoretic approach to automatic program verification,” *Proceedings of the 1st Symposium on Logic in Computer Science*, vol. June, pp. 322–331, 1986.

[124] ——, “Reasoning about infinite computations,” *Information and Computation*, vol. 115(1), pp. 1–37, 1994.

[125] D. Giacomo and M. Vardi, “Automata-theoretic approach to planning for temporally extended goals,” *Proceeding of the 5th European Conference on Planning*, vol. 1809, pp. 226–238, 2000.

[126] G. Fainekos, H. Kress-gazit, and G. Pappas, “Hybrid controllers for path planning: A temporal logic approach,” *Proceeding of the IEEE International Conference on Decision and Control, Sevilla*, vol. December, pp. 4885–4890, 2005.

[127] ——, "Hybrid controllers for path planning: A temporal logic approach," in *Proceedings of the IEEE Int. Conference on Decision and Control*, vol. Sevilla, 2005, pp. 4885–90.

[128] L. Maximova, "Temporal logics with operator 'the next' do not have interpolation or beth property," in *Sibirskii Matematicheskii Zhurnal*, 32(6)1991, pp. 109–113.

[129] A. Montanari and P. Sala, "Interval logics and omega b-regular languages," *LATA*, vol. LNCS, pp. 431–444, 2013.

[130] M. Vardi and P. Wolper, "Reasoning about infinite computations," *Information and Computation*, vol. 115(1), pp. 1–37, 1994.

[131] K. Jobczyk, A. Ligeza, and K. Kluza, "Selected temporal logic systems: An attempt at engineering evaluation," *Proceedings of ICAISC'16*, vol. LNAI II, 2016.

[132] K. Jobczyk and A. Ligeza, "Systems of temporal logic for a use of engineering. toward a more practical approach," in *Intelligent Systems for Computer Modelling*, 2016, pp. 147–157.

[133] M. Mach-Krol, "Perspectives of using temporal logics for knowledge management," *Proceedings of FedCsIS*, vol. 49, pp. 935–938, 2012.

[134] ——, "Perspects of using temporal logics for knowledge management," *ABICT*, vol. 49, pp. 41–52, 2012.

[135] B. Wozna, A. Zbrzezny, and W. Penczek, "Checking reachability properties for timed automata via sat," *Fundam.Inform.*, vol. 55(2), pp. 223–241, 2003.

[136] B. Wozna and A. Zbrzezny, "Checking actl*-properties of discrete timed-automata via bounded model checking," *FORMATS*, pp. 18–33, 2003.

[137] A. Lomuscio, B. Wozna, and A. Zbrzezny, "Bounded model-checking real-time multi-agent systems with clock differences: Theory and implementation," *MoChArt*, pp. 95–112, 2006.

[138] G. Fainekos, H. Kress-gazit, and G. Pappas, "Temporal logic motion planning for mobile robots," in *Proceedings of the IEEE Int. Conference on Robotics and Automaton*, 2005, pp. 2032–37.

[139] G. Holzmann, *The spin model checker, primer and reference manual.* Addison-Wesley, 2004.

[140] A. Jourani, "Controllability and strong controllability of differential inclusions," *Nonlinear Analysis. Theory Methods and Applications*, vol. 75, pp. 1374–84, 2012.

[141] J. Pavelka, "On fuzzy logic i, ii, iii," in *Zeitschrift f. Math. Logik und Grundlagen der Math.*, vol. 25, 1979, pp. 45–52, 119–134, 447–464.

[142] J. Halpern and Y. Moses, "A guide to completeness and complexity for modal logics of knowledge and belief," *Artificial Intelligence*, vol. 54, pp. 319–379, 1992.

[143] W. van der Hoek and M. Wooldridge, "Model checking cooperation, knowledge, and time-a case study," *Research in Economics*, vol. 57(3), pp. 235–265, 2003.

[144] B. Moszkowski, "Reasoning about digital circuits," Ph.D. dissertation, Stanford University, 1983.

[145] R. Thomasson, "Combinations of tense and modality," *Handbook of Philosophical Logic*, vol. 2, pp. 135–165, Reidel.

[146] P. Dasgupta and S. Makherje, "A fuzzy real-time temporal logic," *Int. Journal of Approx. Reasoning*, vol. 54(9), pp. 1452–1470, 2013.

[147] A. Frigeri, L. Pasquale, and P. Spoletini, "Fuzzy time in linear temporal logic," *ACM Trans. Comput. Logic*, vol. 15(4) 30, pp. 1–30, 2014.

Summary

Temporal planning forms conceptually a part of temporal reasoning and it belongs to research area of Artificial Intelligence and it may be seen as an extension of classical planning by temporal aspects of acting. Temporal planing is usually complemented by considering preferences or different types of temporal constraints imposed on execution of actions.

There exist many approaches to this issue. One one hand, there are different paradigms to temporal planning, such as: planning via search in graphs (STRIPS), planning via satisfiability or planning in terms of Markov processes. These approaches are mutually incompatible. In addition, temporal planning requires a subject-specification as it is rather defined in a methodological way. On the other hand, temporal constraints are represented and modeled in different ways dependently on their quantitative or qualitative nature. In particular, Allen's relations between temporal intervals – an important class of temporal constraints – do not have any quantitative aspects and cannot be considered in computational contexts.

According to this situation, this PhD-thesis is aimed at the proposing a depth-analysis of temporal planning with fuzzy constraints which contains some remedies on these difficulties. Namely, two approaches to the representation and modeling of these issues are put forward.

In the first one (chapter 2, chapter 3) – fuzzy Allen's relations as fuzzy temporal constraints are represented by norms of convolutions in a Banach space of Lebesgue integrable functions. It allows us immerse Allen's relations in the computational contexts of temporal planning (based on STRIPS and on Davis-Putnam procedure) and to elucidate their quantitative nature. This approach is developed in a context of Multi-Agent Problem as a subject basis of this approach.

In the second one (chapter 4, chapter 5) – fuzzy temporal constrains with fuzziness introduced by preferences are represented in a logical terms of Preferential Halpern-Shoham Logic. It allows us to adopt these result in a construction of the plan controller. This approach is developed in a context of Temporal Traveling Salesman Problem as a subject basis of this approach.

Finally, an attempt to reconcile these two lines of representation of fuzzy temporal constraints was also proposed.

Streszczenie

Planowanie temporalne stanowi cześć rozumowania temporalnego i stanowi jeden z ważnych obszarów badawczych sztucznej inteligencji i może być widziane jako rozszerzenie planowania klasycznego przez temporalne aspekty działania. Planowanie temporalne jest zazwyczaj uzupełniane przez rozważenie różnych typów ograniczeń temporalnych nałożonych na akcje.

Istnieje wiele podejść do tego zagadnienia. Z jednej strony, istniej różne paradygmaty planowania, takie jak planowanie przez spełnianie lub przeszukiwanie grafu (z wykorzystaniem metody STRIPS) lub planowanie w terminach prosesów Markowa. Te podejścia sa wzajemnie niekompatybilne. Dodatkowo, planowanie temporalne wymaga jakiejś przedmiotowej specyfikacji i jest raczej definiowane w metodologiczny sposób.

Z drugiej strony, ograniczenia temporalne są reprezentowane i modelowane na różne sposóby zależnie od ich jakościowej lub ilościowej natury. W szczególności, relacje Allenowskie – ta ważna klasa ograniczeń temporalnych – nie posiada jakichś ilościowych aspektów i nie może być rozważana w obliczeniowych kontekstach.

Stosownie do tej sytuacji, niniejsza rozprawa jest ukierunkowana na pogłebioną analizę rozmytych ograniczeń temporalnych z preferencjami, która zawiera jakieś remedium na te trudności. Mianowicie, dwa podejścia do reprezentacji i modelowania tych zagadnień są zaproponowane.

W pierwszym z nich (rozdział 2, rozdział 3) – rozmyte relacje Allenowskie jako rozmyte ograniczenia temporalne sa reprezentowane przez normy z odpowiednich funkcji splotu w przestrzeni Banacha funkcji całkowalnych w sensie Lebesgue'a. Pozwala to zanurzyć relacje Allenowskie w obliczeniowych kontekstach planowania temporalnego (opartego na metodzie STRIPS i metodzie Putnama-Davisa) i uwypuklić ich jakościową naturę. To podejście jest rozwijane w kontekście Problemu Wielo-Agentowego jako przedmiotowa baza tego podejścia.

W drugim z podejść (rozdział 4, rozdział 5) – rozmyte ograniczenia temporalne z rozmytością wprowadzoną przez preferencje są reprezentowane w terminach logiki przez preferencjalną logikę Halperna-Shohama. Pozwala to zaadoptować te rezultaty w konstrukcji kontrolera planu. To podejście jest rozwijane w kontekście Czasowego Problemu Komiwojażera jako przedmiotowej bazy dla tego podejścia.

Ostatecznie, pewna próba uzgodniania tych dwóch linii reprezentacji ograniczeń temporalnych zostaĺa zaproponowana.