



Defeasible temporal logics for the specification and verification of exception-tolerant systems

Anasse Chafik

► To cite this version:

Anasse Chafik. Defeasible temporal logics for the specification and verification of exception-tolerant systems. Logic in Computer Science [cs.LO]. Université d'Artois, 2022. English. NNT: . tel-03984186

HAL Id: tel-03984186

<https://hal.science/tel-03984186>

Submitted on 12 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE DOCTORALE SCIENCES TECHNOLOGIES ET SANTÉ N°585

Defeasible temporal logics for the specification and verification of exception-tolerant systems

DOCTORAL THESIS

Publicly defended on 9 September 2022

in partial fulfillment of requirements for the degree of

Doctor of Philosophy of Artois University
in the subject of **Computer Science**

by

Anasse Chafik

Doctoral Committee

<i>President:</i>	Souhila Kaci	Université de Montpellier - France
<i>Reviewers:</i>	Laura Giordano Philippe Balbiani	Università del Piemonte Orientale - Italy CNRS, IRIT - France
<i>Examiners:</i>	Yannick Chevalier Yakoub Salhi	Université Toulouse 3 - France Université d'Artois - France
<i>Supervisors:</i>	Fahima Cheikh-Alili Jean-François Condotta Ivan Varzinczak	Université d'Artois - France Université d'Artois - France Université d'Artois - France

Acknowledgements

I would like to give my warmest thanks to my supervisors Fahima, Jean-François and Ivan. It is thanks to their effort that this work was made possible. I thoroughly enjoyed the discussion we had about this subject and I cherish all the personal advice they gave me over these last years. I would like to extend my thanks to Laura Giordano and Philippe Balbiani for their output that improved the quality of this memoir. I would like to show my appreciation to all the members of the jury for coming to the defence and help broaden my horizon.

My deepest thank you goes to my family. They helped me through the ups and downs of my personal and professional life. Their encouraging words meant the world to me. Thank you Baba, Mama, Mariem and Othmane. Finally, a special thanks goes to the to my friends and colleagues. Thank you all for the adventures and memories that we made along this journey. I wish you all the luck in the world.

"People assume that time is a strict progression of cause to effect, but actually from a non-linear, non-subjective viewpoint - it's more like a big ball of wibbly wobbly... time-y wimey... stuff."
-Doctor Who

Contents

List of Figures	ix
General Introduction	1

Part I Preliminaries

Chapter 1 Non-monotonic reasoning	7
1.1 The KLM approach to non-monotonic reasoning	8
1.1.1 Preamble	9
1.1.2 System P or preferential models	10
1.1.3 System R or rational closure	15
1.2 Use cases of logical formalisms extended with KLM conditionals	19
1.2.1 Description logic	19
1.2.2 Defeasible description logic	21
1.2.3 Modal logic	25
1.2.4 Defeasible modal logic	27
1.3 Summary	30
Chapter 2 Linear temporal logic	31
2.1 Linear temporal logic	33
2.1.1 Syntax and semantics	33
2.1.2 Axomatisation and properties of <i>LTL</i>	36
2.2 Satisfiability of <i>LTL</i> sentences	38
2.2.1 Observations and representations of temporal interpretations	38
2.2.2 <i>LTL</i> fragments	40
2.3 Semantic tableau for <i>LTL</i>	43
2.4 Summary	46

Part II Defeasible linear temporal logic

Chapter 3 Defeasible linear temporal logic	49
3.1 Introducing defeasible temporal operators	50

3.2	Preferential interpretations	51
3.3	Properties of defeasible temporal modalities	53
3.4	LTL^\sim sub-languages	55
3.4.1	The fragment \mathcal{L}_1	55
3.4.2	The fragment \mathcal{L}^\star	55
3.5	Summary	56
Chapter 4 The satisfiability problem of LTL^\sim sentences		57
4.1	State-dependent preferential interpretations	57
4.2	Useful representations of preferential structures	59
4.3	The bounded-model property of the fragment \mathcal{L}_1	62
4.4	The satisfiability problem in \mathcal{L}_1	65
4.5	The bounded-model property of the fragment \mathcal{L}^\star	68
4.6	The satisfiability problem in \mathcal{L}^\star	74
4.7	Summary	77
Chapter 5 A one-pass tableau for LTL^\sim		79
5.1	Tableau method for \mathcal{L}_1	80
5.1.1	Static rules	81
5.1.2	Dynamic rules	83
5.2	Soundness and completeness	85
5.2.1	Soundness	85
5.2.2	Completeness	89
5.3	Illustrations of tableaux for \mathcal{L}_1 sentences	94
5.4	Summary	97

Part III Conclusion

Appendices

Appendix A The satisfiability problem of LTL^{\sim} sentences	107
A.1 Proofs of results in Section 4.2	107
A.2 Proofs of results for Lemma 4.13	109
A.3 Proofs of results in Section 4.4	112
A.4 Proofs of results in Section 4.5	114
A.5 Proof of results in Section 4.6	118

Bibliography	122
---------------------	------------

List of Figures

1.1	Example of a preferential model	13
1.2	Example of a ranked model	18
2.1	Time structures	32
2.2	Graphical representation of <i>LTL</i> sentences	34
2.3	<i>LTL</i> interpretation	36
2.4	Ultimately-periodic interpretation	40
2.5	Dynamic rules	46
3.1	Preferential temporal interpretation $I = (V, \prec)$	52
4.1	Preferential temporal interpretation $I = (V, \prec)$	58
4.2	Induced acceptable sequence	69
4.3	Selected time points of α in $AS(I, N)$	70
4.4	<i>anchors</i> for \Diamond -sentences	71
4.5	<i>anchors</i> for \Diamond -sentences	72
4.6	<i>anchors</i> for \Box -sentences	72
5.1	Illustration of the branch B	85
5.2	Tableau for $\Box(\neg p \vee q) \wedge \Diamond p$	95
5.3	Tableau for $\Box p \wedge \Diamond \neg p$	96
5.4	Tableau for $\Diamond p \wedge \Diamond q$	98

General Introduction

Logic formalizes different modes of reasoning. More precisely, it is a science of investigating how *conclusions* are inferred from a set of *arguments* or *premises*. The study of logic begun with the ancient Greek who started to use natural language to perform reasoning. The most known Aristotelien logic is reasoning in the form of *syllogisms*. Here is an example of a *syllogism*:

Premise: Magpies fly.

Premise: Tweety is a magpie.

Conclusion: Tweety flies.

Knowing that both premises “Magpies fly” and “Tweety is a magpie” are true, we can infer that the conclusion “Tweety flies” is also true. These forms of reasoning have been formalized further and developed into classical logic in the modern era. It consists of propositional and first-order logic. Classical logic, in a sense, is based on fundamental logical intuitions and rules that are agreed upon by logicians. Alongside these rules, a vocabulary is developed using symbols to simplify the expression of premises and conclusions. In artificial intelligence, classical logic as well as many of its extensions (modal, epistemic, description, etc.) are used as formal tools for representing knowledge and reasoning with said knowledge in order to extract new conclusions. In the latter half of the twentieth century, the role of mathematical logic in computer science is to codify problems in a language that machines can interpret and define inference rules based on the different modes of reasoning.

One of these modes of reasoning is called *defeasible inferences*. When using defeasible reasoning, one can draw conclusions based on generic or incomplete information, all whilst keeping the right to retract these conclusions when presented with further information. Consider the following premises:

Premise: *Normally*, birds fly.

Premise: Penguins are birds.

Premise: Penguins do not fly.

Now, given a new premise “Tweety is a bird”, should we infer that it can fly or not? since the only knowledge that we have on Tweety is that it is a bird, we can plausibly infer that “Tweety flies”. However, when presented with the additional premise “Tweety is a penguin”, and knowing that penguins are one of the rare birds that cannot fly, the conclusion is no longer true. The conclusion “Tweety flies” is thereby retracted.

It turns out that this retractable way of reasoning is more similar to the common sense reasoning than the deductive reasoning in classical logic. Classical logic operates on the basis that all current information must be true and complete (without any edge cases, such as the penguins on the previous syllogism). When presented with complete information, all drawn conclusions are not challenged and

any additional information to the set of premises does not falsify them. This property of deductive reasoning is called *monotony*. However, premises such as “normally, birds fly” are not complete. Since we cannot say for sure that all birds fly, we cannot derive conclusions from this premise using deductive reasoning. On the other hand, defeasible reasoning *breaks* the monotony rule by allowing reasoners to draw initial conclusions even when the set of premises is not complete and granting the ability to retract said conclusions when needed. That is why defeasible reasoning belongs to the family of *non-monotonic logics*.

The field of non-monotonic logic has been a topic of interest in philosophy in the recent years. As defeasible reasoning provides a solid framework of reasoning when dealing with incomplete information that might contain exceptional cases, many researchers in artificial intelligence took interest in non-monotonic reasoning as well. The work was pioneered by McCarthy and Hayes [MH69] who discovered the need to represent and implement this sort of reasoning. McCarthy and Hayes [MH69] introduced the *frame problem* which is the challenge of representing the effects of an action without explicitly mentioning a number of intuitively obvious non-effects. This reveals the question of whether it is possible to infer conclusions based only on what is relevant to the current situation, whilst tolerating potential anomalies that might arise in the cases that are not as relevant. Later on, McCarthy [McC80] proposed the principle of *circumscription* as a solution to this problem. In this setting, one considers that we encounter abnormalities in any situation, and it is warranted to conclude whatever is true based on all *preferred* outcomes (which are considered to be minimal in regards to the other outcomes). Several systems of non-monotonic reasoning in classical propositional logic build upon this notion of preference as an ordering relation, amongst them are the work of Bossu and Siegel [BS85], Shoham [Sho87], Geffner [Gef89] and Kraus et al. [KLM90].

We are particularly interested in the seminal work of Kraus et al. [KLM90] (known as the KLM approach) on conditionals of the form “normally, if α is true, then β is true” and denoted by $\alpha \sim \beta$. Kraus et al. generalized patterns of non-monotonic reasoning and defined clearly the set of rules for defeasible inferences. Based on the aforementioned rules that are set for the inference operator \sim , Kraus et al. introduced multiple families of non-monotonic inferences and defined adequate preference relations of models for each of these families. The preference relation is an ordering relation on the worlds which compares them based on their normality and plausibility in general. The preference relation is often denoted by the symbol \prec . In recent years, defeasible aspects have been investigated in more complex logics thanks to the KLM approach. The notable ones are defeasible extensions for description logic proposed by Giordano et al. [GGOP07, GOGP09], Britz et al. [BCM⁺20] and a defeasible extension for modal logic proposed by Britz et al. [BV18]. This work constitutes the continuation of this investigation on other forms of logic.

The second topic of interest is *temporal logic*. Temporal logics cover approaches to reasoning about time. The first temporal logic was introduced by Prior [Pri62] and was called *tense logic*. His idea is inspired by the use of tense in natural language to differentiate between past, present and future. Tense logic introduces temporal operators such as *always* and *eventually* to the language, which are given a syntax and semantics similar to modalities in modal logic. This approach has the advantage of expressing propositions that vary over time using an intuitive language rather than expressing them in first-order logic. Subsequent temporal logics have been developed by logicians over the years. Three prominent temporal logics are the linear temporal logic *LTL* [Pnu77], computational tree logic *CTL* (also known as branching time logic) [EC80] and *CTL** [EH83] which combines both *LTL* and *CTL*. Applications of temporal logic range from its use in philosophy as a framework to define temporal expressions in natural language, to formalisms for encoding temporal knowledge in artificial intelligence. They are mostly used as tools for the specification and verification of executions in computer programs and systems.

The main goal of this work is to investigate aspects of defeasibility in temporal logics. When taking into account the frame problem by McCarthy [MH69], computer systems are neither 100% safe nor

100% faulty. Systems might contain, at some points of time, innocuous cases where exceptions are to be tolerated, or on the contrary, exceptions that must be carefully handled on other points of time that are more relevant. Similarly, the expected behavior of a system may not be correct for every possible execution, but rather for its most “normal” or probable executions. It so happens that temporal logics, because they are logical formalisms built upon the basis of classical logic, i.e., whose underlying reasoning is deductive and not common sense (defeasible), do not allow at all to formalize the different nuances of exceptions and even less to treat them. Technically the reason being that these logics (1) at the level of propositions, are often based on two truth values (true or false), (2) at the level of the object language, have operators behaving in a monotonic way, and (3) at the level of reasoning, have a notion of logical consequence which is also monotonic and, consequently, is not adapted to the evolution of the deductible facts.

We present in this memoir a temporal formalism called defeasible linear temporal logic, denoted by LTL^\sim . In this logic, we investigate the integration of preferential semantics of the KLM approach in LTL . Based on Britz and Varzinczak’s work [BV18], we introduce defeasible temporal operators that can help formalize temporal expressions that implicitly take into account that the execution might be exceptional at some points of time. The intuition here is to provide a framework for expressing defeasible properties on systems, all while following the principle of Prior’s temporal operators in spirit. In other words, classical LTL encodes sentences such as “it will *always* ...”, and LTL^\sim can encode sentences of the form “*normally*, it will *always* ...”.

Synopsis of the memoir

This memoir is split into two parts: the preliminaries and the contributions. In the preliminaries, we set up the building blocks of the formalism LTL^\sim :

- Chapter 1 highlights the KLM approach [KLM90, LM89] to non-monotonic logic. We present their work on an axiomatic characterization of defeasible inference relations alongside their proposed semantics for each of these relations. We present two of the prominent families of relations, which are the preferential and rational inferences. In the latter half of the chapter, we expose use cases of formalisms other than propositional logic extended with KLM approach, such as description logic [GGOP07, GOGP09, BCM⁺20] and modal logic [BV18].
- Chapter 2 goes into linear temporal logic LTL . We define the syntax and semantics of this logic that we shall use in this memoir. Next, we highlight the axioms and properties of temporal operators from a modal logic standpoint. We present the work of Sistla and Clarke [SC85] on satisfiability of LTL sentences and finally the version of tableau proposed by Reynolds [Rey16a].

The second part constitutes the contribution part. We give our report of the findings for LTL^\sim :

- Chapter 3 introduces defeasible linear temporal logic LTL^\sim . This formalism enriches LTL ’s temporal expression by adding defeasible temporal operators. We present the motivation, syntax and semantics for LTL^\sim . We also compare the properties of the newly added modalities with their classical counterparts.
- Chapter 4 is the extension of the work done regarding the satisfiability of LTL^\sim sentences [CCACV20]. We discuss the decidability of this problem for two fragments of the language, namely \mathcal{L}_1 and \mathcal{L}^* . We gave a detailed analysis to obtain the bounded-model property, as well as provide the procedure for checking the satisfiability in said fragments.

- Chapter 5 concerns the topic of tableau for LTL^\sim [CCACV21]. We adapt Reynolds' one-pass tableau [Rey16a] by integrating the preferential semantics into the method. We propose a tableau for one of the previously mentioned fragment. We provide a sound and a complete method that can serve as the basis for further exploring tableaux for this logic.

The conclusion summarizes the memoir, and provides perspectives for future work. We discuss the findings about the introduction of defeasible inferences *à la* KLM to LTL^\sim . We give our intuition for a defeasible extension for CTL as well.

Part I

Preliminaries

Chapter 1

Non-monotonic reasoning

Contents

1.1 The KLM approach to non-monotonic reasoning	8
1.1.1 Preamble	9
1.1.2 System P or preferential models	10
1.1.3 System R or rational closure	15
1.2 Use cases of logical formalisms extended with KLM conditionals	19
1.2.1 Description logic	19
1.2.2 Defeasible description logic	21
1.2.3 Modal logic	25
1.2.4 Defeasible modal logic	27
1.3 Summary	30

Non-monotonic reasoning (NMR) covers a family of formalisms and logics that capture and represent *defeasible inference*. Using defeasible inference, reasoners draw conclusions even when the information is incomplete and they reserve the right to retract said conclusion in the light of further information. This deductive type of reasoning tries to represent a mode of reasoning closer to the common sense, it is used in philosophical fields and expert fields (e.g. suspects list during an investigation, medical diagnoses ...). However, classical logic (ranging from propositional to more complex formalisms like modal and description logic) fails to capture this aspect of *defeasibility* in inferences. Classical (or monotonic) inferences are by nature based on complete information and thus do not allow for the retraction of inferences.

Defeasible reasoning is dynamic in that it allows for a retraction of inferences. A famous introductory example in the literature of defeasible reasoning and non-monotonic reasoning is Tweety the bird. When presented with the background knowledge that “birds usually fly”, we can infer that Tweety the bird can fly. The latter inference can be retracted when presented with the new information that Tweety is a penguin (penguins are an *exceptional* case of birds that cannot fly).

This kind of dynamics is characteristic for non-monotonic reasoning. When trying to express the birds and penguins problem using classical consequence (that of propositional logic), the statement “birds fly” entails that all the birds are able to fly, without any exception. Therefore in this case, when presented with the information that Tweety is a penguin and penguin are birds, the reasoner safely assumes that it can fly. This property in classical logic is referred to as the monotony rule. We shall use \vdash to represent classical consequences:

(Monotony) If $\Sigma \vdash \alpha$, then $\Sigma \cup \Sigma' \vdash \alpha$.

Monotony states that consequence are robust to the addition of new information. If a sentence α is a consequence of a set of premises Σ , then it is also a consequence of any set of premises that contains Σ as a subset. Hence, any consequence entailed using classical logic is not called in question and therefore not retractable. This is why sometimes the monotony property is not desired when going for the retractability in defeasible inferences. Most forms of defeasible reasoning violate the rule of monotony. This is why the field of defeasible reasoning is called non-monotonic. It is a study of formal reasoning systems on which the monotony rule is not present and is replaced by other properties.

So what are the properties of inferences in NMR settings? Given that monotony is abandoned in NMR systems, we are led to the question of which properties that can replace it in order for inferences to be considered defeasible. The central properties in the literature are (defeasible inferences are represented by \vdash):

- (Reflexivity) If $\alpha \in \Sigma$, then $\Sigma \vdash \alpha$.
- (Cut) If $\Sigma \vdash \alpha$ and $\Sigma \cup \{\alpha\} \vdash \beta$, then $\Sigma \vdash \beta$.
- (Cautious Monotony) If $\Sigma \vdash \alpha$ and $\Sigma \vdash \beta$, then $\Sigma \cup \{\alpha\} \vdash \beta$.

Cautious monotony is a weaker form of the monotony property which allows the addition of a new information α as a defeasible premises of β , i.e., Σ , only when α is itself is a defeasible consequence of Σ . Thanks to these principles, a reasoner is able (under a set of restrictions) to draw conclusions that can in turn be used as additional premises to the set of defeasible conclusions. In the work of Gabbay [Gab85], it has been shown that some basic intuitions about non-monotonic derivations of the classical consequence gives rise to defeasible inferences that satisfy Reflexivity, Cut and Cautious monotony. Which is why these properties are considered, by many scholars (Makinson [Mak05], Gabbay [Gab85], Adams [Ada65], Pearl [Pea89] and Geffner [Gef89]), to be the central principles of NMR.

This gave rise to many formalisms that pioneered the work in NMR. Such formalisms have a goal of giving a mathematical characterization of defeasible inferences in reasoning systems. Here is some examples of NMR formalisms: Default logic (Reiter [Rei80], Besnard [Bes89]), Circumscription logic (McCarthy [McC80], Moinard [Moi88]), autoepistemic logic [Moo93] and conditional logic (Kraus et al. [KLM90, LM89], Lewis [Lew73], Pearl [Pea90]) amongst others.

The selling point of non-monotonic reasoning, in the study highlighted in this memoir, is its ability to cope with the notion of exceptionality. Defeasible reasoning by nature is retractable. An agent using defeasible inferences such as “Normally, if α , then β ” can infer that in most plausible, normal or in general preferred cases where α is true, then β is true. All whilst keeping in mind that there might be exceptional cases of α where β can be false. Assertions of the form “Normally, if α , then β ” are referred to as *conditionals*. The work we highlight here is the Kraus, Lehmann and Magidor’s (or KLM) approach [KLM90, LM89] to conditionals. We discuss the properties of defeasible inferences in their approach. Moreover, we expose also other formalisms that extend the preferential approach and express other forms of defeasibility.

This chapter is divided in two sections, we shall discuss different properties of conditionals and their appropriate semantics in Section 1.1. Section 1.2 contains a study of two logics (other than propositional) on which KLM style of semantics have been investigated.

1.1 The KLM approach to non-monotonic reasoning

Since the introduction of Reiter’s work [Rei80] on default logic, many logician and researchers defined logical frameworks which are able to represent and manage conditional type assertions. A conditional

is any assertion of the form “Normally, if α , then β ”. In the aforementioned statement, the normal (we can also say preferred or plausible) worlds that satisfy α are worlds of β . The less normal (exceptional) worlds of α do not satisfy necessarily β . Conditionals diverge from classical assertions, of the form “if α , then β ”, in the sense that all α -worlds need to satisfy β , no matter how unlikely they are. Whilst a conditional statement captures the essence of defeasibility by allowing the unlikely worlds of α to not satisfy β .

In the field of conditionals and conditional logic, many authors developed an array of systems and formalisms in order to formalize conditionals. Nevertheless, these approaches share a common feature, syntax-wise, of introducing a binary inference operator \sim in order to express conditionals. A conditional is a statement of the form $\alpha \sim \beta$ which indicates that generally, α -worlds are β worlds. Many studies in this field were focused on exploring properties and postulates of the inference relation \sim in order to be considered non-monotonic. We cite the work of Gabbay [Gab85], Makinson [Mak05] and Kraus et al. [KLM90, LM89]. This field had also a surge of a multitude of systems that define conditional inferences, some use the preferential semantics [Sho87, BS85] Lewis’s approach [Lew73] or the possibilistic approach [Pea89, BDP97].

In this section, we present the work of Kraus, Lehmann and Magidor [KLM90, LM89], formally known as the KLM approach, on conditionals. In their setting, the KLM approach defines the set of properties for the relation \sim for different systems. The approach is based on the preferential approach of Shoham [Sho87]. Briefly, Shoham suggested that models can be described as a set of worlds equipped with a *preference* relation \prec which is an ordering relation on these worlds. Using the preference relation \prec , a world w is more preferred than v if w is considered to be more plausible, normal than v . A conditional statement $\alpha \sim \beta$ would then, in the model, mean that the sentence β is true for all worlds that satisfy α and are most normal among worlds of α with respect to the preference relation \prec . Kraus et al. investigate this concept further. The KLM approach defines three central systems, namely cumulative **C**, preferential **P** systems and rational closure **R**. In each of these systems, they described the properties of the inference relation \sim and proposed the adequate model that is well suited for \sim .

Since we will be using the preferential models in our work. We describe in what follows the system **P** and the system **R**, since they are considered to be the core of the KLM approach. We present the properties of the \sim relation and their models for each of these systems.

1.1.1 Preamble

Before discussing the KLM approach, we set up the notions and notations. The set of propositional atoms is represented by $\mathcal{P} \stackrel{\text{def}}{=} \{p, q, r, s, t, \dots\}$ (atoms are represented by a lower case letter). We use the standard Boolean connectors: \neg for negation, \wedge for conjunction, \vee for disjunction, \rightarrow for consequence and \leftrightarrow for equivalence. For any given set of propositional atoms \mathcal{P} , let \mathcal{L} be the set of well-formed sentences of propositional logic that can be formed using \mathcal{P} . We shall use small Greek letters $\alpha, \beta, \gamma, \dots$ to denote sentences in \mathcal{L} .

Classical consequence are represented by the Boolean operator \rightarrow . A statement of the $\alpha \rightarrow \beta$ reads as: if α , then β . Non-monotonic inferences on the other hand, are represented by a binary relation \sim . A statement of the form $\alpha \sim \beta$ reads as: normally, if α then β . Using the language opted by researchers in this field of study, monotonic implications are referred to as consequences. While, non-monotonic implications are referred to as inferences. We shall assume these terms to differentiate between them. We discuss the properties of the \sim operators for each of the systems in the upcoming section.

For semantics, the notions of a *world* u refers to a possible assignment of truth values of the given propositional atoms in \mathcal{P} . The set \mathcal{U} denotes the set of all possible worlds and represents a *universe of reference*. The set \mathcal{U} is a subset of all assignments of truth values of the propositional atoms in \mathcal{P} .

We assume that the notion of satisfaction behaves as the standard as far as Boolean connectors are

concerned. Let $u \in \mathcal{U}$, $p \in \mathcal{P}$, $\alpha, \beta \in \mathcal{L}$, the notation $u \models \alpha$ indicates that the world u is a world where α is true. The truth values of \mathcal{L} sentences are recursively defined as follows:

- $u \models p$ if the atom p is assigned to true in u ;
- $u \models \neg \alpha$ if $u \not\models \alpha$ (we use the notation $\not\models$ to indicate that α is false in u);
- $u \models \alpha \wedge \beta$ if $u \models \alpha$ and $u \models \beta$;
- $u \models \alpha \vee \beta$ if $u \models \alpha$ or $u \models \beta$;
- $u \models \alpha \rightarrow \beta$ if $u \models \neg \alpha$ or $u \models \beta$;
- $u \models \alpha \leftrightarrow \beta$ if $u \models \alpha \rightarrow \beta$ and $u \models \beta \rightarrow \alpha$.

We use the classical symbols $\top \stackrel{\text{def}}{=} p \vee \neg p$ to indicate true and $\perp \stackrel{\text{def}}{=} p \wedge \neg p$ to indicate false. We shall say that the world u satisfies the sentences α (or u is an α -world) if $u \models \alpha$. We assume that a sentence is valid if for all $u \in \mathcal{U}$, we have $u \models \alpha$. A valid sentence α is denoted by $\models \alpha$. The last notation we shall set is $\llbracket \alpha \rrbracket$. Let α be a sentence, if a world u is an α -world then $u \in \llbracket \alpha \rrbracket$. In other words, $\llbracket \alpha \rrbracket$ denotes the set of all worlds in the universe of reference \mathcal{U} that satisfy the sentence α .

1.1.2 System P or preferential models

The central system in the KLM approach to non-monotonic reasoning is the preferential system, known as the system **P**. Before discussing this non-monotonic system, we shall recall the monotonic property for the consequence operator in classical systems. Let $\alpha, \beta, \gamma \in \mathcal{L}$, the monotony property states:

$$(\text{Monotony}) \text{ If } \models \alpha \rightarrow \gamma, \text{ then } \models \alpha \wedge \beta \rightarrow \gamma$$

Using classical consequence, the assertion $\alpha \rightarrow \gamma$ is never challenged. This is based on the premise all α -worlds are γ -worlds, without any exception. Therefore, we can deduce new facts based upon this assertion, i.e., $\alpha \wedge \beta \rightarrow \gamma$ (which are, also, never challenged). However, conditionals such as “normally, if α , then β ” are inference which are defeasible in nature. When handling conditionals, the reasoner have the right to retract these inferences when presented with new facts. Moreover, the reasoner needs to be *cautious* when trying to deduce new facts based upon conditionals. Let see the birds and penguins example, we have the following assertions: “penguins are birds” $penguins \rightarrow birds$ and “normally, birds fly” $birds \sim flies$. If Tweety is a bird, we might plausibly deduce that it can fly. When Tweety is a penguin on the other hand, we do not have the inference of penguins being capable of flying, i.e., we do not know whether $penguins \sim flies$ is true. Assuming that inferences behave monotonically (same as a classical consequence), $\models penguins \rightarrow birds$ and $\models birds \sim flies$ entails that $\models penguins \wedge birds \sim flies$. Therefore, Tweety can indeed fly. However, the inference $penguins \wedge birds \sim flies$ does not take in account whether penguins are a special case of birds that do not know how to fly (which is the case in real life). The monotony property may be undesirable in non-monotonic inferences.

Defining properties of non-monotonic inference relation \sim has been the study of many researchers in the NMR field. In the preferential models (or system **P**), Kraus et al. proposed the axioms mentioned below. Let $\alpha, \beta \in \mathcal{L}$:

1. Reflexivity

$$\alpha \sim \alpha \tag{1.1}$$

A sentence α is a plausible inference of itself.

2. **Left logical equivalence**

$$\frac{\models \alpha \leftrightarrow \beta, \quad \alpha \sim \gamma}{\beta \sim \gamma} \quad (1.2)$$

Left logical equivalence expresses that equivalent sentences have the same plausible inferences.

3. **Right weakening**

$$\frac{\models \alpha \rightarrow \beta, \quad \gamma \sim \alpha}{\gamma \sim \beta} \quad (1.3)$$

Right weakening states that we may replace the plausible inference α of γ by β when β is a consequence of α .

4. **Cut**

$$\frac{\alpha \wedge \beta \sim \gamma, \quad \alpha \sim \beta}{\alpha \sim \gamma} \quad (1.4)$$

Cut expresses the fact that, in order to deduce a new conditional $\alpha \sim \gamma$ from $\alpha \wedge \beta \sim \gamma$, we need to have $\alpha \sim \beta$.

5. **Cautious monotony**

$$\frac{\alpha \sim \beta, \quad \alpha \sim \gamma}{\alpha \wedge \beta \sim \gamma} \quad (1.5)$$

Cautious monotony is a restrictive form of the monotony property. In the way that, we can deduce $\alpha \wedge \beta \sim \gamma$ from $\alpha \sim \gamma$, only when β is also plausibly inferred from α . This is considered as the primordial rule for many non-monotonic systems (Adams [Ada65], Pearl [Pea89], Geffner [Gef89]).

6. **OR**

$$\frac{\alpha \sim \gamma, \quad \beta \sim \gamma}{\alpha \vee \beta \sim \gamma} \quad (1.6)$$

The OR rule states that if γ can be plausibly inferred from both α and β , then γ can be plausibly inferred from their disjunction $\alpha \vee \beta$.

An interesting observation to highlight is that when using the axioms of reflexivity and right weakening, we have the property:

$$\text{if } \models \alpha \rightarrow \beta, \text{ then } \alpha \sim \beta.$$

Since we have $\models \alpha \rightarrow \beta$, and know that $\alpha \sim \alpha$ (thanks to reflexivity), we obtain $\alpha \sim \beta$ using right weakening. This observation concurs with the nature of conditionals. In the sense that, since there are no exceptional worlds of α that do not satisfy β , we can safely assume that $\alpha \sim \beta$.

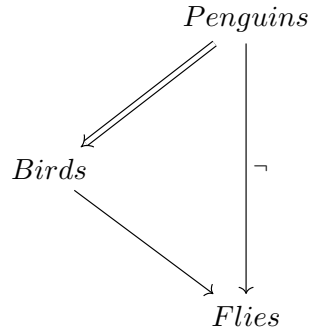
Note that the axioms from 1.1 to 1.5 are all rules of the cumulative system (or system **C**) in the KLM approach. The conjunction of cut and cautious monotony can be expressed together by the following principle: if $\alpha \sim \beta$, then the plausible inferences of α and $\alpha \wedge \beta$ are the same, i.e., $\alpha \sim \gamma$ and $\alpha \wedge \beta \sim \gamma$. This principle is called *cumulativity* (Makinson [Mak05]). The system **P** is an extension of system **C** with the addition of the OR rule.

For a concrete example on the purpose of cautious monotony, we go back to the example of birds and penguins.

Example 1.1 (Birds and penguins). We have the following assertions: “penguins are birds” and “normally, birds fly”. If we express these assertions using consequence and non-monotonic inference, we obtain $\text{penguin} \rightarrow \text{bird}$ and $\text{bird} \sim \text{flies}$. We would like to know whether we can infer $\text{penguin} \sim \text{flies}$ or not.

Using Reflexivity (1.1) and Right weakening (1.3), we have the assertion $\text{penguin} \sim \text{bird}$. Even if $\text{bird} \sim \text{flies}$ holds, we cannot infer that $\text{bird} \wedge \text{penguin} \sim \text{flies}$ because we do not have $\text{bird} \sim \text{penguin}$. Since birds are not a premise of penguins,

Let us introduce $\text{penguin} \sim \neg \text{flies}$ next. Using Cautious monotony (1.5), from $\text{penguin} \sim \text{bird}$ and $\text{penguin} \sim \neg \text{flies}$, we can safely infer that $\text{penguin} \wedge \text{bird} \sim \neg \text{flies}$. Expressing this inference using natural language, birds that are penguins usually do not fly.



Given a set of starting conditionals Δ , a *preferential inference* relation between two sentences α, β is defined as:

Definition 1.2 (Preferential inference relation). Let $\alpha, \beta \in \mathcal{L}$, and Δ be a set of conditionals. An inference relation $\alpha \sim \beta$ is said to be preferential if $\alpha \sim \beta$ can be deduced from Δ using only rules of system **P**.

We shall discuss the semantics proposed by Kraus et al. for system **P**. Let \mathcal{P} be the set of atomic propositions. Each preferential model consists of a non-empty set of states S (each state $s \in S$ is a world in the universe of reference \mathcal{U}) and a binary ordering relation \prec on these states. The relation \prec represents the preference that are between the states. Before defining formally preferential models, we define the ordering relation \prec first.

Definition 1.3 (Asymmetry). Let \prec be a binary relation on a set U . The relation \prec is asymmetric if for all elements $s, t \in U$ such that $s \prec t$, we do not have $t \prec s$ (noted $t \not\prec s$).

Definition 1.4 (Irreflexivity). Let \prec be a binary relation on U . The relation \prec is irreflexive if for all elements $s \in U$ $s \not\prec s$.

Definition 1.5 (Transitivity). Let \prec be a binary relation on U . The relation \prec is transitive if for all $s, t, u \in U$ if $s \prec t$ and $t \prec u$ then $s \prec u$.

Definition 1.6 (Strict partial order). Let \prec be a binary relation on U . \prec is a strict partial order if \prec is asymmetric, irreflexive and transitive.

Definition 1.7 (Minimal). Let $V \subseteq U$ and $s \in V$, a state s is minimal in V if there is no $t \in V$ such that $t \prec s$.

The set $\min_{\prec}(V)$ denotes the set of minimal elements of V with respect to the ordering relation \prec .

Definition 1.8 (Smooth). Let $V \subseteq U$ and \prec be a strict partial order on U . We say that V is smooth if for all the element $t \in V$, either there is an element $s \in V$ such that s is minimal in V and $s \prec t$ or t is minimal in V .

We define the preference relation \prec as a strict partial order on the set of states S . Next, we shall define preferential models of system **P**.

Definition 1.9 (Preferential models). Let \mathcal{P} be a set of atomic propositions, and \mathcal{U} be a universe of reference. A preferential model $\mathcal{P} \stackrel{\text{def}}{=} (S, l, \prec)$ is a tuple where S is a set, elements of which are called states, $l : S \rightarrow \mathcal{U}$ assigns to each state s a single world $u \in \mathcal{U}$. The relation \prec is a strict partial order of the states of S , the relation \prec satisfies the smoothness condition (see Definition 1.12).

With the introduction of the ordering relation \prec , states in S can be compared amongst themselves. Let $s, t \in S$, if $s \prec t$, then s is more normal, plausible or in general preferred than t . Next, we define the notion of satisfiability in preferential models.

Definition 1.10 (Satisfiability in preferential models). Let $\mathcal{P} := (S, l, \prec)$ be a preferential model, and $\alpha \in \mathcal{L}$. We say that a state $s \in S$ satisfies α (denoted by $s \models \alpha$) if its corresponding world $l(s)$ satisfies α , i.e., $l(s) \models \alpha$.

It is worth to point out that each state $s \in S$ is mapped to one and only one world $u \in \mathcal{U}$ thanks to the mapping function l . The notion of satisfiability of \mathcal{L} -sentences in preferential models stays the same as discussed in the preamble (see Section 1.1.1). We say that a preferential model $\mathcal{P} = (S, l, \prec)$ satisfies a sentence α (denoted by $\mathcal{P} \models \alpha$) if $s \models \alpha$ for all $s \in S$.

Definition 1.11 (α -states). Let $\mathcal{P} := (S, l, \prec)$ be a preferential model, and $\alpha \in \mathcal{L}$. The set of states that satisfy α is defined as follows: $\llbracket \alpha \rrbracket^{\mathcal{P}} \stackrel{\text{def}}{=} \{s \mid s \in S, s \models \alpha\}$.

Thanks to the ordering relation \prec , states that satisfies a given sentence α can be compared between each other. They can be ordered from the least normal states of α to the more plausible ones. Given an $\alpha \in \mathcal{L}$ and a preferential model $\mathcal{P} = (S, l, \prec)$, the set $\min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{P}})$ denotes the set of preferred states with respect to \prec . The addition of the smoothness condition ensures that the smoothness $\llbracket \alpha \rrbracket^{\mathcal{P}}$ for all well formed sentences of \mathcal{L} with respect to \prec .

Definition 1.12 (Smoothness condition). A preferential model $\mathcal{P} := (S, l, \prec)$ satisfies the smoothness condition if for all $\alpha \in \mathcal{L}$, the set $\llbracket \alpha \rrbracket^{\mathcal{P}}$ is smooth.

Another way to look at the smoothness condition is that for all $\alpha \in \mathcal{L}$ such that $\llbracket \alpha \rrbracket^{\mathcal{P}}$ is not empty, the set $\min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{P}})$ is not empty as well. With smoothness condition, we are sure to find a set of preferred states of any well formed sentence in \mathcal{L} with respect to \prec .

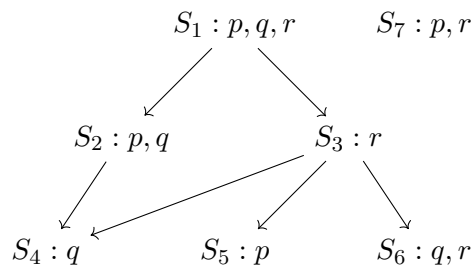


Figure 1.1: Example of a preferential model

Figure 1.1 shows an example of a preferential model $\mathcal{P} = (S, l, \prec)$. Each S_i is a state in S that has a mapping to a world $l(S_i)$. The arrows indicate the preferential ordering of these states. For instance, $S_1 \rightarrow S_2$ means that S_1 is more preferred than S_2 with respect to \prec . The transitive closure is not shown in Figure 1.1. Based on the transitive properties of \prec , from $S_1 \rightarrow S_3$ and $S_3 \rightarrow S_6$, we deduce that $S_1 \rightarrow S_6$. It is worth to point out that the ordering relation \prec is a strict partial order. It is possible to have states that are not compared to others. In this example, the state S_7 fits this criteria. The state S_7 , in this case, is not less preferred than the other states of S .

Kraus et al. also defined the non-monotonic inference $\vdash_{\mathcal{P}}$ for preferential models. Recall that a conditional statement of the form $\alpha \vdash \beta$ refers to all normal worlds of α are worlds of β . Thanks to the introduction of the preference relation \prec , it is possible to order α -states from the more plausible states of α to the least plausible ones. As discussed earlier, the $\min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{P}})$ denotes the set of all minimal states of α with respect to \prec . In preferential models setting, an inference $\alpha \vdash_{\mathcal{P}} \beta$ is true if all states in $\min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{P}})$ are β -states. Formally, the non-monotonic inference $\vdash_{\mathcal{P}}$ for preferential models is defined as follows:

Definition 1.13. Let $\mathcal{P} = (S, l, \prec)$ be a preferential model, $\alpha, \beta \in \mathcal{L}$ and $s \in S$.

$$s \models \alpha \vdash_{\mathcal{P}} \beta \quad \text{if} \quad \text{either } s \notin \min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{P}}) \text{ or } s \in \llbracket \beta \rrbracket^{\mathcal{P}}.$$

We can generalize Definition 1.13 further. Let $\mathcal{P} = (S, l, \prec)$ be a preferential model, $\alpha, \beta \in \mathcal{L}$. We say that \mathcal{P} satisfies $\alpha \vdash_{\mathcal{P}} \beta$ (and write $\mathcal{P} \models \alpha \vdash_{\mathcal{P}} \beta$) if for all $s \in \min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{P}})$, we have $s \in \llbracket \beta \rrbracket^{\mathcal{P}}$. We can write it also using the set inclusion operator \subseteq . We obtain:

$$\mathcal{P} \models \alpha \vdash_{\mathcal{P}} \beta \quad \text{if} \quad \min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{P}}) \subseteq \llbracket \beta \rrbracket^{\mathcal{P}}.$$

Example 1.14. Going to the preferential model \mathcal{P} represented in Figure 1.1. The set of p -states is $\llbracket p \rrbracket^{\mathcal{P}} = \{S_1, S_2, S_5, S_7\}$. The set of preferred p -states with respect to \prec is $\min_{\prec}(\llbracket p \rrbracket^{\mathcal{P}}) = \{S_1, S_7\}$. The consequence $p \rightarrow q \vee r$ is false, since $S_5 \models p$ and $S_5 \not\models q \vee r$. However, $p \vdash_{\mathcal{P}} q \vee r$ is true, since S_5 is not a preferred p -state with respect to \prec . Both preferred p -states, i.e., S_1 and S_7 , satisfy $q \vee r$. To summarise, we have:

- $\mathcal{P} \not\models p \rightarrow q \vee r$;
- $\mathcal{P} \models p \vdash_{\mathcal{P}} q \vee r$.

Finally, Kraus et al. proved in their work, that the inference $\vdash_{\mathcal{P}}$ of a preferential model \mathcal{P} is a preferential inference relation (see Definition 1.2). They showed that all the rules of system **P** are satisfied by the inference $\vdash_{\mathcal{P}}$ of any preferential model. Not only that, starting from a preferential inference, it is possible to define a preferential model \mathcal{P} such that the inference $\vdash_{\mathcal{P}}$ is the same as the inference relation \vdash .

Theorem 1.15 (Representation theorem for preferential inferences [KLM90]). *An inference relation \vdash is preferential if it is defined by a preferential model \mathcal{P} .*

System **P** is a good reasoning system, in the sense that, all of its rules enable the deduction of new inferences starting from a base of conditionals Δ that are coherent with it. In addition, the preferential models studied in the KLM framework are intuitive (the ordering relation \prec order the worlds from the unlikely to the most probable) and offer a well-defined inference relation that satisfies all the rules of system **P**. Many logics and formalisms were extended using preferential models. We highlight in this memoir a use case for preferential models in both Description and Modal logic.

However, a problem that system **P** cannot offer is a solution to problem of "non-pertinence" in de-feasible reasoning. The problem states as follows:

“If γ is a plausible inference of α and β is not an exceptional case of α , then γ is a plausible inference of $\alpha \wedge \beta$.”

Let us consider that an agent uses a reasoning similar to the common sense. This “rational” agent, when presented with a conditional of the form $\alpha \sim \gamma$, is able to infer that $\alpha \wedge \beta \sim \gamma$ in the absence of the fact that β is an exceptional case of α . Due to the nature of the cautious monotony rule, the agent can infer $\alpha \wedge \beta \sim \gamma$ only when $\alpha \sim \beta$ is present in the base of conditionals, i.e., when it is known that β is plausible case of α . Many studies were made by Freund [Fre91, Fre93] and Lehmann and Magidor [LM89] in order to come up with models which inference can satisfy the non-pertinence problem. In the upcoming section, we highlight yet another system in the KLM called the rational closure, denoted by **R**, that answers this problem.

1.1.3 System R or rational closure

To better illustrate the problem of non-pertinence, we return to the birds and penguins example:

Example 1.16. Let Δ contain the following conditional:

$$\Delta = \{\text{birds} \sim \text{flies}\}$$

Assuming that the symbol ‘red’ indicates “being red”. Starting from Δ , it is possible to infer that “red birds fly”, i.e., $\text{birds} \wedge \text{red} \sim \text{flies}$, using only rules of system **P**. Same as the case of penguins, red birds could potentially be an exceptional case of birds that do not fly (same as the case of penguins).

The cautious monotony does not take in account the non-pertinence of inferences. That is why, the KLM approach introduces a new system called rational closure or system **R**. In addition to the rules of **P**, Lehmann and Magidor [LM89] proposed a new rule called *rational monotony*:

- **Rational monotony**

$$\frac{\alpha \sim \gamma, \quad \alpha \not\sim \neg\beta}{\alpha \wedge \beta \sim \gamma} \quad (1.7)$$

This rule states that when γ is a plausible inference of α , it is possible add β to the premise of $\alpha \sim \gamma$, in the case when β does not plausibly contradicts α . The contradiction is formally expressed $\neg(\alpha \sim \neg\beta)$ (the absence of $\neg\beta$ being a plausible inference of α).

Given a set of starting conditionals Δ , a *rational inference* relation between two sentences α, β is defined as:

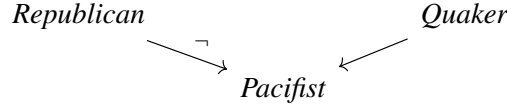
Definition 1.17 (Rational inference relation). Let $\alpha, \beta \in \mathcal{L}$, and Δ be a set of conditionals. An inference relation $\alpha \sim \beta$ is said to be rational if $\alpha \sim \beta$ can be deduced from Δ using only rules of system **R**.

Going back to Example 1.16 and thanks to the rational monotony rule, we can infer that “red birds normally fly”. Since $\text{birds} \sim \text{flies} \in \Delta$ and $\text{birds} \not\sim \neg\text{red} \notin \Delta$, we have $\text{red} \wedge \text{birds} \sim \text{flies}$. The rule of rational gives the agent the ability to infer more assertions than the cautious monotony rule. However, this create the problem of consistency of the base of conditionals Δ when using the rules of system **R**. First of all, we say that base of conditionals Δ is consistent if it does not contain two conditionals such as $\alpha \sim \beta$ and $\alpha \sim \neg\beta$. We illustrate the consistency problem using the “Nixon diamond” example. Nixon is a *republican* and a *quaker*. Quakers are generally *pacifists*. Whereas, republicans are generally *not pacifists*. The question is what defeasible conclusions are warranted on the basis of these two conditionals. In particular, should we infer that Nixon is a pacifist or that he is not a pacifist?

Example 1.18 (Nixon Diamond). *Let us consider the following conditional base Δ :*

$$\Delta = \{\text{republican} \sim \neg \text{pacifist}, \text{quaker} \sim \text{pacifist}\}.$$

*Let Δ^r be the closure of conditional statements inferred from Δ using the rules of **R**. We first initialize Δ^r by $\Delta^r = \Delta$.*



*fly Using rules of system **R**, from $\text{quaker} \sim \text{pacifist} \in \Delta^r$ and $\text{quaker} \sim \neg \text{republican} \notin \Delta^r$, we deduce that $\text{republican} \wedge \text{quaker} \sim \text{pacifist}$. We can add the new conditional to Δ^r . Similarly, from $\text{republican} \sim \neg \text{pacifist} \in \Delta^r$ and $\text{republican} \sim \neg \text{quaker} \notin \Delta^r$, we have $\text{republican} \wedge \text{quaker} \sim \neg \text{pacifist}$. However, since $\text{republican} \wedge \text{quaker} \sim \text{pacifist}$ is in Δ^r , adding the conditional $\text{republican} \wedge \text{quaker} \sim \neg \text{pacifist}$ to Δ^r renders it inconsistent.*

In general, the consistent closure of conditional bases Δ^r is not unique. In the Nixon diamond example, there are three consistent Δ^r . They are as follows:

- $\Delta_1^r = \{\text{republican} \sim \neg \text{pacifist}, \text{quaker} \sim \text{pacifist}, \text{republican} \wedge \text{quaker} \sim \text{pacifist}\};$
- $\Delta_2^r = \{\text{republican} \sim \neg \text{pacifist}, \text{quaker} \sim \text{pacifist}, \text{republican} \wedge \text{quaker} \sim \neg \text{pacifist}\};$
- $\Delta_3^r = \{\text{republican} \sim \neg \text{pacifist}, \text{quaker} \sim \text{pacifist}\}.$

Lehmann and Magidor [LM89] proposed a method of selecting a unique rational extension of a conditional base Δ , called the *rational closure* and denoted by Δ^{RC} . The rational closure of Δ can be seen as the “minimal” set of rational inferences that completes the set of conditionals. The basic idea is to assign a rank (a numerical value) for each sentence which indicates how exceptional it is relative to Δ . Then the ranks of formulas are minimized which means that each formula is interpreted as normally as possible. A conditional $\alpha \sim \beta$ is in the rational closure Δ^{RC} if, in the presence of α , we would like to conclude β rather than $\neg\beta$. Based on their ranking, the rank of $\alpha \wedge \beta$ is strictly less than $\alpha \wedge \neg\beta$. In what, we define in more details the rational closure.

Definition 1.19 (Exceptionality of sentences). *Let Δ be a base of conditionals and $\alpha, \beta \in \mathcal{L}$. The sentence α is said to be exceptional for Δ if $\top \sim \neg\alpha$ can be inferred from Δ . A conditional $\alpha \sim \beta$ is said to be exceptional for Δ if its premise α is exceptional for Δ . The set of conditionals of Δ which are exceptional for Δ is denoted by $E(\Delta)$.*

Assuming that the starting conditional base Δ is finite, it is possible to define a non-increasing sequence of subsets of Δ , $\Delta_0 \supseteq \Delta_1 \supseteq \dots \supseteq \Delta_k$ such that $\Delta_0 = \Delta$ and $\Delta_i = E(\Delta_{i-1})$ for all $i > 0$. The set Δ_i contains all exceptional conditionals of Δ_{i-1} . Since Δ is finite, there is an $k \geq 0$ where $\Delta_j = \Delta_k$ for all $j \geq k$. Note that Δ_k can either be empty $\Delta_k = \emptyset$ or not (in this case, conditionals in Δ_k are exceptional to all Δ_i where $0 \leq i \leq k$). The sets Δ_i are used to define the ranks of sentences.

Definition 1.20 (Rank of sentences). *Let Δ be a conditional base and $\alpha \in \mathcal{L}$. The sentence α has a rank $\text{rank}(\alpha) \stackrel{\text{def}}{=} i$ if i is the smallest natural number for which α is not exceptional for Δ_i . If α is exceptional for all Δ_i , we say that α has no rank and write $\text{rank}(\alpha) \stackrel{\text{def}}{=} \infty$.*

A conditional $\alpha \sim \beta$ has a rank equal to $\text{rank}(\alpha)$, and $\Delta_i \setminus \Delta_{i-1}$ is the set in Δ of conditionals having the rank i or a larger rank. When $\text{rank}(\alpha) = \infty$, we say that the conditional $\alpha \sim \beta$ has no rank.

Example 1.21. Let Δ contain the following conditionals:

- $Student \sim \neg PayTaxes$
- $Student \sim Young$
- $Student \wedge Employee \sim PayTaxes$

We start the construction by $\Delta_0 = \Delta$. We find that $rank(Student) = 0$, since $\top \sim \neg Student$ is not true in Δ_0 . Therefore both of the conditionals $Student \sim \neg PayTaxes$ and $Student \sim Young$ have a rank of 0. However $Student \wedge Employee$ is exceptional for Δ_0 . Using the rational monotony rule, since $Student \sim \neg PayTaxes$ and $Student \not\sim \neg Employee$, then we have $Student \wedge Employee \sim \neg PayTaxes$. This conditional statement conflicts with $Student \wedge Employee \sim PayTaxes$ in Δ_0 . We can infer $\top \sim \neg(Student \wedge Employee)$ in Δ_0 .

We obtain Δ_1 by $\Delta_1 = E(\Delta_0)$. The set Δ_1 has only the conditional $Student \wedge Employee \sim PayTaxes$ in it. We find that $Student \wedge Employee$ is not exceptional for Δ_1 (we cannot infer $\top \sim \neg(Student \wedge Employee)$ in Δ_1). Therefore, $rank(Student \wedge Employee) = 1$ and the conditional $Student \wedge Employee \sim PayTaxes$ has a rank of 1.

The set Δ_2 is empty since $E(\Delta_1) = \emptyset$. The process stops at this point.

- $\Delta_0 = \Delta$;
- $\Delta_1 = \{Student \wedge Employee \sim PayTaxes\}$.

Rational closure uses the notion of exceptionality. Let Δ^{RC} be a rational closure of Δ , a conditional $\alpha \sim \beta$ is in the rational closure Δ^{RC} if $\alpha \wedge \beta$ is less exceptional than $\alpha \wedge \neg\beta$. Intuitively speaking, we rather, in the presence of α , deduce β rather than $\neg\beta$. Lehmann and Magidor defined rational closure as follows.

Definition 1.22 (Rational closure). Let Δ be a conditional base and $\alpha, \beta \in \mathcal{L}$. The rational closure of Δ , denoted by Δ^{RC} is defined as:

$$\Delta^{RC} \stackrel{\text{def}}{=} \{\alpha \sim \beta \mid \text{either } rank(\alpha) < rank(\alpha \wedge \neg\beta) \text{ or } rank(\alpha) = \infty\}.$$

The definition of the rational closure as $\Delta^{RC} \stackrel{\text{def}}{=} \{\alpha \sim \beta \mid \text{either } rank(\alpha \wedge \beta) < rank(\alpha \wedge \neg\beta) \text{ or } rank(\alpha) = \infty\}$ yields the same Δ^{RC} as Definition 1.22. Lehmann and Magidor [LM89] showed that the rational closure Δ^{RC} is unique and minimal with respect to the set of all possible rational extension of Δ .

Moving on to the semantics of rational inferences. In what follows, we describe *ranked models* which are an extension of preferential models that use ordering relation \prec that are *modular*.

Definition 1.23 (modular ordering relation). Let \prec be a partial order on a set U and x, y be any element of the set U . The relation \prec is said to be modular if there is a totally ordered set Ω (the strict order on Ω is symbolised by $<$) and a ranking function $r : U \rightarrow \Omega$ such that $x \prec y$ if $r(x) < r(y)$.

A modular ordering relation is characterised by a ranking function which orders the elements of a set. In the context of models and given any two states s, s' , either s is preferred than s' (in this case $s \prec s'$), s' is preferred than s (in this case $s' \prec s$) or s is equally preferred than s' (in this case $r(s) = r(s')$).

Definition 1.24 (Ranked models). A ranked model $\mathcal{R} \stackrel{\text{def}}{=} (S, l, \prec)$ is a preferential model for which \prec is modular.

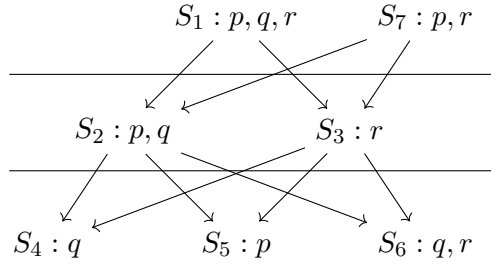


Figure 1.2: Example of a ranked model

Figure 1.2 depicts a ranked model $\mathcal{R} = (S, l, \prec)$.

We can see the ranks of states as drawers. The first drawer contains the states S_1, S_7 . The second contains S_2, S_3 . And the last drawer contains S_4, S_5, S_6 . All the states in the first drawer are more preferred than all the states in the second drawer. Similarly, states in the second drawer are more preferred than the third. States in the same drawers are not comparable amongst each other, they are considered to be equally preferable.

The satisfiability of rational inferences is similar to the satisfiability of preferential inferences in preferential models. Let $\mathcal{R} = (S, l, \prec)$ be a ranked model and $s \in S$, a conditional $\alpha \sim_{\mathcal{R}} \beta$ is true in a state s (denoted by $s \models \alpha \sim_{\mathcal{R}} \beta$) if either $s \notin \min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{R}})$ or $s \in \llbracket \beta \rrbracket^{\mathcal{R}}$. Moreover, we define the generalized satisfiability as follows:

$$\mathcal{R} \models \alpha \sim_{\mathcal{R}} \beta \quad \text{if } \min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{R}}) \subseteq \llbracket \beta \rrbracket^{\mathcal{R}}.$$

The representation theorem was given by Lehmann and Magidor.

Theorem 1.25 (Representation theorem for rational inferences [LM89]). *An inference relation \sim is rational if it is defined by a ranked model \mathcal{R} .*

Lehmann and Magidor [LM89] proposed also a semantics for a rational closure conditional base Δ^{RC} by selecting a unique ranked model. Let $\mathcal{P} = (S, l, \prec)$ be a preferential model of a conditional base Δ . We can build a ranked model $\mathcal{P}_{RC} = (S_{RC}, l_{RC}, \prec_{RC})$ such that $S_{RC} = S$ and $l_{RC} = l$. The ordering relation \prec_{RC} is a total order that completes \prec . Each state $s \in S_{RC}$ has a rank (a natural number) $r(s)$ which indicates the longest ascending chain on which s is the minimal element. Lehmann and Magidor [LM89] showed that any conditional $\alpha \sim \beta \in \Delta^{RC}$ if \mathcal{P}_{RC} satisfies it.

Theorem 1.26. *Let $\mathcal{P} = (S, l, \prec)$ be a preferential model of a conditional base Δ . Let Δ^{RC} be the rational closure of Δ and $\mathcal{P}_{RC} = (S_{RC}, l_{RC}, \prec_{RC})$ be the corresponding ranked model of \mathcal{P} . We have the following:*

$$\alpha \sim \beta \in \Delta^{RC} \quad \text{if} \quad \mathcal{P}_{RC} \models \alpha \sim \beta.$$

The rule of rational monotony is a less prudent approach than the rule of cautious monotony. In the sense that when adding new inferences, the rational monotony property is more lenient than the cautious monotony. However, a new issue arises when reasoning using system **R**. Let us go back to the birds and penguins (see Example 1.1). So far, the conditional base contains $\Delta = \{penguin \rightarrow bird, bird \sim flies, penguin \sim \neg flies\}$. Using cautious monotony, $penguin \wedge bird \sim \neg flies$ can be inferred. Which is the case, since penguins are an exceptional case of birds that do not fly. When adding the conditional $bird \sim wings$ (most of the birds have wings) and knowing that penguins are an exceptional case of birds; it is not possible to infer that “normally, birds that are penguins have wings”. Using rules of **R**, a

reasoner is still sceptical about attributing other plausible properties of birds that are penguins. System **R** and system **P** as NMR systems are said to suffer from the *drowning problem* Benferhat et al. [BCD⁺93].

Nonetheless, both of the systems proposed in the KLM approach (although being sceptical at times) offer a reliable framework for extending defeasible reasoning to other classical formalisms. In the upcoming section, we shall present studies of more complex formalisms, on which the KLM approach have been investigated. We shall see how their semantics are extended thanks to the preferential models. Moreover, we shall see how new aspects of defeasibility can be expressed in these formalisms.

1.2 Use cases of logical formalisms extended with KLM conditionals

We focus next on the theory developed around the notion of coping with exceptionality that NMR field can offer in general and the KLM approach in particular. As shown in the previous section, the two systems (**P** and **R**) provide a roadmap on how non-monotonic inferences should behave in order to express defeasibility. Many aspects of defeasibility in propositional and complex logics have been studied. We cite the works of (Boutilier [Bou94], Booth et al. [BMV12], Giordano et al. [GGOP07, GOGP09, GGOP13, GGOP09], Castilho [CHV02] and Britz et al. [BMV11b, BV18, BV19]) as examples. In this memoir, we shall expose two case studies of formalisms extended thanks to the KLM approach [KLM90, LM89]. These formalisms are description logic and modal logic. The reason is to show how the integration of preferential semantics within their respective semantics is done. The second reason is to show the adequacy of the KLM approach for expressing defeasibility in these formalisms. We expose defeasible description first as a vehicle to show that defeasibility is not present only within the context conditionals $\alpha \sim \beta$. It is possible to express defeasibility in other types of inferences. We discuss description logic and its defeasible extension in sections 1.2.1, 1.2.2. We also highlight a defeasible extension of modal logic, in the goal of showing a new aspect of defeasibility (outside the scope of inferences) made possible using the KLM approach. We discuss modal logic and its defeasible extension in sections 1.2.3, 1.2.4.

1.2.1 Description logic

Description Logics (DL) is a knowledge representation formalism (see Baader et al. [BCM⁺07] for a reference). DL can be used for modern AI and database applications in the field of knowledge representation and ontologies. A Description Logic knowledge base consists of two major elements: the *TBox* introduces concepts, relationships, and constraints of the domain, and the *ABox* contains assertions of individuals in the knowledge base. There is a family of description logics, we shall summarize the general DL language called \mathcal{ALC} .

The DL \mathcal{ALC} is built upon a finite set of atomic *concept names* \mathcal{C} , a finite set of *role names* \mathcal{R} and a finite set of *individuals* \mathcal{I} . In this section, we use the example of students (see Example 1.21) to highlight a DL knowledge base. For example, we can have $\mathcal{C} = \{Student, EmpStudent, Employee, Young, Tax\}$, $\mathcal{R} = \{pays, worksFor\}$ and $\mathcal{I} = \{alice, bob\}$. Complex concepts can be built from the atomic concepts C, D, \dots using the constructors \neg (complement), \sqcap (concept conjunction), \sqcup (concept disjunction), \forall (value restriction) and \exists (existential restriction). Let A be an atomic concept, C be a complex concept and r be a relation, the language of \mathcal{ALC} is recursively defined as follows:

$$C ::= A \mid C \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \mid \forall r.C$$

Examples of these complex concepts using the students and employees (Example 1.21) scenario are $Student \sqcap Employee$ and $\exists pays.Tax$. We explain the meaning of each of these sentences further down this section. With \mathcal{L} , we denote the language of all \mathcal{ALC} concepts. Moving on to the semantics of DL

\mathcal{ALC} , an *interpretation* is a pair $\mathcal{I} \stackrel{\text{def}}{=} \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where $\Delta^{\mathcal{I}}$ is a non-empty set called a *domain*, and $\cdot^{\mathcal{I}}$ is an interpretation function. The interpretation function $\cdot^{\mathcal{I}}$ maps each concept C to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, roles r to $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each individual a to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

Going back to our example, let assume that *alice* is a normal student and *bob* is an employee student. The domain is then $\Delta^{\mathcal{I}} = \{\text{alice}, \text{bob}\}$ and $\text{Student}^{\mathcal{I}} = \{\text{alice}\}$ and $\text{EmpStudent}^{\mathcal{I}} = \{\text{bob}\}$. Let $\mathcal{I} \stackrel{\text{def}}{=} \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, we define $r^{\mathcal{I}}(x) \stackrel{\text{def}}{=} \{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\}$. The set $r^{\mathcal{I}}(x) \subseteq \Delta^{\mathcal{I}}$ is the set of individuals in domains $\Delta^{\mathcal{I}}$ that are paired up with x through the relation $r^{\mathcal{I}}$. The truth values of complex concepts are defined as follows:

- $\top^{\mathcal{I}} \stackrel{\text{def}}{=} \Delta^{\mathcal{I}}$;
- $\perp^{\mathcal{I}} \stackrel{\text{def}}{=} \emptyset$;
- $(\neg C)^{\mathcal{I}} \stackrel{\text{def}}{=} \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$;
- $(C \sqcap D)^{\mathcal{I}} \stackrel{\text{def}}{=} C^{\mathcal{I}} \cap D^{\mathcal{I}}$;
- $(C \sqcup D)^{\mathcal{I}} \stackrel{\text{def}}{=} C^{\mathcal{I}} \cup D^{\mathcal{I}}$;
- $(\exists r.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(x) \cap C^{\mathcal{I}} \neq \emptyset\}$;
- $(\forall r.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(x) \subseteq C^{\mathcal{I}}\}$.

The final element in DL \mathcal{ALC} is the subsumption statement (also called concept inclusion), denoted by $C \sqsubseteq D$. As its name suggests, C is subsumed by D means that all individuals in C are in D . In the context of knowledge bases, subsumption statements are considered to be the constraints that a knowledge base must adhere to; more on that after Example 1.27. The \mathcal{ALC} *TBox* \mathcal{T} is a finite set of subsumption statements. The \mathcal{ALC} *ABox* \mathcal{A} contains assertions of the form $a : C$ and $(a, b) : r$, that read as a is an instance of C and b is paired to a by r respectively. A knowledge base (KB) is defined as a pair $\mathcal{K} \stackrel{\text{def}}{=} (\mathcal{T}, \mathcal{A})$. Example 1.27 depicts a DL knowledge for the student and employee scenario.

Example 1.27. Here is a DL representation of the students and employee students scenario. The knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of:

$$\mathcal{T} = \left\{ \begin{array}{l} \text{Student} \sqsubseteq \neg \exists \text{pays.Tax}, \\ \text{Student} \sqsubseteq \text{Young}, \\ \text{EmpStudent} \sqsubseteq \text{Student} \sqcap \text{Employee}, \\ \text{EmpStudent} \sqsubseteq \exists \text{pays.Tax}, \end{array} \right\}$$

$$\mathcal{A} = \{\text{alice} : \text{Student}, \text{bob} : \text{EmpStudent}\}$$

Given a KB \mathcal{K} and an interpretation \mathcal{I} , \mathcal{I} satisfies a subsumption statement $C \sqsubseteq D$, denoted by $\mathcal{I} \models C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. The interpretation \mathcal{I} satisfies an assertion $a : C$ (respectively $(a, b) : r$), denoted by $\mathcal{I} \models a : C$ (respectively $\mathcal{I} \models (a, b) : r$, if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (respectively $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$). We say that \mathcal{I} is a *model* of \mathcal{T} (denoted by $\mathcal{I} \models \mathcal{T}$) if $\mathcal{I} \models \alpha$ for every subsumption statement $\alpha \in \mathcal{T}$. A similar definition can be obtained for models of *ABoxes* \mathcal{A} . An interpretation \mathcal{I} is said to be a model of the knowledge base \mathcal{K} , if \mathcal{I} is a model of its *TBox* \mathcal{T} and its *ABox* \mathcal{A} .

A statement α (α can either be a subsumption rule $C \sqsubseteq D$ or an assertion statement $a : C$) is entailed by a knowledge base \mathcal{K} if every model of \mathcal{K} satisfies α . In Example 1.27, we can see that $\mathcal{K} \models \text{EmpStudent} \sqsubseteq \text{Student}$ and therefore $\mathcal{K} \models \text{EmpStudent} \sqsubseteq \neg \exists \text{pays.Tax}$. This conflicts with the statement $\text{EmpStudent} \sqsubseteq \exists \text{pays.Tax}$ in \mathcal{T} , i.e., $\mathcal{K} \models \text{EmpStudent} \sqsubseteq \perp$. The concept *EmpStudent*

is an atypical (or exceptional) case of *Student* that pay taxes. The statement $Student \sqsubseteq \neg \exists pays.Tax$ is inadequate for representing about edge cases of *Student* that do pay their taxes.

It turns out, the subsumption operator behaves in a similar manner as the classical entailment \rightarrow . Both of these operators behave monotonically, in the sense that, they do not allow to formalize different nuances of exceptions and how to treat them. On the other hand, NMR is a field that offers many formalisms for expressing and modelling problems that might contain exceptional cases. Just as highlighted in Section 1.1, the work of KLM lays the foundation of models extended with a relation \prec that captures the nuances of preference between different elements in a model. Moreover, the KLM study gives a thorough analysis of properties in order for an inference to be deemed non-monotonic (from the preferential inferences of system **P**, to the rational closure of system **R**). Over the years, many authors introduced defeasible extensions of description logic. We focus in this memoir, on the preferential extension $\mathcal{ALC}+\mathbf{T}$ and the notion *typicality* of a concept in the work of Giordano et al. [GOGP09]. We highlight also defeasible description logic (defeasible \mathcal{ALC}) in the work of Britz et al. [BCM⁺20].

1.2.2 Defeasible description logic

Going back to Example 1.27, one might consider that “*Student typically do not pay taxes*”. In other words, the most typical or normal instances of *Student* do not pay taxes. Leaving in a sense, other atypical instances of *Student* to pay taxes. Such is the case of the concept *EmpStudent* in general and the individual *bob* in particular. In the defeasible extension $\mathcal{ALC}+\mathbf{T}$, Giordano et al. [GOGP09] proposed the introduction of the operator of *typicality* **T**. The intended meaning is that, for any concept C , $\mathbf{T}(C)$ contains the instance that are deemed to be the most typical of C . Therefore subsumption rules such as “*Student typically do not pay taxes*” can be represented by $\mathbf{T}(Student) \sqsubseteq \neg \exists pays.Tax$.

In addition of the \mathcal{ALC} language \mathcal{L} (defined in Section 1.2.1), the notion of extended concepts is introduced. Given a concept C , the concept itself and $\mathbf{T}(C)$ are considered as *extended concepts*. Moreover, all the Boolean combination of extended concepts are also extended concepts. Knowledge bases are still defined as the pair $\mathcal{K} \stackrel{\text{def}}{=} (\mathcal{T}, \mathcal{A})$ where \mathcal{T} is the *TBox* of \mathcal{K} and \mathcal{A} is the *ABox*. Subsumption rules $C \sqsubseteq D$ are contained in \mathcal{T} , where $C \in \mathcal{L}$ is an *extended concept* that can either be C' or $\mathbf{T}(C')$, and $D \in \mathcal{L}$ is a *normal concept*. *ABoxes* \mathcal{A} contain assertions of the form $a : C$ and $(a, b) : r$ where C is an extended concept.

Interpretations in $\mathcal{ALC}+\mathbf{T}$ are defined as follows.

Definition 1.28. (*Semantics of **T***) An interpretation is a tuple $\mathcal{I} \stackrel{\text{def}}{=} \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, f_{\mathbf{T}} \rangle$. $\Delta^{\mathcal{I}}$ is the domain. The function $\cdot^{\mathcal{I}}$ maps each concept C to $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, role r to $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and individual a to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The function $f_{\mathbf{T}} : \text{Pow}(\Delta^{\mathcal{I}}) \rightarrow \text{Pow}(\Delta^{\mathcal{I}})$ is a selection function that returns the instances of the extended concept $\mathbf{T}(C)$ for each concept C , i.e., $f_{\mathbf{T}}(C^{\mathcal{I}}) = (\mathbf{T}(C))^{\mathcal{I}}$. We say that the function $f_{\mathbf{T}}$ selects typical instances of concepts if it satisfies the following postulates.

1. $f_{\mathbf{T}}(C^{\mathcal{I}}) \subseteq C^{\mathcal{I}}$;
2. if $C^{\mathcal{I}} \neq \emptyset$, then $f_{\mathbf{T}}(C^{\mathcal{I}}) \neq \emptyset$;
3. if $f_{\mathbf{T}}(C^{\mathcal{I}}) \subseteq D^{\mathcal{I}}$, then $f_{\mathbf{T}}(C^{\mathcal{I}}) = f_{\mathbf{T}}((C \sqcap D)^{\mathcal{I}})$;
4. $f_{\mathbf{T}}(\bigsqcup (C_i)^{\mathcal{I}}) \subseteq \bigcup f_{\mathbf{T}}((C_i)^{\mathcal{I}})$;
5. $\bigcap f_{\mathbf{T}}(C_i) \subseteq f_{\mathbf{T}}(\bigcap C_i)$.

Given a concept $C \in \mathcal{L}$, the extended concept $\mathbf{T}(C)$ captures the typical individuals of C . The postulate (1) states that typical individuals of C are in C . The postulate (2) requests that any non-empty concept C has typical individuals. According to the postulate (3), if the typical individuals of C are in

D , then they coincide with the typical element of both, i.e., $C \sqcap D$. This postulate expresses a weak form of monotony (namely, cautious monotony). The postulate (4) stipulates that the typical individuals of a disjunction of concepts C_i is included in the union of typical individuals of each concept. Whereas for postulate (5), the intersection of typical individuals of each C_i is included in the typical individuals of their conjunction.

Giordano et al. [GOGP09] provide semantics for \mathbf{T} based on the preference relation \prec as defined in KLM. The intuition is to extend DL interpretations by adding an ordering relation \prec on individuals of the domain $\Delta^{\mathcal{I}}$ which symbolizes the preference of individuals amongst each other. In this setting, the selection function that returns the typical individuals of a concept $f_{\mathbf{T}}(C^{\mathcal{I}})$ consists of the minimal individual of C with respect to the ordering relation \prec . The characteristics of the preference relation \prec are analogous to the ones described in the KLM approach, we invite the reader to refer back from Definition 1.6 to Definition 1.8 for its properties. Formally, preferential interpretations of $\mathcal{ALC}+\mathbf{T}$ are defined as follows.

Definition 1.29 (Preferential interpretations). *A preferential interpretation is a tuple $\mathcal{I} \stackrel{\text{def}}{=} \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \prec^{\mathcal{I}} \rangle$. $\Delta^{\mathcal{I}}$ is the domain. The function $\cdot^{\mathcal{I}}$ maps each concept C to $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each role r to $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each individual a to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The relation $\prec^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ is a strict partial order on the domain $\Delta^{\mathcal{I}}$ that satisfies the smoothness condition. For each extended concept $\mathbf{T}(C)$, we have $\mathbf{T}(C)^{\mathcal{I}} \stackrel{\text{def}}{=} \min_{\prec^{\mathcal{I}}}(C^{\mathcal{I}})$.*

Giordano et al. [GOGP09] showed that, given an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, f_{\mathbf{T}} \rangle$ with a typical selection function $f_{\mathbf{T}}$, it is possible to define in the same domain $\Delta^{\mathcal{I}}$ a preference relation \prec such that for all $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, $f_{\mathbf{T}}(C)$ is the set of minimal individuals of the concept C with respect to the relation \prec . Moreover, it is possible to define, from a preferential interpretation with a preference relation \prec , an interpretation with a selection function $f_{\mathbf{T}}$ that satisfies the postulates (1-5) of typicality. The representation states the following:

Theorem 1.30 (Representation theorem - Giordano et al. [GOGP09]). *Given any interpretation $\mathcal{I} \stackrel{\text{def}}{=} \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, f_{\mathbf{T}} \rangle$ satisfying the postulates (1-5) (see Definition 1.28) if it is possible to define on $\Delta^{\mathcal{I}}$ a strict partial order $\prec^{\mathcal{I}}$ satisfying the smoothness condition such that for all $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, $f_{\mathbf{T}}(C^{\mathcal{I}}) = \min_{\prec^{\mathcal{I}}}(C^{\mathcal{I}})$.*

Using the syntax of $\mathcal{ALC}+\mathbf{T}$ and preferential interpretations (see Definition 1.29), subsumption rules of the form “typically, individuals in C are in D ” are expressed using the typicality operator $\mathbf{T}(C) \sqsubseteq D$. DL interpretations extended with the preferential relation \prec offers an intuitive framework for the selection of typical individuals of a given concept. In the sense that, the minimal individual of a concept with respect to \prec are considered to be the most typical of said concept. The satisfiability and entailment are defined in the same way as in the classical DL interpretations (they are defined after Example 1.27).

Example 1.31. *Here is a DL representation of the students and employee students scenario using the language $\mathcal{ALC}+\mathbf{T}$. The knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of:*

$$\mathcal{T} = \left\{ \begin{array}{l} \mathbf{T}(\text{Student}) \sqsubseteq \neg \exists \text{ pays.Tax}, \\ \mathbf{T}(\text{Student}) \sqsubseteq \text{Young}, \\ \text{EmpStudent} \sqsubseteq \text{Student} \sqcap \text{Employee}, \\ \text{EmpStudent} \sqsubseteq \exists \text{ pays.Tax}, \end{array} \right\}$$

$$\mathcal{A} = \{\text{alice} : \text{Student}, \text{bob} : \text{EmpStudent}\}$$

Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \prec^{\mathcal{I}} \rangle$ be a DL interpretation where $\Delta^{\mathcal{I}} = \{\text{alice}, \text{bob}\}$. We initialize the concepts $\text{Student}^{\mathcal{I}} = \{\text{alice}\}$, $\text{EmpStudent}^{\mathcal{I}} = \{\text{bob}\}$ and the preference relation $\prec^{\mathcal{I}} = \{(\text{alice}, \text{bob})\}$ (which indicates $\text{alice} \prec \text{bob}$). From \mathcal{K} , the assertion $\text{bob} : \text{Student}$ is entailed. However, the individual bob is

not a typical instance of *Student* in the interpretation \mathcal{I} since $(alice, bob) \in \prec^{\mathcal{I}}$. In this case, we have $bob \notin (\mathbf{T}(\text{Student}))^{\mathcal{I}}$. There is no conflict in \mathcal{K} and the interpretation \mathcal{I} satisfies its TBox \mathcal{T} and its ABox \mathcal{A} .

Giordano et al. [GOGP09] proposed a DL framework where *defeasible* subsumption can be expressed. The extended concept $\mathbf{T}(C)$ by itself is the concept that contains all the typical individuals of the concept C . Thus, defeasible subsumption rules are of the form $\mathbf{T} C \sqsubseteq D$. Britz et al. [BCM⁺20] studied the properties of such defeasible subsumptions relation. In their setting, they introduced a defeasible subsumption rule operator \sqsubseteq that can replace statement such as $\mathbf{T} C \sqsubseteq D$ by $C \sqsubseteq D$. Britz et al. [BCM⁺20] showed that defeasible subsumption rules induced by preferential models (see Definition 1.29) follow the same rules of system **P** (Axiom 1.1 to Axiom 1.6). We formalize in what follows the syntax and semantics of the operator \sqsubseteq in Britz et al. [BCM⁺20] defeasible extension to DL.

For the remainder of this section, we shall assume that \mathcal{L} denotes the set of well-formed statements expressed using the classical \mathcal{ALC} language (without the operator \mathbf{T}). We shall also assume that preferential interpretations in defeasible DL are defined in the same manner as Definition 1.29.

Definition 1.32 (Defeasible subsumption relation). *A defeasible subsumption relation is a binary relation $\sqsubseteq \subseteq \mathcal{L} \times \mathcal{L}$.*

Statements of the form $C \sqsubseteq D$ read as “Normally, individuals in C are in D ”. A knowledge base \mathcal{K} in defeasible DL has three components: a TBox \mathcal{T} , an ABox \mathcal{A} and a DTBox \mathcal{D} . The defeasible TBox (denoted by *DTBox*) contains defeasible subsumption rule of \mathcal{K} . Formally a knowledge base in defeasible is defined as a tuple $\mathcal{K} \stackrel{\text{def}}{=} (\mathcal{T}, \mathcal{D}, \mathcal{A})$.

Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \prec^{\mathcal{I}} \rangle$ be a preferential DL interpretation. The interpretation \mathcal{I} is said to satisfy a defeasible subsumption $C \sqsubseteq D$ (denoted by $\mathcal{I} \models C \sqsubseteq D$) if the following is true:

$$\mathcal{I} \models C \sqsubseteq D \quad \text{if} \quad \min_{\prec^{\mathcal{I}}}(C^{\mathcal{I}}) \subseteq D^{\mathcal{I}}.$$

A statement $C \sqsubseteq D$ is true in an interpretation \mathcal{I} if all minimal individuals of C with respect to $\prec^{\mathcal{I}}$ are contained in D . A defeasible subsumption is said to be *preferential* if it satisfies the axioms mentioned below.

1. Reflexivity

$$A \sqsubseteq A \tag{1.8}$$

2. Left Logical Equivalence

$$\frac{A \equiv B \ (A \sqsubseteq B \text{ and } B \sqsubseteq A), \quad A \sqsubseteq C}{B \sqsubseteq C} \tag{1.9}$$

3. Right weakening

$$\frac{A \sqsubseteq B, \quad C \sqsubseteq A}{C \sqsubseteq B} \tag{1.10}$$

4. Cut

$$\frac{A \sqcap B \sqsubseteq C, \quad A \sqsubseteq B}{A \sqsubseteq C} \tag{1.11}$$

5. **Cautious monotony**

$$\frac{A \sqsubseteq B, \quad A \sqsubseteq C}{A \sqcap B \sqsubseteq C} \quad (1.12)$$

6. **OR**

$$\frac{A \sqsubseteq C, \quad B \sqsubseteq C}{A \sqcup B \sqsubseteq C} \quad (1.13)$$

The aforementioned properties are a direct translation of the properties of the non-monotonic inference \vdash in preferential systems. Since the non-monotonic inferences \vdash captures defeasibility when reasoning in the case of propositional logic. The goal here is to show defeasible subsumption \sqsubseteq captures defeasibility when reasoning about DL knowledge bases. Britz et al. [BCM⁺20] provide the representation theorem preferential subsumption relation.

Definition 1.33 (\mathcal{I} -induced defeasible subsumption [BCM⁺20]). *Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \prec^{\mathcal{I}} \rangle$ be a preferential interpretation. Then $\sqsubseteq_{\mathcal{I}} \stackrel{\text{def}}{=} \{(C, D) \mid \mathcal{I} \models C \sqsubseteq D\}$ is the defeasible subsumption induced by \mathcal{I} .*

There is a full correspondence between the class of preferential subsumption relations \sqsubseteq (postulates 1.8 to 1.13 are true) and the class of defeasible subsumption induced by preferential interpretations (interpretations where the preference relation \prec is a strict partial order that satisfies the smoothness condition).

Theorem 1.34 (Representation theorem for preferential subsumption [BCM⁺20]). *A defeasible subsumption \sqsubseteq is preferential if there is a preferential interpretation \mathcal{I} such that $\sqsubseteq_{\mathcal{I}} = \sqsubseteq$.*

In addition to preferential interpretations, Britz et al. [BCM⁺20] studied defeasible subsumption relations that might be considered *rational*. Following the KLM approach to system **R**, a defeasible subsumption relation is said to be rational if it satisfies, in addition to preferential rules, the rational monotony rule.

• **Rational monotony**

$$\frac{A \sqsubseteq C, \quad A \not\sqsubseteq \neg B}{A \sqcap B \sqsubseteq C} \quad (1.14)$$

Since the preference relation \prec for DL interpretations is similar to the preference relation in the case of propositional. *Ranked* DL interpretations are described preferential DL interpretations where \prec is modular (see Definition 1.23).

Definition 1.35 (Ranked interpretations). *A ranked interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \prec^{\mathcal{I}} \rangle$ is a preferential interpretation where $\prec^{\mathcal{I}}$ is modular.*

Subsumption relation induced from ranked are defined in the same fashion as in Definition 1.33.

Definition 1.36 (\mathcal{I} -induced defeasible subsumption). *Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \prec^{\mathcal{I}} \rangle$ be a ranked interpretation. Then $\sqsubseteq_{\mathcal{I}} \stackrel{\text{def}}{=} \{(C, D) \mid \mathcal{I} \models C \sqsubseteq D\}$ is the defeasible subsumption induced by \mathcal{I} .*

Analogous to system **R**, Britz et al. [BCM⁺20] showed that rational subsumption can be represented by ranked DL interpretations.

Theorem 1.37 (Representation theorem for preferential subsumption [BCM⁺20]). *A defeasible subsumption \sqsubseteq is modular if there is a ranked interpretation \mathcal{I} such that $\sqsubseteq_{\mathcal{I}} = \sqsubseteq$.*

The results of Theorem 1.34 and Theorem 1.37 are significant. They provide the characterization of defeasible subsumption rule, that is, preferential and rational subsumption. One sees that defeasible subsumption \sqsubseteq (whether it is preferential or rational) is characterized by the same set of axioms as its defeasible inference counterpart in propositional logic. Therefore, when inferring new defeasible subsumption rules, the reasoner can apply the same principles of the KLM approach to NMR (from inferring “cautiously” when handling preferential subsumption, to inferring “rationally” when handling rational subsumption). The rational closure is also studied in defeasible DL, we refer the reader to Giordano et al. [GGOP15] and Britz et al. [BCM⁺20].

Defeasible DL is a use case for a complex formalism (more expressive than propositional logic) where it is possible to perform defeasible reasoning using the KLM approach to NMR. The second use case we highlight in this memoir is defeasible modal logic, studied by Britz and Varzinczak [BV18]. Defeasible modal logic introduces a new aspect of defeasibility outside the scope of defeasible inferences, i.e., \vdash and \sqsubseteq .

1.2.3 Modal logic

The second use case of formalisms extend with the KLM approach we shall highlight is defeasible modal logic. We introduce first modal logic. Broadly speaking, modal logic studies reasoning that involves the use of the expressions “necessarily” and “possibly”. However, the term “modal logic” is used to cover a family of logics with similar rules and a variety of different symbols. Here is a list describing the known varieties of modal logic

Logic	Symbols	Expressions Symbolized
Modal logic	\Box	It is necessary that ...
	\Diamond	It is possible that ...
Deontic logic	O	It is obligatory that ...
	P	It is permitted that ...
	F	It is forbidden that ...
Temporal logic	\Box	It will always be the case that ...
	\Diamond	It will eventually be the case that ...
Doxastic logic	Bx	x believes that ...

We shall provide, in this section, the syntax and semantics of modal logic **K** (after the seminal work of Kripke [Kri80, Kri59]). Let \mathcal{P} be a set of atomic propositions. The set of Boolean connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and a set of operators \Box, \Diamond called modalities. The operator \Box symbolises necessity and \Diamond symbolises possibility. The modal logic language is recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha$$

All the other Boolean connectives are defined in terms of \neg and \wedge in the usual way (see preamble in Section 1.1.1). With \mathcal{L} , we denote the set of well-formed sentences of modal logic. The semantics is the standard possible-world semantics:

Definition 1.38 (Kripke semantics). *Let \mathcal{P} be a set of atomic propositions. A Kripke model is a tuple $\mathcal{J} \stackrel{\text{def}}{=} (S, R, V)$ where S is a set of states, $R \subseteq S \times S$ is a relation called the accessibility relation,*

$V : S \longrightarrow 2^{\mathcal{P}}$ is a valuation function which assigns truth values of atomic propositions to each state $s \in S$.

When modelling a problem of a system using modal logic, the set S are states s of the systems. The valuation function V maps each state $s \in S$ to a world $V(s) \in 2^{\mathcal{P}}$ in the universe of reference. The accessibility relation R describes outcomes of actions made on the system. In the sense that, the system may go from a state s to a state s' as a result an exercised action on the system, denoted by $s \longmapsto s'$. The accessibility relation R stores which states can access to which states of the system. If there are multiple agents interacting with the system at hand, it is possible to define multiple accessibility relations R_i in order to describe different actions. For the sake of simplicity, we shall assume in this memoir that there is only one agent interacting with a system. We use the notation $R(s) \stackrel{\text{def}}{=} \{s' \in S \mid (s, s') \in R\}$ to indicate all states s' that are accessible to s via the accessibility relation R .

Next, we highlight the truth values of sentences in modal logic. Let $\mathcal{J} = (S, R, V)$ be a Kripke model and $s \in S$ be a state, the truth values of sentences in s (denoted by $\mathcal{J}, s \models \alpha$) is defined as follows:

- $\mathcal{J}, s \models p$ if $p \in V(s)$;
- $\mathcal{J}, s \models \neg\alpha$ if $\mathcal{J}, s \not\models \alpha$;
- $\mathcal{J}, s \models \alpha \wedge \beta$ if $\mathcal{J}, s \models \alpha$ and $\mathcal{J}, s \models \beta$;
- $\mathcal{J}, s \models \alpha \vee \beta$ if $\mathcal{J}, s \models \alpha$ or $\mathcal{J}, s \models \beta$;
- $\mathcal{J}, s \models \Box\alpha$ if $\mathcal{J}, s' \models \alpha$ for all s' in $R(s)$;
- $\mathcal{J}, s \models \Diamond\alpha$ if $\mathcal{J}, s' \models \alpha$ for some s' in $R(s)$.

Given $\alpha \in \mathcal{L}$ and $\mathcal{J} = (S, R, V)$, we say that \mathcal{J} satisfies α if there is at least one state $s \in S$ such that $\mathcal{J}, s \models \alpha$. We say that \mathcal{J} is a model of α , denoted by $\mathcal{J} \models \alpha$, if $\mathcal{J}, s \models \alpha$ for all states $s \in S$. A sentence $\alpha \in \mathcal{L}$ is valid if every Kripke model \mathcal{J} is a model of α . We shall denote valid sentences by $\models \alpha$. One such validity is duality between the possibility and necessity operator. We have the following:

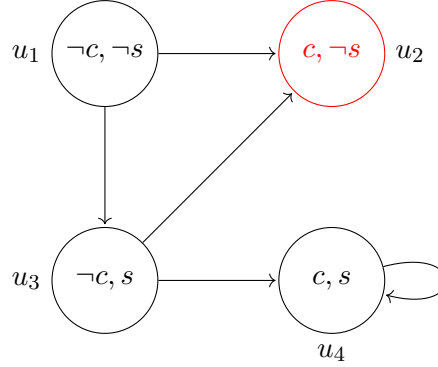
$$(\text{Duality}) \models \Box\alpha \leftrightarrow \neg\Diamond\neg\alpha$$

We are highlighting the system of normal modal logic **K**, of which all the other modal logic are extensions. Semantically, **K** is represented by the class of all Kripke models. Syntactically, these two principles (RN) and (K) characterize sentences of modal logic **K**:

- Necessitation rule: (RN) if α is valid, so is $\Box\alpha$;
- Distributivity axiom: (K) $\models \Box(\alpha \rightarrow \beta) \rightarrow (\Box\alpha \rightarrow \Box\beta)$.

According to the necessitation rule, any valid sentence of the logic is necessary. The distributivity axiom says if it is necessary that α implies β , then if necessarily α , then necessarily β . These two principles are considered the basis for accounting necessity. Many more principles and modal logic systems were developed over the years in order to capture the properties of the necessity. As we are interested in an extension of temporal logic called linear temporal logic, we shall discuss its axioms in the upcoming chapter. For more details about different modal logic systems, we refer the reader to Chellas [Che80].

Example 1.39 (Message encryption). We want to model a system of messages encryption. We shall check two properties of the system. The first property is whether the message is encrypted or not (denoted by c). The second property to check is whether the message is safe or not (denoted by s). Let $\mathcal{P} = \{c, s\}$ and $\mathcal{I} = (S, R, V)$ be the model for this system. We have four states $S = \{u_1, u_2, u_3, u_4\}$ where $V(u_1) = \{\}$, $V(u_2) = \{c\}$, $V(u_3) = \{s\}$ and $V(u_4) = \{c, s\}$. The relation R describes the transitions from states u_i to u_j . We also have $R = \{(u_1, u_2), (u_1, u_3), (u_3, u_2), (u_3, u_4), (u_4, u_4)\}$. Below is a graphical representation of the model.



We want to make sure that, necessarily in every state transition, an encrypted message implies that it is safe. In other words, is \mathcal{I} a model for the sentence $\Box(c \rightarrow s)$. This is not the case, we have $\mathcal{I}, u_1 \not\models \Box(c \rightarrow s)$. See that since the state u_2 is in $R(u_1)$ and $\mathcal{I}, u_2 \not\models c \rightarrow s$, we conclude that $\mathcal{I}, u_1 \not\models \Box(c \rightarrow s)$. Therefore, we have $\mathcal{I} \not\models \Box(c \rightarrow s)$.

The agent interacting with the system in Example 1.39 might consider the state u_2 as an exceptional state. Moreover, between the two accessible states of u_1 , i.e., u_2 and u_3 , the most normal transition from u_1 is u_3 . In this sense, rather than checking whether all possible outcomes satisfy $c \rightarrow s$, one might check that the *normal* outcomes satisfy it. As seen in the KLM approach and the defeasible extension of DL, the theory developed around preferential models *à la* KLM focuses on the ability to cope with exceptionality when performing inferences. However, in the case of modal logic, defeasibility might present itself as an outcome of an action (such as Example 1.39) rather than the premise of inference. The aim of a defeasible extension is then to formalize the notion of defeasible necessity and possibility. These notions were investigated by Britz and Varzinczak [BV18].

1.2.4 Defeasible modal logic

Britz and Varzinczak [BV18] defined defeasible versions of necessity (\Box) and possibility (\Diamond) in modal logics and their role to describe the *normality* of the outcome of an action. We shall introduce first, the preferential extension of Kripke models.

Definition 1.40 (Preferential Kripke models). A preferential Kripke model is a tuple $\mathcal{I} \stackrel{\text{def}}{=} (S, R, V, \prec)$ where S is a set of states, R is the accessibility relation, $V : S \rightarrow 2^{\mathcal{P}}$ is a valuation function and \prec is a strict partial order on S that satisfies the smoothness condition.

Preferential Kripke models are extensions of Kripke models that have a preference relation \prec over states. Intuitively speaking, the preference relation \prec is a strict partial order such that the more states are lower w.r.t. \prec , the more preferred they are (or deemed to be more normal) than those that are higher up w.r.t. \prec . The smoothness condition ensures that for any non-empty subset $S' \subseteq S$, the set of minimal states of S' is always non-empty.

The language of defeasible modal logic is recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha \mid \Box\alpha \mid \Diamond\alpha$$

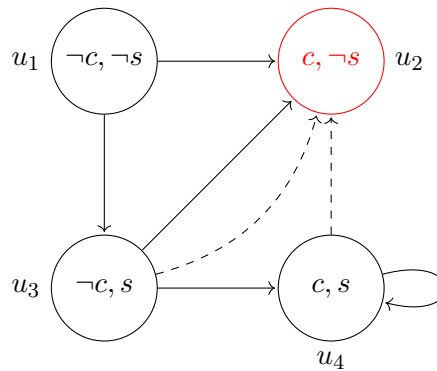
The defeasible modality \Box reads as *defeasible necessity*, and \Diamond reads *defeasible possibility*. With \mathcal{L}^\Box , we denote the set of well-formed sentences of defeasible modal logic. Next, we shall discuss the truth values behind these defeasible operators. Let $S' \subseteq S$, then $\min_{\prec}(S')$ denotes the set of minimal elements of S' with respect to \prec . The truth values modalities in defeasible modal logic are defined as follows.

Definition 1.41. Let $\mathcal{J} \stackrel{\text{def}}{=} (S, R, V, \prec)$ be a preferential Kripke model and $s \in S$.

- $\mathcal{J}, s \models \Box\alpha$ if $\mathcal{J}, s' \models \alpha$ for all s' in $R(s)$;
- $\mathcal{J}, s \models \Diamond\alpha$ if $\mathcal{J}, s' \models \alpha$ for some s' in $R(s)$;
- $\mathcal{J}, s \models \Box\alpha$ if $\mathcal{J}, s' \models \alpha$ for all s' in $\min_{\prec}(R(s))$;
- $\mathcal{J}, s \models \Diamond\alpha$ if $\mathcal{J}, s' \models \alpha$ for some s' in $\min_{\prec}(R(s))$.

The sentence $\Box\alpha$ is true if all the minimal states that are accessible to s via R satisfy the sentence α . The sentence $\Diamond\alpha$ is true if some minimal states that are accessible to s via R satisfy the sentence α . We can see that defeasible modalities behave in a similar fashion as their classical counterparts. In addition, defeasible modalities single out the preferred reachable state, by taking into account their order with respect to the relation \prec . The truth values of classical sentences are preserved in preferential Kripke models. The notion of satisfiability, models and validity in preferential Kripke models are defined the same way as in classical Kripke models.

Example 1.42 (Message encryption). The following preferential Kripke is a better representation of the system for message encryption. Let $\mathcal{J} = (S, R, V, \prec)$ be a model such that S, R and V are the same as in Example 1.39. The preference relation contains $\prec = \{(u_3, u_2), (u_4, u_2)\}$ which indicates that both u_3 and u_4 are more preferred than u_2 . Below is a graphical representation of the system (the preference relation is highlighted by dashed arrows).



We can check now for $\Box(c \rightarrow s)$. The latter sentence indicates that, for each state $u_i \in S$, the most normal outcomes of an action in this system are states where $c \rightarrow s$ is true. For u_1 , we have $R(u_1) = \{u_2, u_3\}$ and $\min_{\prec}(R(u_1)) = \{u_3\}$. Since $\mathcal{J}, u_3 \models c \rightarrow s$, we conclude that $\mathcal{J}, u_1 \models \Box(c \rightarrow s)$. For u_2 , since $R(u_2) = \emptyset$, we have $\min_{\prec}(R(u_2)) = \emptyset$. Nonetheless, we consider that $\mathcal{J}, u_2 \models \Box(c \rightarrow s)$. We leave the reader to check the remaining two states. The preferential Kripke model \mathcal{J} is a model for the sentence $\Box(c \rightarrow s)$, i.e., $\mathcal{J} \models \Box(c \rightarrow s)$.

It is worth to note that the preference relation is not explicit in the syntax. In the sense that, defeasible sentences of the form $\boxdot\alpha$ and $\diamond\alpha$ implicitly single out the most normal accessible states without directly referring to the order between them. This approach of defeasible operator is inspired by Booth et al.'s [BMV12] work on the extension of propositional logic that deals with “typical” α -worlds. The current definition of defeasible modalities stands in contrast to the approaches of Boutilier [Bou94] and Giordano et al. [GGOP09] which cast the preference as a modality (usually using the operator \Box) in the language level. This is why in Britz et Varzinczak's [BV18] approach, the preference relation is not seen as an extra accessibility relation. This makes it easier to integrate KLM preferential models in the case of modal logic. Defeasible modalities offer a way to express defeasible version of necessity and possibility that are similar in spirit to their classical counterparts.

We discuss now the properties of defeasible modalities. Same as classical modalities, there is a duality between the operator \boxdot and \diamond .

$$(\text{Duality}) \models \boxdot\alpha \leftrightarrow \neg\diamond\neg\alpha$$

The two principles of necessity in modal logic **K**, namely (RN) and (K), are also true for *defeasible necessity*. Here is a version of these two axioms:

- Necessitation rule: (RNN) if α is valid, so is $\boxdot\alpha$;
- Distributivity axiom: $(\tilde{\mathbf{K}}) \models \boxdot(\alpha \rightarrow \beta) \rightarrow (\boxdot\alpha \rightarrow \boxdot\beta)$.

The following validity puts defeasible necessity in contrast to classical necessity:

$$(\tilde{\mathbf{N}}) \models \Box\alpha \rightarrow \boxdot\alpha$$

This properties represents the essence of defeasible necessity, as it is able to single the normal states from all the accessible states. This observation concurs with an accepted principle of defeasibility, which is whatever is classically the case is also defeasibly so (such as $\alpha \rightarrow \beta$ entails $\alpha \sim \beta$ in propositional logic, and $C \sqsubseteq D$ entails $C \sqsubseteq_{\sim} D$ in DL). This ties up \boxdot as a defeasible version of \Box and attests to the adequacy of preferential Kripke as an approach to define defeasible modalities. Thanks to $\tilde{\mathbf{N}}$, we can also have a correlation between \diamond and \diamond :

$$(\tilde{\mathbf{N}}) \models \diamond\alpha \rightarrow \diamond\alpha$$

If defeasible necessity \boxdot is a “weaker” version of classical necessity \Box that targets only the normal accessible states instead of all of them, then as a consequence of $\tilde{\mathbf{N}}$, the defeasible possibility \diamond is “stronger” version of classical possibility \Diamond which indicates that a *preferred* accessible state implicitly entails that it is accessible. That is why Britz and Varzinczak [BV18] opted to name \diamond as “distinct possibility”.

It also worth to mention that, with preferential Kripke models, it is possible to define a KLM style defeasible inference. Britz and Varzinczak [BV18] proposed another fragment of defeasible modal that contains defeasible modalities and the defeasible entailment \sim . This fragment is recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \diamond\alpha \mid \boxdot\alpha \mid \diamond\alpha \mid \alpha \sim \alpha$$

We denote the sentence of well formed sentences in this fragment by $\mathcal{L}^{\boxdot, \sim}$. Provided that preferential Kripke models are equipped with a preference relation \prec , we shall interpret \sim as in preferential inferences in the case of propositional logic. Let $\mathcal{J} = (S, R, V, \prec)$ be a preferential Kripke models, we define $\llbracket \alpha \rrbracket^{\mathcal{J}}$ as the set of states in S that satisfy α , i.e., $\llbracket \alpha \rrbracket^{\mathcal{J}} \stackrel{\text{def}}{=} \{s \in S \mid \mathcal{J}, s \models \alpha\}$.

Definition 1.43 (Defeasible entailment). *Let $\mathcal{J} = (S, R, V, \prec)$ be a preferential Kripke model and $s \in S$. For every $\alpha, \beta \in \mathcal{L}^{\mathcal{N}^+}$, we have:*

$$\mathcal{J}, s \models \alpha \sim \beta \text{ if either } s \notin \min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{J}}) \text{ or } s \in \llbracket \beta \rrbracket^{\mathcal{J}}.$$

Definition can be generalized further, we say that \mathcal{J} is a model for a sentence $\alpha \sim \beta$ if the following is true:

$$\mathcal{J} \models \alpha \sim \beta \quad \text{if} \quad \min_{\prec}(\llbracket \alpha \rrbracket^{\mathcal{J}}) \subseteq \llbracket \beta \rrbracket^{\mathcal{J}}.$$

This version of defeasible entailment satisfies the postulates of preferential models [BMV11a].

- (Reflexivity) $\mathcal{J} \models \alpha \sim \alpha$;
- (Left logical equivalence) if $\models \alpha \leftrightarrow \beta$ and $\mathcal{J} \models \alpha \sim \gamma$, then $\mathcal{J} \models \beta \sim \gamma$;
- (Right weakening) if $\mathcal{J} \models \alpha \sim \beta$ and $\models \beta \rightarrow \gamma$, then $\mathcal{J} \models \alpha \sim \gamma$;
- (Cut) if $\mathcal{J} \models \alpha \wedge \beta \sim \gamma$ and $\mathcal{J} \models \alpha \sim \beta$, then $\mathcal{J} \models \alpha \sim \gamma$;
- (Or) if $\mathcal{J} \models \alpha \sim \gamma$ and $\mathcal{J} \models \beta \sim \gamma$, then $\mathcal{J} \models \alpha \vee \beta \sim \gamma$;
- (Cautious Monotony) $\mathcal{J} \models \alpha \sim \beta$ and $\mathcal{J} \models \alpha \sim \gamma$, then $\mathcal{J} \models \alpha \wedge \gamma \sim \beta$.

It is possible to also define a ranked semantics for Kripke models. Ranked Kripke models are preferential Kripke models where the preference relation \prec is *modular* (see Definition 1.23). The defeasible entailment of ranked Kripke models satisfies the postulates of rational systems. Note that since ranked Kripke models are a subclass of preferential Kripke models, this would not affect the semantics of the defeasible modalities \sim and \triangleright .

- (Rational monotony) If $\mathcal{J} \models \alpha \sim \beta$ and $\mathcal{J} \models \neg(\alpha \sim \neg\gamma)$, then $\mathcal{J} \models \alpha \wedge \gamma \sim \beta$.

For a more in depth look on defeasible entailment, we refer the reader to the work of Britz et al. [BMV11a]. The main goal of this section is to show that another aspect of defeasibility, namely the defeasibility of necessity in modal logic.

1.3 Summary

Preferential Kripke models and defeasible modalities offers the reasoner the tools to model and reason on systems that might have exceptions (such is the case in Example 1.42). The operator \sim is a defeasible version of the classical necessity \Box that grants the reasoner the ability to describe properties of the form “the most normal outcome of an action is α ”. We shall discuss the notion of defeasibility in the context of temporal logics, specifically Linear Temporal Logic. We shall base our approach on the work on this study of Britz and Varzinczak [BV18]. We shall show how to use defeasible modalities in order to express new temporal notions. Such is the goal of our study.

Chapter 2

Linear temporal logic

Contents

2.1	Linear temporal logic	33
2.1.1	Syntax and semantics	33
2.1.2	Axomatisation and properties of <i>LTL</i>	36
2.2	Satisfiability of <i>LTL</i> sentences	38
2.2.1	Observations and representations of temporal interpretations	38
2.2.2	<i>LTL</i> fragments	40
2.3	Semantic tableau for <i>LTL</i>	43
2.4	Summary	46

The second element of our study is temporal logic. Temporal logic is a family of formalisms for representing and reasoning about propositions that vary over time. What is the meaning when one says propositions that vary over time? Let us take the statement “It will *eventually* rain” as an example. In natural language, the sentence *eventually rain* entails that at some time in the future, it will rain. The emphasis here is on the time. The truth of these sentences depends on how the weather changes over time. There are other temporal concepts that are used in the natural language (temporal notions are italicized):

- *After* the operation is finished, display the output.
- The system is *always* on.
- Display the loading screen *until* the program is finished loading.

Formalizing and representing temporal notions such as the aforementioned (but not limited to) appeared first in Greek philosophy. It is with the introduction of modal logic and Kripke semantics [Kri59] that researchers in the fields of philosophy and mathematics renew their interest in temporal logic. The first attempt of formalizing temporal notions was proposed by Prior’s Tense Logic [Pri62]. Tense logic is a modal-based of temporal logic that uses modalities to express temporal notions. Examples of which are the modality *G* for *always*, the modality *F* for *eventually*, amongst others. And thanks to Kripke models, Prior showed that it is possible to use the accessibility relation *R* (see Definition 1.38) to indicate the passage of time from one point to another. Temporal logic, thanks to Prior’s Tense logic, has found its niche in modal logic. This gave rise to many temporal formalisms that use modalities in order to express propositions that vary over time and use Kripke-like structures to model time.

In 1977, Pnueli [Pnu77] showed that temporal logic can specify properties of *concurrent programs* (programs in which, during a period of time, multiple processes are being executed). Thanks to the seminal work of Pnueli, temporal logics found their use in the computer science field as formal tools for the logic of programs. In his work, Pnueli [Pnu77] introduced a new formalism called Linear Temporal Logic (*LTL*). In *LTL*, time is viewed as a linear succession of time points that extends to infinity (see Figure 2.1a). Therefore, we can interpret the changes of the states of a system from an initial time point 0 to 1, to 2 and so on \dots , in a linear fashion.

Since the introduction of *LTL*, other temporal formalisms using Kripke-like semantics have been developed over the years. Some of which view time as a tree-like structure such as Figure 2.1b. Time can branch into multiple paths in the future. Each of these paths might be the path on which the program is. There are two notable formalisms that use branching time structures. The first is Computational Tree Logic (*CTL*) by Emerson and Clarke [EC80]. The second is a combination of *LTL* and *CTL*, called *CTL**, and introduced by Emerson and Halpern [EH83].

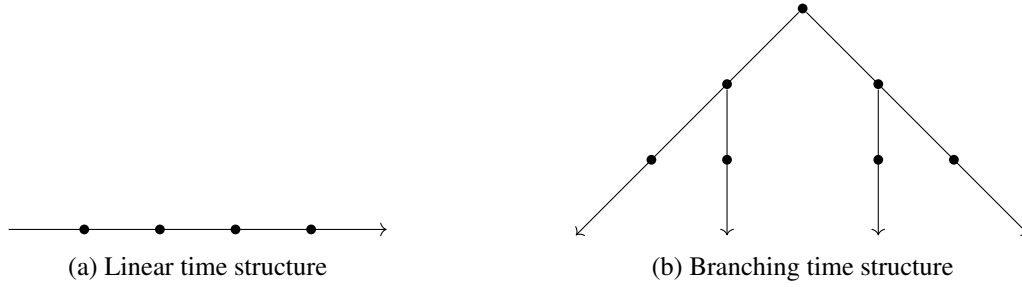


Figure 2.1: Time structures

Temporal logics *LTL*, *CTL* and *CTL** are used as formal tools for the verification and specification of programs. The verification problem, also referred to as model checking [CHV18], is the formalization of whether a given system meets a certain specification (a wanted property for the system). Specifications are expressed using mathematical sentences and a system is interpreted by having an adequate structure that shows its different states, and the relation between them. Specification can be given by temporal logics in order to describe the desired outcome of the system at different points of time. Therefore, these formalisms are suitable for the verification of systems that change states over time. For more in depth look in model checking and temporal logics, we refer the reader to the work of Baier and Katoen [BK08].

However, in our study of defeasibility in temporal logics, we shall focus on the symbolic side of *LTL*. The main goal is to show how defeasibility can be expressed in temporal logic, and motivate its use in this context. There are two reasons behind this choice. The first reason is that *LTL* uses the syntax and semantics of modal logic to express properties that vary over time. Moreover, the syntax and semantics of *LTL* are also present in *CTL* and *CTL**. As such, *LTL* serves as a bridge between defeasibility in modal logic [BV18] and temporal logics. The second reason is the flexibility of *LTL* language and its semantics. Many extensions of *LTL* have been developed in literature over the years, examples of which are: *LTL* with past operators [LPZ85, Gab89, Rey00]. There is also a robust extension *rLTL* proposed by Tabuada and Neider [TN16]. In their setting, a system is robust when small violations of the environment assumptions should lead to small violations of the system guarantees. In particular, the semantics proposed sentences of the form $\alpha \rightarrow \beta$ is such that when the premise α breaks at some time points, the consequence β can be true sometimes, instead of being outright false. There are many other extensions of *LTL* that represents different behaviours of the system. We shall expose, in the upcoming chapters, a defeasible extension of *LTL* [CCACV20], and the computational analysis of the satisfiability problem of this language [CCACV20, CCACV21].

This chapter is split into three parts. In Section 2.1, we discuss the syntax and semantics of *LTL*.

In Section 2.2, we discuss the observations and work of Sistla and Clarke [SC85] on the complexity of different fragments of the *LTL* language. Finally, in Section 2.3, we present briefly a tableau method developed by Reynolds [Rey16a] as a tool for determining the satisfiability of *LTL* sentences.

2.1 Linear temporal logic

Linear Temporal Logic (*LTL*) was introduced by Pnueli [Pnu77] as a formal tool for hardware and software specification and verification. This formalism allows for the description of a program's execution. *LTL* is a modal temporal logic, it uses modalities to refer to time. We can encode sentences that describe the future of an execution, e.g., a statement is *always* true, or, will *eventually* hold.

2.1.1 Syntax and semantics

We highlight the syntax of *LTL* that we use throughout the memoir. Let \mathcal{P} be a finite set of *atomic propositions*. The set of operators in *LTL* can be split into two parts: the set of *Boolean connectives* \neg, \wedge, \vee , and that of *temporal operators* $\Box, \Diamond, \bigcirc, \mathcal{U}$, where \Box reads as *always*, \Diamond as *eventually*, \bigcirc as *next* and \mathcal{U} as *until*. The set of well-formed sentences expressed in *LTL* is denoted by \mathcal{L} . Sentences of \mathcal{L} are built up according to the following grammar:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha \mid \bigcirc\alpha \mid \alpha \mathcal{U} \alpha$$

Other standard Boolean operators are part of the syntax of *LTL*. Let α, β be two sentences of *LTL* language, the symbol \top is an abbreviation of $\alpha \vee \neg\alpha$, \perp is an abbreviation of $\alpha \wedge \neg\alpha$. The implication operator is defined by $\alpha \rightarrow \beta \stackrel{\text{def}}{=} \neg\alpha \vee \beta$ and the equivalence operator is defined by $\alpha \leftrightarrow \beta \stackrel{\text{def}}{=} (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. Here are some examples of well-formed sentences in \mathcal{L} .

Example 2.1. Let $p, q \in \mathcal{P}$:

$$p, \quad \neg p, \quad \Diamond(p \wedge q) \rightarrow \Diamond p, \quad \Diamond\Box p \leftrightarrow q, \quad \neg\Diamond p \wedge \Box\neg q$$

The negation and temporal operators have higher precedence than the other operators. As such, the sentence $\neg\Diamond p \wedge \Box\neg q$ is the same as $(\neg\Diamond p) \wedge (\Box\neg q)$ and not $\neg(\Diamond p \wedge \Box\neg p)$.

The temporal structure is a chronological linear and infinite succession of time points. In this frame, the state of a system switches from a state s_0 to s_1 , from s_1 to s_2 , and so on (the emphasis here is that there is one and only one transition from t to $t + 1$).

$$s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots$$

LTL interpretations are Kripke structures $\mathcal{K} = (S, R, V)$ of which the accessibility relation R is *reflexive*, *transitive*, *linear* and *total* relation. We discuss them at length in Section 2.1.2. Here we set the notations and the structures we shall use throughout this memoir. We use the set of natural numbers in order to label each of time points, i.e., $(\mathbb{N}, <)$. A *temporal interpretation* I is defined by a mapping function $V : \mathbb{N} \rightarrow 2^{\mathcal{P}}$ which associates each time point $t \in \mathbb{N}$ of the temporal structure with a set of propositional atoms $V(t)$ that are true in t (atoms not belonging to $V(t)$ are assumed to be false at the given time point). The truth conditions of *LTL* sentences are recursively defined as follows, where I is a temporal interpretation and t a time point in \mathbb{N} :

- $I, t \models p$ if $p \in V(t)$;
- $I, t \models \neg\alpha$ if $I, t \not\models \alpha$;

- $I, t \models \alpha \wedge \alpha'$ if $I, t \models \alpha$ and $I, t \models \alpha'$;
- $I, t \models \alpha \vee \alpha'$ if $I, t \models \alpha$ or $I, t \models \alpha'$;
- $I, t \models \Box \alpha$ if $I, t' \models \alpha$ for all $t' \in \mathbb{N}$ s.t. $t' \geq t$;
- $I, t \models \Diamond \alpha$ if $I, t' \models \alpha$ for some $t' \in \mathbb{N}$ s.t. $t' \geq t$;
- $I, t \models \bigcirc \alpha$ if $I, t + 1 \models \alpha$;
- $I, t \models \alpha \mathcal{U} \alpha'$ if $I, t' \models \alpha'$ for some $t' \geq t$ and for all $t \leq t'' < t'$ we have $I, t'' \models \alpha$.

Figure 2.2 contains a graphical representation of the truth values of temporal sentences in *LTL*.

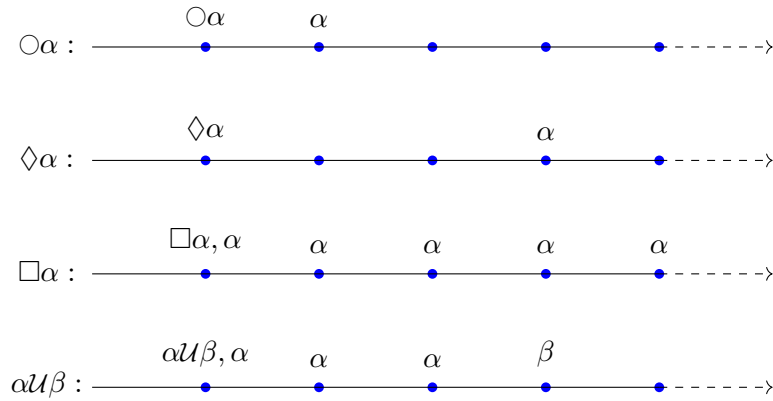
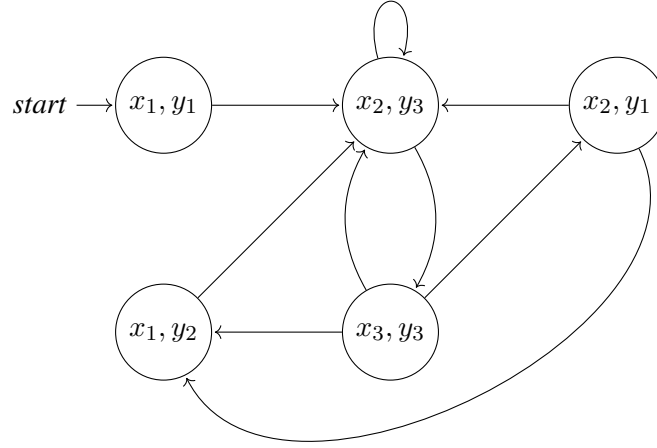


Figure 2.2: Graphical representation of *LTL* sentences

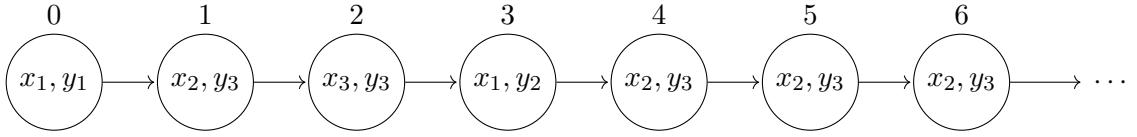
We say $\alpha \in \mathcal{L}$ is *satisfiable* if there are I and $t \in \mathbb{N}$ such that $I, t \models \alpha$. A sentence α is *valid* if for all temporal interpretations I and all $t \in \mathbb{N}$, we have $I, t \models \alpha$. A valid sentence $\alpha \in \mathcal{L}$ is denoted by $\models \alpha$. We shall discuss the satisfiability of sentences of \mathcal{L} in depth in the upcoming two sections.

We introduce an example to better illustrate the use of the language of *LTL*. We shall also use this example to motivate the defeasible extension of *LTL* in Chapter 3.

Example 2.2 (Transition system of two variables). *We have a computer program in which the values of its variables change with time. In particular, the agent wants to check two parameters, say x and y . These two variables take one and only one value between 1 and 3 on each iteration of the program. We represent the set of atomic propositions by $\mathcal{P} = \{x_1, x_2, x_3, y_1, y_2, y_3\}$ where x_i (resp. y_i) for all $i \in \{1, 2, 3\}$ is true if the variable x (resp. y) has the value i . The system has five states. The following figure represents the different transitions between these states:*



The initial state of the system is $\{x_1, y_1\}$ (both of the variables are set to 1). Transitions from a state s_i to s_j are represented by an edge $s_i \rightarrow s_j$. There are many possible runs of this system. Starting from the initial state and following the transitions, we could have this run:



For $t > 4$, $V(t) = V(4) = \{x_2, y_3\}$.

The corresponding interpretation to this run is denoted by I_1 . In the current run, the program assigns the value 3 to y whenever $x = 2$. The interpretation satisfies the sentence $\Box(x_2 \rightarrow y_3)$ at 0. Moreover, there are not any time point such that $y = 1$ when $x = 2$. Therefore, I_1 does not satisfy $\Diamond(x_2 \wedge y_1)$ at time 0. Hence, we have:

- $I_1, 0 \models \Box(x_2 \rightarrow y_3)$;
- $I_1, 0 \models \neg\Diamond(x_2 \wedge y_1)$.

Temporal interpretations are used to model the changes of a program over time. We use the term *run*, *path* or *trace* to depict a possible sequence of an execution of a program (these sequences are represented by a temporal interpretation). In a run, we are able to capture specifications (desired outcomes) by expressing sentences about the current and future time points. Here are some specifications that can be expressed in *LTL*.

- Safety: $\Box\alpha$ means that the property α will always hold, from this point of the execution onwards.
- Liveness: $\Diamond\alpha$ means that the property α will hold eventually. In other words, at some time point of the run, α is true.
- Response: $\Box\Diamond\alpha$ means that for any time point in the run there is a later time point where α holds.
- Persistence: $\Diamond\Box\alpha$ means that there exists a time point in the run such that from then and onwards, α holds.

2.1.2 Axomatisation and properties of LTL

One of the area of interest in modal logics is the axiomatic theory. Indeed, different modal logics can be obtained by placing restrictions or constraints on the accessibility relation R (see Definition 1.38). These restrictions are characterized by validities (using modalities) that represent interpretations with said restrictions. For more in depth look on the automatisation in modal logics, we refer the reader to the work of Blackburn and van Benthem [BvB07]. In this section, we lay down the properties of LTL 's temporal modalities.

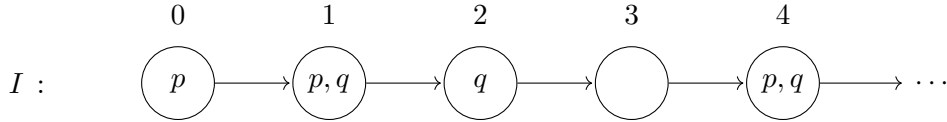


Figure 2.3: LTL interpretation

It is possible to model linear temporal structures using Kripke semantics. Let $\mathcal{K} = (S, R, V)$ be a Kripke structure that we shall use to model linear temporal structures. Time points in a linear temporal structure, such as the interpretation I in Figure 2.3, are represented by states in S . Moreover, the accessibility relation R can be used to show the succession of states. The valuation function V maps each state $s \in S$ (which indicate a time point in temporal structures) to a set of true valued atomic propositions. We shall assume, from this point on, that all the sentences in this section are well-formed sentences in LTL . The following validities are true in LTL . First, the axioms of modal logic **K** are also true for LTL interpretations.

$$(Duality) \models \Box\alpha \leftrightarrow \neg\Diamond\neg\alpha;$$

$$(RN) \text{ if } \alpha \text{ is valid, so is } \Box\alpha;$$

$$(K) \models \Box(\alpha \rightarrow \beta) \rightarrow (\Box\alpha \rightarrow \Box\beta).$$

Beside the characterization of logic **K**, the linear temporal structure is a particular case of normal Kripke structure. As the accessibility relation R models a linear passage of time, it is a more restrictive form of the normal case. LTL interpretations are Kripke structures that have the following properties: *reflexive, transitive, linear and total*.

To represent linear temporal interpretations, Kripke structures are required to be reflexive. Let take the sentence $\Diamond\alpha$, whose intuitive meaning is *eventually* α is true. There must be a time point in the future where α is true. Said time point could be the present moment. Therefore, each time points can be reachable to themselves. Reflexive Kripke structures, including LTL interpretations, are defined by the axiom (T).

$$(T) \models \Box\alpha \rightarrow \alpha, \quad \models \alpha \rightarrow \Diamond\alpha.$$

LTL interpretations are also transitive. Consider three time point t_1, t_2, t_3 such that t_3 comes after t_2 that comes after t_1 . Consider that the sentence $\Diamond\alpha$ is true in t_2 because α is true in t_3 . It would be intuitive, from a language perspective, to entail that $\Diamond\alpha$ is true in t_1 because t_3 has α and it can be used to prove it. A Transitive Kripke structures, including LTL interpretations, are defined by the axioms (4).

$$(4) \models \Box\alpha \rightarrow \Box\Box\alpha, \quad \models \Diamond\Diamond\alpha \rightarrow \Diamond\alpha.$$

Combining axiom (T) and (4), we obtain the following validities.

$$\models \Box\Box\alpha \leftrightarrow \Box\alpha, \quad \models \Diamond\Diamond\alpha \leftrightarrow \Diamond\alpha.$$

In addition, *LTTL* interpretations are *total*. Knowing that in the case *LTTL*, time is seen as an infinite succession of time points. Thus, any given time point t has successors that come after it. Moreover, as time is seen as *linear* sequence of time points, then for each time point t , there is at most one time point $t + 1$ that is considered as its *next* step. From the linear and total properties, we can infer that for each time point t , there is exactly one time point $t + 1$ that comes after it. The total property does not hold for some extensions of *LTTL*, such is the case of *LTTL* on finite traces [GMM14]. However, in this study, we focus on the general definition of *LTTL* interpretation, which are by design, infinite. The operator \bigcirc is used to define the linearity and totality of *LTTL* interpretations.

$$(\text{Linearity and totality}) \models \bigcirc\alpha \leftrightarrow \neg\bigcirc\neg\alpha.$$

Since there is exactly one time point $t + 1$ that comes after any given time point t . It would be strange that such time point would contain a sentence α and its negation $\neg\alpha$. Given these characteristics, the accessibility can be inferred without being explicitly present in *LTTL* interpretations. That is why *LTTL* interpretations are either defined as $I \stackrel{\text{def}}{=} (S, V)$ with S being a ordered sequence of time points $S \stackrel{\text{def}}{=} \{s_0, s_1, s_2, \dots\}$, or it is possible to use the temporal structure $(\mathbb{N}, <)$ in order to represent it (the current representation that we are using in this study). Here are some additional properties of *LTTL* interpretations.

$$\models \Box\alpha \rightarrow \Diamond\alpha.$$

Given a time point t , the sentence $\Box\alpha$ is true only when α is true for all of futures of t . This implies that there is a time point in the future of t that satisfies α , and therefore $\Diamond\alpha$ is true in t .

Next, we discuss the collapsing properties of the temporal modalities \Box and \Diamond .

$$\models \Box\Box\alpha \leftrightarrow \Box\alpha;$$

$$\models \Diamond\Diamond\alpha \leftrightarrow \Diamond\alpha;$$

$$\models \Box\Diamond\Box\alpha \leftrightarrow \Diamond\Box\alpha;$$

$$\models \Diamond\Box\Diamond\alpha \leftrightarrow \Box\Diamond\alpha.$$

In sentences without the \bigcirc, \mathcal{U} operators, no more than two operators need to appear in sequence. Any sequence of three or more \Box, \Diamond collapses into at most a pair of these operators.

$$\models \Box\alpha \leftrightarrow \alpha \wedge \bigcirc\Box\alpha;$$

$$\models \Diamond\alpha \leftrightarrow \alpha \vee \bigcirc\Diamond\alpha;$$

$$\models \alpha \mathcal{U} \beta \leftrightarrow \beta \vee (\alpha \wedge (\bigcirc(\alpha \mathcal{U} \beta))).$$

These are the inductive form of temporal operator in *LTTL*. These validities can be understood by reading them using natural language. For a sentence α to be *always* true, α needs to be true at this moment, and it must be always true in the next step. For a sentence α to be *eventually* true, it is either true in the current moment or it is eventually true in the next step. We leave it to the reader to interpret the *until* statement. The operator \bigcirc plays a crucial role in algorithms and methods for deciding the satisfiability of sentences in *LTTL* (more on that in Section 2.3), but it is rarely used to express specifications of programs. In general, the correctness of specifications is dictated by the entire run rather than interleaving time points within said run. Therefore, they are usually expressed with operators such as $\Box, \Diamond, \mathcal{U}$.

For additional properties of *LTTL* interpretation and its operators, we refer the reader to the work of Ben-Ari [BA12b].

2.2 Satisfiability of LTL sentences

We shall discuss the problem of the satisfiability of LTL sentences. This problem states the following: “given a sentence $\alpha \in \mathcal{L}$, is there an \mathcal{L} interpretation I that satisfies it, i.e., there is a $t \in \mathbb{N}$ such that $I, t \models \alpha$ ”. Without loss of generality, we look for interpretations that satisfy α at the initial time point 0. The first task is to find whether this problem is *decidable*, meaning we have an answer for this problem (either yes or no) for any well-formed \mathcal{L} sentence. The second task is to establish the computational analysis for the satisfiability problem. Such work was done by Sistla and Clarke [SC85]. The particularity of LTL interpretations is that they are infinite in their size. Sistla and Clarke show through a number of observations that it is possible to have compact representations of interpretations, that are bounded in size and are helpful to determine the satisfiability of sentences. As we shall see in the contribution part, we use similar observations and structures in order to find compact representations for defeasible LTL interpretations [CCACV20]. In this section, we shall discuss, in more details, the fragmentation of the \mathcal{L} language and the compact representations in each of these cases.

2.2.1 Observations and representations of temporal interpretations

Let \mathcal{P} be a set of finite atomic propositions. An interesting observation made by Sistla and Clarke [SC85] is that in every LTL interpretation, there is a time point t after which every t -successor’s valuation occurs infinitely many times. This is an obvious consequence of having an infinite set of time points and a finite number of possible valuations (thanks to \mathcal{P} being finite).

Lemma 2.3 ([SC85]). *Let I be an LTL interpretation. There exists a $t \in \mathbb{N}$ s.t. for all $l \in [t, \infty[$, there is a $k > l$ where $V(l) = V(k)$.*

For each interpretation I , there is a first time point where the condition in Lemma 2.3 is met. Such time point is denoted by t_I . The interval of time points of the interpretation that comes before t_I is called the *initial* part of I . Time points that come after t_I are in the *final* part of I .

Definition 2.4. *Let I be an LTL interpretation. Sistla and Clarke define:*

- $init(I) \stackrel{\text{def}}{=} [0, t_I[$;
- $final(I) \stackrel{\text{def}}{=} [t_I, \infty[$;
- $range(I) \stackrel{\text{def}}{=} \{V(i) \mid i \in final(I)\}$;
- $val(I) \stackrel{\text{def}}{=} \{V(i) \mid i \in \mathbb{N}\}$;
- $size(I) \stackrel{\text{def}}{=} length(init(I)) + card(range(I))$, where $length(\cdot)$ denotes the length of a sequence and $card(\cdot)$ set cardinality.

Note that for any interpretation, the initial part is always a finite sequence of time points. Moreover, the final part is an infinite sequence of time points. In the size of I , we count the number of time points in the initial part and the number of valuations contained in the final part. As we shall see in Section 2.2.2, the order on which time points are in the final part does not matter in some fragments of the LTL language.

In order to find the upper-bound of interpretations on different fragments of the language. Sistla and Clarke introduced intermediary structures. First of which are subsequences. The general idea is that starting from an interpretation I that satisfies a sentence, it is possible to find subsequence of time points within \mathbb{N} that still satisfies the sentence α . Formally, they are defined as follows:

Definition 2.5 (Sub-sequence). *Let N, N' be two sequences of natural number. N' is a subsequence of N (written as $N' \subseteq N$) iff for all $i \in N'$, we have $i \in N$.*

We shall introduce the notion of pseudo-interpretations next. Given an interpretation I and sequence $N \subseteq \mathbb{N}$, a pseudo-interpretation I^N is the restriction of the valuation of I to only time points that are in N . The pseudo-interpretation has a valuation function $V^N : N \rightarrow 2^{\mathcal{P}}$ such that for all $t \in N$, we have $V^N(t) = V(t)$. With pseudo-interpretations, we can check the truth values of sentences within sequences of the starting interpretation I . The truth values of \mathcal{L} sentences in pseudo-interpretations are defined in a similar fashion as for LTL interpretations. With $\models_{\mathcal{P}}$ we denote the truth values of sentences in a pseudo-interpretation.

- $I^N, t \models_{\mathcal{P}} p$ if $p \in V^N(t)$;
- $I^N, t \models_{\mathcal{P}} \neg\alpha$ if $I^N, t \not\models_{\mathcal{P}} \alpha$;
- $I^N, t \models_{\mathcal{P}} \alpha \wedge \beta$ if $I^N, t \models_{\mathcal{P}} \alpha$ and $I^N, t \models_{\mathcal{P}} \beta$;
- $I^N, t \models_{\mathcal{P}} \alpha \vee \beta$ if $I^N, t \models_{\mathcal{P}} \alpha$ or $I^N, t \models_{\mathcal{P}} \beta$;
- $I^N, t \models_{\mathcal{P}} \Box\alpha$ if $I^N, t' \models_{\mathcal{P}} \alpha$ for all $t' \in N$ s.t. $t' \geq t$;
- $I^N, t \models_{\mathcal{P}} \Diamond\alpha$ if $I^N, t' \models_{\mathcal{P}} \alpha$ for some $t' \in N$ s.t. $t' \geq t$;
- $I^N, t \models_{\mathcal{P}} \bigcirc\alpha$ if we have $t+1 \in N$ and $I^N, t+1 \models_{\mathcal{P}} \alpha$;
- $I^N, t \models_{\mathcal{P}} \alpha\mathcal{U}\beta$ if there is $t_2 \in N$ s.t. $I^N, t_2 \models_{\mathcal{P}} \beta$ and $I^N, t_1 \models_{\mathcal{P}} \alpha$ for all $t_1 \in N$ s.t. $t \leq t_1 < t_2$.

A sequence represents a set of time points. They can either be in a continuous interval of integers, a discontinuous sequence or a combination of both. Sequences can also be finite or infinite. When a sequence N is finite, the pseudo-interpretation can be considered to not have a final part. In this case $size(I^N) \stackrel{\text{def}}{=} length(N)$. When a sequence N is infinite, the condition of Lemma 2.3 is met. Therefore, a t_{I^N} can be defined. Therefore, $init(I^N)$, $final(I^N)$ and $size(I^N)$ can be introduced as well. However, given some cases, time points that were initially in $init(I)$ can be found in $final(I^N)$ and vice-versa. As such, given the definition of $size(\cdot)$, it is possible to find sequences $N \subseteq \mathbb{N}$ such that $size(I^N) \geq size(I)$. Sistla and Clarke [SC85] introduced sequences that display a certain behaviour called *acceptable sequences* that solves this problem.

Definition 2.6 (Acceptable sequence w.r.t. I). *Let I be an interpretation and N be a sequence of temporal time points. N is an acceptable sequence w.r.t. I iff for all $i \in N \cap final(I)$ and for all $j \in final(I)$ s.t. $V(i) = V(j)$, we have $j \in N$.*

The particularity we are looking for is that any picked time point in $init(\cdot)$ (resp. $final(\cdot)$) will remain in the initial (resp. final) part of the new pseudo-interpretation. It is worth pointing out that an acceptable sequence w.r.t. an interpretation also can be either finite or infinite.

Definition 2.7. *Let I be an interpretation, and let N be an acceptable sequence w.r.t. I . We define the following:*

$$\begin{aligned}
 init(I, N) &\stackrel{\text{def}}{=} N \cap init(I); \\
 final(I, N) &\stackrel{\text{def}}{=} N \setminus init(I, N); \\
 range(I, N) &\stackrel{\text{def}}{=} \{V(t) \mid t \in final(I, N)\}; \\
 val(I, N) &\stackrel{\text{def}}{=} \{V(t) \mid t \in N\}; \\
 size(I, N) &\stackrel{\text{def}}{=} length(init(I, N)) + card(range(I, N)).
 \end{aligned}$$

Thanks to Definition 2.7, the following properties are true:

Proposition 2.8 ([SC85]). *Given an interpretation I and an acceptable N sequence w.r.t. I . We have:*

$$\text{size}(I, N) \leq \text{size}(I)$$

Let N_1, N_2 be two sequences of integers. The union of N_1 and N_2 , denoted by $N_1 \cup N_2$, is the sequence containing only and all elements of N_1 and N_2 .

Proposition 2.9 ([SC85]). *Let I be an interpretation, N_1, N_2 be two acceptable sequences w.r.t. I . Then $N_1 \cup N_2$ is an acceptable sequence w.r.t. I s.t. $\text{size}(I, N_1 \cup N_2) \leq \text{size}(I, N_1) + \text{size}(I, N_2)$.*

Sistla and Clarke defined also a class of *LTL* interpretations called ultimately periodic interpretations. These interpretations have an initial part which is finite and a final part that periodically replicates following time points that are in the interval of the first period. Figure 2.4 is a graphical representation of ultimately periodic

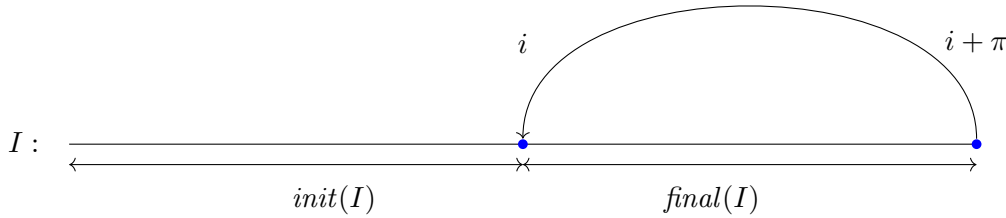


Figure 2.4: Ultimately-periodic interpretation

In Figure 2.4, all time points between i and $i + \pi - 1$ are replicated indefinitely. The time point $i + \pi$ has the same valuation as i , the time point $i + \pi + 1$ has the same valuation as $i + 1$, and so on. Formally, ultimately-periodic interpretations are defined as follows:

Definition 2.10 (Ultimately periodic interpretations). *An *LTL* interpretation I with a valuation V is said to be ultimately periodic if after a starting time point $i \geq 0$, there is a period $\pi > 0$ such that for all $k \geq i$ $V(k) = V(k + \pi)$.*

Ultimately periodic-interpretations are a compact representations of *LTL* interpretations. Sistla and Clarke showed that from an interpretation I that satisfies α , and depending on the fragment of the \mathcal{L} language, an ultimately-periodic interpretation I' that satisfies the same sentence α can be induced. We discuss these fragments in the second part of this Section.

Acceptable sequences and ultimately periodic interpretations are useful tools for finding bounded representations. We shall extend these notions more in the contribution part of this memoir.

2.2.2 *LTL* fragments

With the structures introduced, the next step is to highlight the complexity analysis of the satisfiability problem of different fragments of the *LTL* language. In their work, Sistla and Clarke introduced three fragments: The first where the only temporal operators allowed are \Box, \Diamond (denoted $L(\Diamond)$). The second fragment contains only \Diamond, \bigcirc and sentences are in normal negation form NNF (negation is only allowed on atomic proposition level). This fragment is denoted by $L_{NNF}(\Diamond, \bigcirc)$. The third fragment is the entire language $L(\mathcal{U}, \bigcirc)$, which covers the entire \mathcal{L} language. For each of these fragments, we discuss the steps to obtain the bounded model property and the complexity of their respective satisfiability problem. The

goal of this section is to show Sistla and Clarke's [SC85] approach to the satisfiability problem for LTL , specially, their fragmentation of the language and the methods used to prove an upper-bound for LTL interpretations in each of these cases. As we shall see in the contribution part, many fragmentations and methods for checking the satisfiability problem of the defeasible extension of LTL are based on their approach.

Given a sentence $\alpha \in \mathcal{L}$, we shall use the notation $|\alpha|$ to denote the length of the sentence, i.e., the number of symbols that are in α .

The satisfiability of $L(\Diamond)$ sentences

The fragment $L(\Diamond)$ is recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha$$

Let $\alpha \in L(\Diamond)$ be a satisfiable sentence in LTL . Then, there exists an interpretation I such that $I, 0 \models \alpha$. Not much is known about this initial interpretation I and its size $size(I)$. Sistla and Clarke [SC85] state that from I , we can find an acceptable sequence N such that $I^N \models \alpha$ and $size(I, N) \leq |\alpha|$ that also satisfies α . From I^N , an induced LTL interpretation I' has the same size as I^N and keeps the satisfiability of α .

As discussed in the previous Section 2.2.1, the order of time in the final part does not matter. Such is the case for interpretation of $L(\Diamond)$ sentences.

Lemma 2.11. *Let I be an interpretation and $i \leq i'$ be time points of $final(I)$ where $V(i) = V(i')$. Then for every $\alpha \in L(\Diamond)$, we have $I, i \models \alpha$ iff $I, i' \models \alpha$.*

Theorem 2.12 (Bounded model property). *Let $\alpha \in L(\Diamond)$ be a satisfiable sentence. There exists an interpretation I such that $I, 0 \models \alpha$ and $size(I) \leq |\alpha|$.*

Sistla and Clarke showed that the satisfiability of $L(\Diamond)$ sentences is a NP-hard problem by reducing it to a 3-SAT problem [SC85]. With the bounded model property obtained, a non-deterministic Turing machine guesses an interpretation I such that the sum of elements in $init(I)$ and distinct valuations in $range(I)$ is less than $|\alpha|$. Sistla and Clarke defined a labelling function that check the satisfiability of α -subsentences in I . The Turing machine accepts I as an interpretation for α if α is true in 0. Otherwise, the interpretation I is rejected. Sistla and Clarke showed that the verification procedure is polynomial-time bounded [SC85].

Theorem 2.13 ([SC85]). *The satisfiability of $L(\Diamond)$ sentences is NP-complete.*

The satisfiability of $L_{NNF}(\Diamond, \bigcirc)$ sentences

The negation operator \neg in this fragment is only allowed within the scope of atomic propositions. The fragment $L_{NNF}(\Diamond, \bigcirc)$ is recursively defined as follows:

$$\alpha ::= p \mid \neg p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Diamond\alpha \mid \bigcirc\alpha$$

It turns out that in the absence of the \Box operator and for a specific time point t that satisfies α , there is a sequence of time points that keeps the satisfiability of α in t and is always *finite*. As such, acceptable sequence are not needed in this case.

Theorem 2.14 (Bounded model property). *Let $\alpha \in L_{NNF}(\Diamond, \bigcirc)$ be a satisfiable sentence. There exists an interpretation I such that $I, 0 \models \alpha$ and $size(I) \leq |\alpha|$.*

The NP-completeness of the satisfiability of $L_{NNF}(\Diamond, \bigcirc)$ is similar to the fragment $L(\Diamond)$. The only difference is that the Turing machine guesses a finite sequence which size is less than $|\alpha|$. The labelling function operates the same way.

Theorem 2.15 ([SC85]). *The satisfiability of $L_{NNF}(\Diamond, \bigcirc)$ sentences is NP-complete.*

The satisfiability of $L(\mathcal{U}, \bigcirc)$ sentences

The third fragment is $L(\mathcal{U}, \bigcirc)$. It is worth to point that, the following validity $\models \Diamond\alpha \leftrightarrow \top\mathcal{U}\alpha$ is true in LTL . Therefore sentences in \mathcal{L} can be expressed using the operators of this fragment. The fragment $L(\mathcal{U}, \bigcirc)$ can be recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \bigcirc\alpha \mid \alpha\mathcal{U}\alpha$$

In the presence of \bigcirc, \mathcal{U} operators, the Lemma 2.11 does not hold. The order of time points in the final part of an interpretation does matter in this case and therefore the approach of acceptable sequences is not helpful in this case. Sistla and Clarke [SC85] provide another method to find the upper-bound for LTL interpretations. Let $\alpha \in \mathcal{L}$ be a satisfiable sentence and I be an interpretation such that $I, 0 \models \alpha$. We shall see that starting from the I , an ultimately periodic interpretation I' can be induced such that $I', 0 \models \alpha$.

Let $I, i \models \alpha_1\mathcal{U}\alpha_2$ be an interpretation that satisfies $\alpha_1\mathcal{U}\alpha_2$ in i . We say that $\alpha_1\mathcal{U}\alpha_2$ in i is fulfilled before j if there is $i \leq k \leq j$ such that $I, k \models \alpha_2$. The key is to find two time points $i, i + \pi$ in the initial interpretation I that satisfy the same set α -subsentences such that all until sentences i are fulfilled before $i + \pi$. Sistla and Clarke [SC85] showed that by keeping time points in $[0, i[$ as the initial interpretation I , followed by replicating time points in the interval $[i, i + \pi[$, the sentence α remains satisfied in 0 on this induced ultimately-periodic interpretation.

Theorem 2.16 (Bounded model property). *A sentence $\alpha \in L(\mathcal{U}, \bigcirc)$ is satisfiable if there is an ultimately periodic interpretation I such that the starting time point is $i \leq 2^{|\alpha|}$, the period is $\pi \leq 4^{|\alpha|}$ and $I, 0 \models \alpha$.*

Theorem 2.17 ([SC85]). *The satisfiability of $L(\mathcal{U}, \bigcirc)$ sentences is PSPACE-complete.*

The satisfiability in the fragments $L(\mathcal{U})$, $L(\Diamond, \bigcirc)$ are in the same complexity class as the fragment $L(\mathcal{U}, \bigcirc)$. A detailed version of the theorems from 2.12 to 2.17 as well as Sistla and Clarke's approach can be found in [SC85]. Table 2.1 recapitulates the complexity of the satisfiability problem of multiple LTL fragments. We also refer the reader to work of Demri and Rabinovich [DR10] for another outlook on the complexity study of the satisfiability of LTL sentences.

Fragment	Satisfiability
$L(\Diamond)$	NP-complete
$L_{NNF}(\Diamond, \bigcirc)$	NP-complete
$L(\Diamond, \bigcirc)$	PSPACE-complete
$L(\mathcal{U})$	PSPACE-complete
$L(\mathcal{U}, \bigcirc)$	PSPACE-complete

Table 2.1: Complexity of some LTL fragments

2.3 Semantic tableau for LTL

A tableau approach for a logic is a decision procedure for checking the satisfiability of its sentences. These approaches trace their origin to the semantic tableaux for propositional logic, developed by Smullyan [Smu68]. Classical tableaux typically have a tree-shaped form, with the root node containing the initial sentence. Moreover, tableaux methods are governed by a set of semantics rules that determine their expansion. At any point of the tableau's expansion, a parent node can branch out to zero or multiple children nodes. An expanded tableau that has a successful branch entails that the initial sentence is satisfiable while failing to have successful branches means that the sentence is not satisfiable. Many logics, specially modal logic \mathbf{K} (Goré [Gor99]), implements tree-shaped tableaux methods for checking the satisfiability of its sentences.

However, in the case of LTL , variations on the traditional tableau idea have been prevalent and the tree-shaped form tableaux for LTL sentences was scraped away. The first implementation was proposed by Wolper [Wol85]. Starting from a root node with an LTL sentence and following only the semantic rules, branch may go indefinitely. That is why in Wolper's method, nodes with same set of sentences are linked with each other. Therefore, the edges of the tableau generally end up heading upwards or crossing branches. After the full expansion phase, there is a pruning phase which eliminates nodes according to another set of rules. This style of tableaux is considered to be *graph-shaped* (edges are cross-branching) and *multi-pass* (the expansion phase and the pruning phase). Over the years, graph-shaped tableaux for LTL have been optimized further [Gou89, SGL97, RV07]. For a more in depth look on this method, we refer the reader to the work of Goré [Gor99].

Although it is the most known approach for LTL tableaux, implementing a graph like tableau for a preferential extension of LTL proved to be quite difficult. In the literature, logics extended with KLM's preferential semantics have tree-shaped tableaux. Examples of which are Giordano et al. [GGOP09] for propositional logic, Giordano et al. [GOGP09], Britz and Varzinczak [BV19] for description logic and Britz and Varzinczak [BV18] for modal logic. That is why we turned our focus on tree-shaped tableaux for LTL in order to bridge the gap between tableaux with preferential semantics and LTL tableaux. This was first done by Schwendimann [Sch98]. Schwendimann's tableau is not only tree-shaped, it is also *one-pass*, not relying on a two-phase procedure: expanding and pruning (they are done simultaneously). Recently, Reynolds proposed another version of tree-shaped one-pass LTL that resembles traditional tableaux. Reynolds's approach does not need uplink connection between nodes (the direction of edges follow the direction of root to leaf nodes). Such is the case for the proposed methods of logics extended with the preferential semantics of KLM.

Many researchers renew their interest for the one-pass tree-shaped style of tableaux. The noteworthy method is the tableau for LTL with past operators [GMR17]. There are also some interesting development for CTL tableaux [ABHL20] using a similar approach as Reynolds's. In this section, we highlight Reynolds's tableaux for LTL sentences that serves as a basis for a tableau method for our defeasible extension of LTL [CCACV21].

Definition 2.18 (Labelled node). *A labelled node is a triple of the form $n : \Gamma$ where $n \in \mathbb{N}$ and $\Gamma \subseteq \mathcal{L}$.*

The integer n is a label for the temporal state of the node and Γ is the set of sentences that are true in the node. Note that multiple nodes can have the same label n . For any sentence $\alpha \in \mathcal{L}$, the *root* is the labelled node $0 : \{\alpha\}$. A node can also have zero to multiple children nodes. In the case of a node having no children, we say that it is a *leaf node*.

Definition 2.19 (Branch). *A branch is a sequence $B \stackrel{\text{def}}{=} \langle x_0, x_1, x_2, \dots \rangle$ of labelled nodes $x_i := n_i : \Gamma_i$.*

Branches are sequences of labelled nodes. Each branch represents a potential model for the starting sentence of the tableau. We shall say that a branch B is *successful* if it has a leaf node with a check mark

(✓). Otherwise, a branch is *unsuccessful* if it has a leaf node with a cross (✗).

A tree is a set of finite branches $\mathcal{T} \stackrel{\text{def}}{=} \{B_0, B_1, B_2, B_3, \dots, B_k\}$ where $k \geq 0$. A tableau \mathcal{T} for α is the limit of a sequence of trees $\langle \mathcal{T}^0, \mathcal{T}^1, \mathcal{T}^2, \dots \rangle$ where the initial tree is $\mathcal{T}^0 := \{(\langle 0 : \{\alpha\} \rangle)\}$ and every \mathcal{T}^{i+1} is obtained from \mathcal{T}^i by applying a rule on one of its branches. We say that a tableau \mathcal{T} for α is *saturated* if no more rules can be applied after a tree \mathcal{T} .

We have two types of rules, static and dynamic rules. We introduce static rules first. Let \mathcal{T} be a tree, and let B be a branch of \mathcal{T} that has a leaf $n : \Gamma$. We say that a static rule (ρ) is applicable at the leaf $n : \Gamma$ if a sentence in Γ instantiates the pattern ρ . Static rules application is not deterministic. This means when multiple patterns emerge in a node with a label n , there is no static rule that takes precedence over an other. Eventually every static rule that can be applied on node with the label n will eventually be applied. A static rule is a rule of the form:

$$(\rho) \frac{n : \Gamma}{n : \Gamma_1 \mid \dots \mid n : \Gamma_k}$$

The symbol ‘|’ indicates the occurrence of a split in the branch, i.e., a non-deterministic choice of possible outcomes, each of which needs to be explored. It is worth to mention that after applying a static rule on $n : \Gamma$, the leaf nodes of all the new branches keep the same label n .

In what follows, we show the rules for Boolean and temporal operators. We also show two stopping conditions, namely, Empty and Contradiction. The symbol \cup is the union of two sets. The symbol \uplus represents the union between disjoint sets.

$$\begin{array}{ll} \text{(Contradiction)} \frac{n : \{\alpha, \neg\alpha\} \uplus \Sigma}{(\text{✗})} & \text{(Empty)} \frac{n : \emptyset}{(\text{✓})} \\ \\ (\wedge) \frac{n : \{\alpha_1 \wedge \alpha_2\} \uplus \Sigma}{n : \{\alpha_1, \alpha_2\} \cup \Sigma} & (\vee) \frac{n : \{\alpha_1 \vee \alpha_2\} \uplus \Sigma}{n : \{\alpha_1\} \cup \Sigma \mid n : \{\alpha_2\} \cup \Sigma} \\ \\ (\Box) \frac{n : \{\Box\alpha_1\} \uplus \Sigma}{n : \{\alpha_1, \Box\alpha_1\} \cup \Sigma} & (\Diamond) \frac{n : \{\Diamond\alpha_1\} \uplus \Sigma}{n : \{\alpha_1\} \cup \Sigma \mid n : \{\Diamond\alpha_1\} \cup \Sigma} \\ \\ (\mathcal{U}) \frac{n : \{\alpha_1 \mathcal{U} \alpha_2\} \uplus \Sigma}{n : \{\alpha_2\} \cup \Sigma \mid n : \{\alpha_1, \Box(\alpha_1 \mathcal{U} \alpha_2)\} \cup \Sigma} \end{array}$$

As discussed in Section 2.1.2, the inductive form of temporal sentences is utilized in the expansion of tableaux. For each $\Box\alpha_1$ at a leaf with the label n at a branch B , α_1 and $\Box\alpha_1$ are true in n . For each sentence $\Diamond\alpha_1$, two possible outcomes are explored: either α_1 is true in n or $\Diamond\alpha_1$ is true in n . Same goes for \mathcal{U} -sentences. These rules are well accepted since their introduction in Wolper’s method [Wol85]. Note that rules over sentences α *consume* (or remove) the initial sentence, and either propagates its subsentences α_i or replaces them with \Box -sentences. In the case where a leaf contains no sentence left (Empty condition), we consider there was no conflict up at the current leaf node and all of the sentences were all consumed by applying the aforementioned rules. As such the branch is ticked with (✓), indicating that it is a successful branch. Note that when a node contains a sentence α and its negation $\neg\alpha$ in the same node x (Contradiction condition), we consider that there is conflict in the current branch. As such, the branch is crossed (✗), indicating that it is an unsuccessful branch.

Following Reynolds’s approach [Rey16a], let an eventuality $\Diamond\alpha$ (\mathcal{U} -sentences are also considered eventualities) be in a node x with a label n . If the sentence α appears in a successor node x' with the label $m \geq n$, we say that $\Diamond\alpha$ at the position n is *fulfilled* in m . Next, we define *state-labelled nodes*.

Definition 2.20 (State-labelled node). *Let $n : \Gamma$ be a labelled node. We say that it is state-labelled node if Γ contains only sentences of the form p , $\neg p$ or $\bigcirc\alpha$.*

In a branch B of a tree \mathcal{T} with a leaf node x_i , after applying every static rule aforementioned (the order of application of these rules is non-deterministic) that can be applied, all leaf nodes of the generated branches contain only sentences of the form p , $\neg p$ or $\bigcirc\alpha$ in their Γ . State-labelled nodes mark the full expansion of all sentences that hold in a state n . In a state-labelled node, the *transition* rule is applied.

$$\text{(Transition)} \quad \frac{n : \{\bigcirc\alpha_1, \bigcirc\alpha_2, \bigcirc\alpha_3, \dots, \bigcirc\alpha_k\} \uplus \Sigma \ (\Sigma \text{ is the set of literals})}{n + 1 : \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_k\}}$$

After the transition rule is applied to a state-labelled node $n : \Gamma$, we add a node with the label $n + 1$. It marks the start of a new temporal state $n + 1$. Only sentences in the scope of \bigcirc -sentences are carried to the new node. The particularity of Reynolds's method is that before applying the transition rule, a set of checks are added in order to prevent branches from expanding indefinitely. These checks are called *loop* and *prune* rules. These rules, together with the transition rule, are called *dynamic rules*. Let $B := \langle x_0, x_1, x_2, \dots, v \rangle$ be a branch where v is a state-labelled node $n : \Gamma_v$. These rules are applied following the order on this list:

1. **[Loop]:** Suppose that there is a state ancestor node u that has the labelled node $m : \Gamma_u$ such that $m < n$, $\Gamma_v \subseteq \Gamma_u$, and for each sentence $\bigcirc(\alpha\mathcal{U}\beta)$ or $\bigcirc\Diamond\beta$ in Γ_u , we have a node x with the label $l : \Gamma_x$ with $m < l \leq n$ such that $\beta \in \Gamma_x$. The branch B in this case is ticked with (\checkmark).
2. **[Prune]:** Suppose that there are two state-labelled ancestors u_1 with the label $m_1 : \Gamma_{u_1}$ and u_2 with the label $m_2 : \Gamma_{u_2}$ such that $m_1 < m_2 < n$ and the same set of sentences $\Gamma_{u_1} = \Gamma_{u_2} = \Gamma_v = \Gamma$. Moreover, for any sentence $\bigcirc(\alpha\mathcal{U}\beta)$ or $\bigcirc\Diamond\beta$ in Γ , if there is x with the label $l_1 : \Gamma_x$ such that $m_2 < l_1 \leq n$ and $\beta \in \Gamma_x$ then there is a node y with $l_2 : \Gamma_y$ with $m_1 < l_2 \leq m_2$ and $\beta \in \Gamma_y$. The branch B in this case is crossed with (\times).
3. **[Transition]:** If none of the above rules apply, then apply transition rule.

Figure 2.5 is a graphical representation of the prune and loop rules. The loop rule states that if a subset of sentences is repeated along two nodes of a branch B such that all of the eventualities within said sentences are satisfied in between, then we can build a model by looping the states (more on that in the upcoming). That is why when this case occurs, the branch B is considered a successful branch.

The prune rule states that if a lead state node at the end of a branch B has a label which already appeared twice before, and between the second and third appearance there were no new eventualities fulfilled that were not already fulfilled between the first and the second occurrence. The branch B is then crossed in order to prioritize other branches that satisfy additional eventualities. Here lies the novelty of Reynolds's method [Rey16a] instead of linking nodes with an ancestor node and pruning it later on the second phase of the tableaux (such is the case in Wolper's method [Wol85]). Branches that do not prioritize the fulfilment of eventualities are not considered to be candidate branches for the satisfiability of the initial sentence.

It can be shown using these set of rules, the termination of the tableau is guaranteed because there can be no branch that expands indefinitely (the sketch proof is provided in the long report of Reynolds's work [Rey16b]). Therefore any tableau \mathcal{T} for an \mathcal{L} sentence is a saturated tableau.

Given a fully expanded tableaux \mathcal{T} for a sentence $\alpha \in \mathcal{L}$ with a successful branch B , Reynolds [Rey16b] shows that using only B , an LTL interpretation I_B can be induced such that $I_B, 0 \models \alpha$. Depending on which condition the branch B is on (either empty or loop), Reynolds [Rey16b] provides the

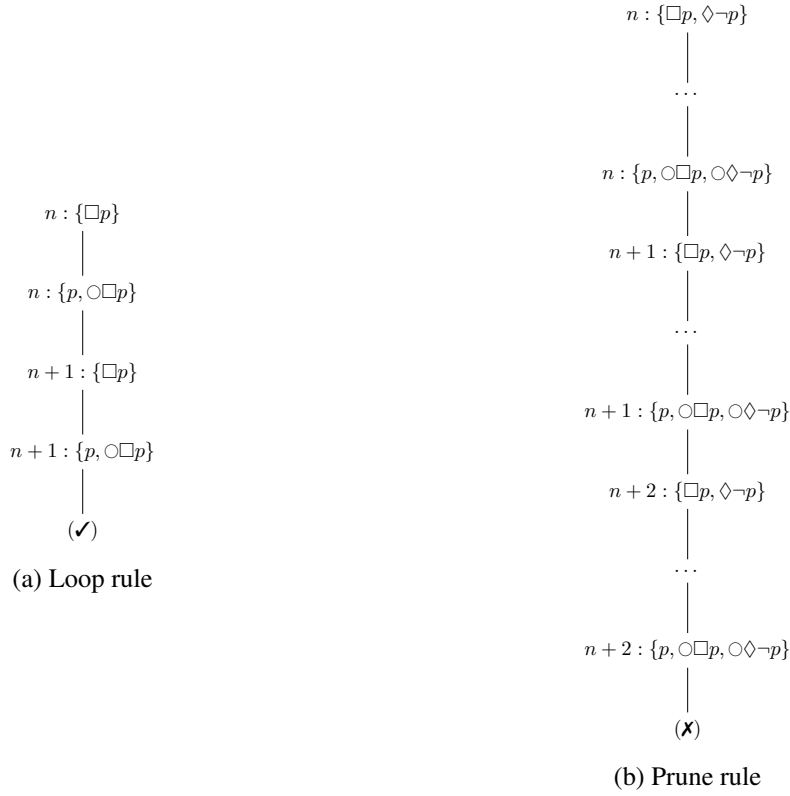


Figure 2.5: Dynamic rules

adequate interpretation I_B that satisfies the sentence α . Reynolds [Rey16b] shows also the completeness of the method by proving that any tableau (Since the order of static rules application does not matter, a sentence can multiple tableaux) for a satisfiable sentence α has a successful branch B .

2.4 Summary

This chapter highlights the notions and notations of *LTL* that we shall use in the upcoming chapters. Furthermore, we show two seminal works in the area of satisfiability in *LTL*. First of which is the work of Sistla and Clarke [SC85] that provides a roadmap for determining the computational properties of this problem. The second one is the work of Reynolds [Rey16a], which is an algorithmic procedure for answering the problem. This concludes the preliminaries portion of the memoir. In the contribution part of the memoir, we shall report the findings of our study, which features a definition of a defeasible extension of *LTL* called defeasible *LTL* and the results obtained around this new formalism.

Part II

Defeasible linear temporal logic

Chapter 3

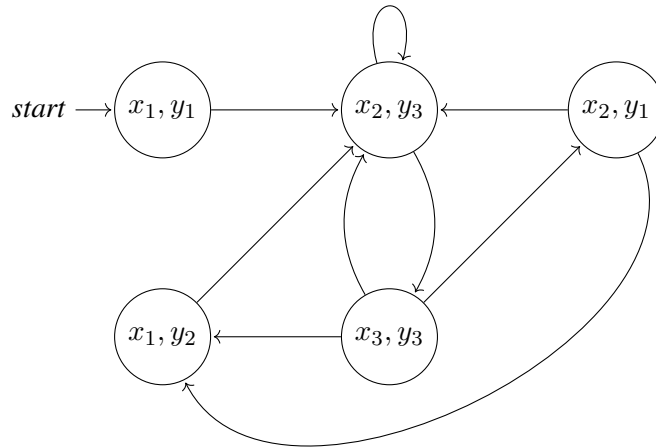
Defeasible linear temporal logic

Contents

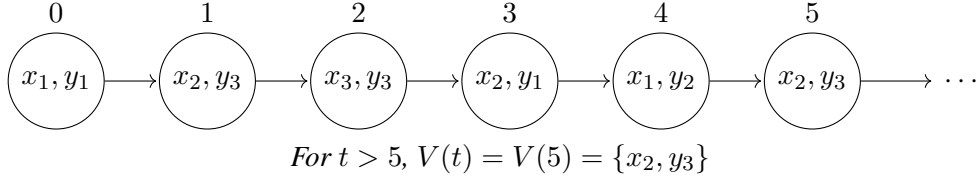
3.1	Introducing defeasible temporal operators	50
3.2	Preferential interpretations	51
3.3	Properties of defeasible temporal modalities	53
3.4	LTL^\sim sub-languages	55
3.4.1	The fragment \mathcal{L}_1	55
3.4.2	The fragment \mathcal{L}^\star	55
3.5	Summary	56

In this chapter, we describe a formalism for reasoning about time that is able to handle exceptional points of time [CCACV20]. We do so by investigating a defeasible extension of LTL with a preferential semantics. The following example introduces a case scenario we shall be using in the remainder of this chapter, with the purpose of giving a motivation for this formalism and better illustrating the definitions in what follows.

Example 3.1. *We shall resume the Example 2.2. Here is the transition system as a reminder:*



Recall that we want to check whether $\Box(x_2 \rightarrow y_3)$ is true. However in this transition system, it is possible to have run where $x_2 \rightarrow y_3$ is false at some time points. Below, is an interpretation that depicts such case:



Under normal circumstances, the program assigns the value 3 to y whenever $x = 2$. We can express this fact using classical *LTL* as follows: $\Box(x_2 \rightarrow y_3)$, with $x_2 \rightarrow y_3$ defined by $\neg x_2 \vee y_3$. Nevertheless, the agent notices that there is one exceptional time point (the time point 3) where the program assigns the value 1 to y when $x = 2$.

Some might consider that the current program is defective at some points of time. In *LTL*, the statement $\Box(x_2 \rightarrow y_3) \wedge \Diamond(x_2 \wedge y_1)$ will always be false, since y cannot have two different values in an iteration where $x = 2$. Nonetheless we want to propose a logical framework that is exception tolerant for reasoning about a system's behaviour. We would like express that $(x_2 \rightarrow y_3)$ is true in all normal time points while taking into account that there might be some exceptional time points where $(x_2 \rightarrow y_3)$ is not necessarily true.

3.1 Introducing defeasible temporal operators

Britz and Varzinczak [BV18] introduced new modal operators called defeasible modalities. Defeasible operators, unlike their classical counterparts, are able to single out normal worlds from those that are less normal or exceptional in the reasoner's mind. Using a similar approach, we extend the vocabulary of classical *LTL* with the *defeasible temporal operators* \Box and \Diamond . Sentences of the resulting logic *LTL*[~] are built up according to the following grammar:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha \mid \bigcirc\alpha \mid \alpha\mathcal{U}\alpha \mid \Box\alpha \mid \Diamond\alpha$$

Other standard Boolean operators are part of the syntax of *LTL*[~]. Let α, β be two sentences of *LTL*[~] language, the symbol \top is an abbreviation of $\alpha \vee \neg\alpha$, \perp is an abbreviation of $\alpha \wedge \neg\alpha$, the implication operator is $\alpha \rightarrow \beta \stackrel{\text{def}}{=} \neg\alpha \vee \beta$ and the equivalence operator is $\alpha \leftrightarrow \beta \stackrel{\text{def}}{=} (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. The intuition behind the defeasible operators in *LTL*[~] is the following: \Box reads as *defeasible always* and \Diamond reads as *defeasible eventuality*. The set of all well-formed *LTL*[~] sentences is denoted by \mathcal{L}^{\sim} . It is worth to mention that any well-formed sentence $\alpha \in \mathcal{L}$ is a sentence of \mathcal{L}^{\sim} . Here are some examples of well-formed sentences in *LTL*[~].

Example 3.2. Let $p, q \in \mathcal{P}$:

$$p, \quad \neg p, \quad \Box(p \wedge q) \rightarrow \Diamond p, \quad \Diamond\Box p, \quad \Box\Diamond p \wedge \Box\neg q$$

Same as the negation and temporal operators, defeasible operators have higher precedence than the other operators. As such, the sentence $\Box\Diamond p \wedge \Box\neg q$ is the same as $(\Box\Diamond p) \wedge (\Box\neg q)$ and not $\Box\Diamond(p \wedge \Box\neg q)$.

Example 3.3. Going back to Example 3.1, we can describe the normal behaviour of the program using the statement $\Box(x_2 \rightarrow y_3) \wedge \Diamond(x_2 \wedge y_1)$. In all normal future time points, the program assigns the value 3 to y whenever $x = 2$. Although unlikely, there are some exceptional time points in the future where $x = 2$ and $y = 1$. But those are “ignored” by the defeasible always operator.

Normality can be expressed using non-monotonic operators. A similar version of the classical properties (see Section 2.1.1) can be expressed over runs that contain exceptional time points. These defeasible properties target future time points that are *normal* on one hand, and ignore states that are *exceptional* on the other. Here are some defeasible properties that can be expressed in LTL^\sim .

- Defeasible safety: $\Box\alpha$ means that the property α holds for all normal future time points of the run.
- Pertinent liveness: $\Diamond\alpha$ means that the property α will hold in a normal future time point of the run.
- Defeasible response: $\Box\Diamond\alpha$ means that for all normal time points of the run, there is a later normal time point where α holds.
- Defeasible persistence: $\Diamond\Box\alpha$ means that there exists a normal time point in the run such that from then onward α holds for all normal time points.

The reasoner can therefore express defeasible properties using these new modalities, and more importantly, use it alongside the rest of other operators of LTL^\sim . Next we shall discuss how to interpret statements of LTL^\sim formalism and how to determine the truth values of each well-formed sentence in \mathcal{L}^\sim .

3.2 Preferential interpretations

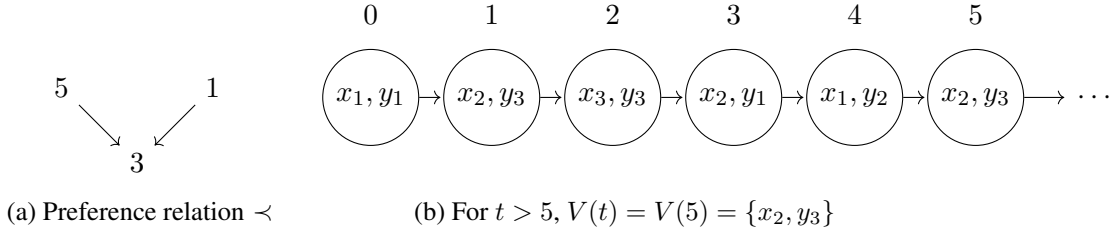
Moving on to the semantics, an LTL^\sim interpretation I is a pair $I \stackrel{\text{def}}{=} (V, \prec)$. Recall that in Section 2.1.1, a temporal structure is represented by the ordering of integers $(\mathbb{N}, <)$. This shall not change for defeasible LTL interpretations. The function V is a valuation function which associates each time point $t \in \mathbb{N}$ with a truth assignment of all propositional atoms $V(t)$. The preferential component \prec of the interpretation of LTL^\sim is directly inspired by the preferential semantics proposed by Shoham [Sho88] and used in the KLM approach [KLM90]. The preference relation \prec is a strict partial order on points of time. Following the KLM approach [KLM90], $t \prec t'$ means that t is more normal or preferred than t' . Time points can be ordered using the relation \prec , the closer they are to being minimal with respect to \prec , the more preferable they are, and vice versa, the farther they are to being minimal with respect to \prec , the more exceptional they become. We also use the notation $(t, t') \in \prec$ to indicate that t is more preferred than t' . The relation \prec is an ordering relation for time points of a temporal structure, which symbolizes the preference over them. Before setting the formal definition for LTL^\sim interpretation, we introduce the notion of minimality and well-foundedness w.r.t. the relation \prec .

Definition 3.4. (Minimality w.r.t. \prec) Let \prec be a strict partial order on \mathbb{N} and $N \subseteq \mathbb{N}$. The set of the minimal elements of N w.r.t. \prec , denoted by $\min_\prec(N)$, is defined by $\min_\prec(N) \stackrel{\text{def}}{=} \{t \in N \mid \text{there is no } t' \in N \text{ such that } t' \prec t\}$.

Definition 3.5 (Well-founded set). Let \prec be a strict partial order on \mathbb{N} . We say \mathbb{N} is well-founded w.r.t. \prec iff $\min_\prec(N) \neq \emptyset$ for every $\emptyset \neq N \subseteq \mathbb{N}$.

Definition 3.6 (Preferential temporal interpretation). An LTL^\sim interpretation on a set of propositional atoms \mathcal{P} , also called preferential temporal interpretation on \mathcal{P} , is a pair $I \stackrel{\text{def}}{=} (V, \prec)$ where V is a mapping function $V : \mathbb{N} \rightarrow 2^{\mathcal{P}}$, and $\prec \subseteq \mathbb{N} \times \mathbb{N}$ is a strict partial order on \mathbb{N} such that \mathbb{N} is well-founded w.r.t. \prec . We denote the set of preferential temporal interpretations by \mathfrak{I} .

Example 3.7. Going back to the run in Example 3.1, time points where $x = 2$ and $y = 3$ are more preferred than time points where $x = 2$ and $y = 1$. For example, the time point 1 and 5 are more preferred than 3. We extend the interpretation I in Example 3.1 by adding the set $\prec := \{(5, 3), (1, 3)\}$. Figure 3.1 represents a preferential temporal interpretation $I = (V, \prec)$ of the second run. Directed edges ($1 \rightarrow 3$ for example) in Sub-figure a represent the pairs in the preference relation \prec .


 Figure 3.1: Preferential temporal interpretation $I = (V, \prec)$

In what follows, given an ordering relation \prec and a time point $t \in \mathbb{N}$, the set of *preferred future time points relative to t* is the set $\min_{\prec}([t, \infty[)$ which is denoted in short by $\min_{\prec}(t)$. It is also worth pointing out that given a preferential interpretation $I = (V, \prec)$ and \mathbb{N} , the set $\min_{\prec}(t)$ is always a non-empty subset of $[t, \infty[$ at any time point $t \in \mathbb{N}$.

Preferential temporal interpretations provide us with an intuitive way of interpreting sentences of \mathcal{L}^{\sim} . Let $\alpha \in \mathcal{L}^{\sim}$, let $I = (V, \prec)$ be a preferential interpretation, and let t be a time point in \mathbb{N} . Satisfaction of α at t in I , denoted $I, t \models \alpha$, is defined as follows:

- $I, t \models \Box\alpha$ if $I, t' \models \alpha$ for all $t' \in \min_{\prec}(t)$;
- $I, t \models \Diamond\alpha$ if $I, t' \models \alpha$ for some $t' \in \min_{\prec}(t)$.

The truth values of Boolean connectives and classical modalities are defined as in *LTL*. The intuition behind a sentence of the form $\Box\alpha$ is that α holds in *all* preferred time points that come after t . $\Diamond\alpha$ intuitively means that α holds on at least one preferred time point relative to the future of t .

We say $\alpha \in \mathcal{L}^{\sim}$ is *preferentially satisfiable* if there is a preferential temporal interpretation I and a time point t in \mathbb{N} such that $I, t \models \alpha$. Without loss of generality, we can say that $\alpha \in \mathcal{L}^{\sim}$ is preferentially satisfiable if there is a preferential temporal interpretation I s.t. $I, 0 \models \alpha$. A sentence $\alpha \in \mathcal{L}^{\sim}$ is *valid* (denoted by $\models \alpha$) if for all preferential temporal interpretations I and time points t in \mathbb{N} , we have $I, t \models \alpha$. We shall highlight the study of the satisfiability of *LTL* $^{\sim}$ sentences in the upcoming chapters of the contribution.

Example 3.8. In the interpretation $I = (V, \prec)$ in Figure 3.1, the set of future preferred time points relative to 0 is $\min_{\prec}(0) = \{0, 1, 2, 4\} \cup [5, \infty[$. We have the following:

- The time point 3 has the valuation $V(3) = \{x_2, y_1\}$. Thus, we have $I, 0 \models \Diamond(x_2 \wedge y_1)$ because $I, 3 \models x_2 \wedge y_1$. Moreover, we have $I, 0 \not\models \Box(x_2 \rightarrow y_3)$ because $I, 3 \not\models x_2 \rightarrow y_3$. Therefore, we conclude that $I, 0 \not\models \Box(x_2 \rightarrow y_3) \wedge \Diamond(x_2 \wedge y_1)$. See that since x and y can have one and only one value, then the $\Box(x_2 \rightarrow y_3) \wedge \Diamond(x_2 \wedge y_1)$ is always false (at most, either $\Box(x_2 \rightarrow y_3)$ or $\Diamond(x_2 \wedge y_1)$ but never both).
- Using defeasible temporal operators, we have $I, t \models x_2 \rightarrow y_3$ for all $t \in \min_{\prec}(0)$. See that the exceptional time point 3, on which the statement $x_2 \rightarrow y_3$ is false, is not in $\min_{\prec}(0)$. Therefore, we can infer that $I, 0 \models \Box(x_2 \rightarrow y_3) \wedge \Diamond(x_2 \wedge y_1)$.

We can see that the addition of \prec relation preserves the truth values of all classical temporal sentences. Moreover, for every $\alpha \in \mathcal{L}$, we have that α is satisfiable in *LTL* if and only if α is preferentially satisfiable in *LTL* $^{\sim}$.

3.3 Properties of defeasible temporal modalities

In this section, we discuss properties of defeasible temporal modalities next and compare them to their classical versions. A detailed version of classical temporal modalities can be found in Section 2.1.2.

Proposition 3.9 (Duality). *Let α be a well-formed sentence in \mathcal{L}^\sim . We have:*

$$\models \Box\alpha \leftrightarrow \neg\Diamond\neg\alpha$$

Proof. We take an arbitrary $I = (V, \prec) \in \mathfrak{I}$, $\alpha \in \mathcal{L}^\sim$ and $t \in \mathbb{N}$. For the only-if part, we assume that $I, t \models \Box\alpha$ and suppose that $I, t \not\models \neg\Diamond\neg\alpha$. Since $I, t \models \Box\alpha$, we have $I, t' \models \alpha$ for all $t' \in \min_{\prec}(t)$. By our assumption, we have $I, t \models \Diamond\neg\alpha$. Thus, there is a time point $t' \in \min_{\prec}(t)$ such that $I, t' \models \neg\alpha$. This contradicts with the above fact that $I, t' \models \alpha$ for all $t' \in \min_{\prec}(t)$. Thus, $I, t \models \neg\Diamond\neg\alpha$ and therefore we conclude that $\models \Box\alpha \rightarrow \neg\Diamond\neg\alpha$.

For the if part, we assume that $I, t \models \neg\Diamond\neg\alpha$. Going back to the semantics of the operator \Diamond , if $I, t \models \Diamond\neg\alpha$, then there is a $t' \in \min_{\prec}(t)$ such that $I, t' \models \neg\alpha$. Therefore, $I, t \models \neg\Diamond\neg\alpha$ means that there is no $t' \in \min_{\prec}(t)$ where $I, t' \models \neg\alpha$. Hence, for all $t' \in \min_{\prec}(t)$, we have $I, t' \models \alpha$, and consequently $I, t \models \Box\alpha$. We conclude that $I, t \models \Box\alpha$ and therefore $\models \neg\Diamond\neg\alpha \rightarrow \Box\alpha$. \square

Analogously as for the classical modalities, we have a duality between the \Box and \Diamond operators. The validity $\models \Diamond\alpha \leftrightarrow \neg\Box\neg\alpha$ is also true.

Proposition 3.10. *Let α be a well-formed sentence in \mathcal{L}^\sim . We have:*

$$\models \Box\alpha \rightarrow \Box\alpha \text{ and } \models \Diamond\alpha \rightarrow \Diamond\alpha$$

Proof. We take an arbitrary $I = (V, \prec) \in \mathfrak{I}$, $\alpha \in \mathcal{L}^\sim$ and $t \in \mathbb{N}$.

- We assume that $I, t \models \Box\alpha$. Then, we have $I, t' \models \alpha$ for all $t' \in [t, \infty[$. Moreover, since $\min_{\prec}(t) \subseteq [t, \infty[$, we have $I, t' \models \alpha$ for all $t' \in \min_{\prec}(t)$. Therefore, we have $I, t \models \Box\alpha$. We conclude that $\models \Box\alpha \rightarrow \Box\alpha$.
- We assume that $I, t \models \Diamond\alpha$. Then, there is a $t' \in \min_{\prec}(t)$ such that $I, t' \models \alpha$. Since $\min_{\prec}(t) \subseteq [t, \infty[$ and $t' \in \min_{\prec}(t)$, there is $t' \in [t, \infty[$ such that $I, t' \models \alpha$. Therefore, we have $I, t \models \Diamond\alpha$. We conclude that $\models \Diamond\alpha \rightarrow \Diamond\alpha$.

\square

Proposition 3.10 states that if a statement holds in all of future time points of any given point of time t , it holds on all *preferred future* time points. As intended, this property establishes the defeasible always as “weaker” than the classical always. It can commonly be accepted since the set of all preferred future states are in the future. This is why we named \Box *defeasible always*. On the other hand, we see that \Diamond is “stronger” than classical eventually, the statement within \Diamond holds at a preferable future.

Next, we discuss the axioms that hold for classical modalities (\Box, \Diamond) and compare them with defeasible modalities (\Box, \Diamond). In the case of classical modalities, the distributivity axiom (K) $\models \Box(\alpha \rightarrow \beta) \rightarrow (\Box\alpha \rightarrow \Box\beta)$, the reflexivity axiom (T) $\models \Box\alpha \rightarrow \alpha$ and the transitivity axiom (4) $\models \Box\alpha \rightarrow \Box\Box\alpha$ are valid [BA12b]. As for defeasible modalities, we have the following:

Proposition 3.11 (Axiom \tilde{K}). *Let $\alpha, \beta \in \mathcal{L}^\sim$. We have:*

$$(\tilde{K}) \quad \models \Box(\alpha \rightarrow \beta) \rightarrow (\Box\alpha \rightarrow \Box\beta)$$

Proof. We take an arbitrary $I = (V, \prec) \in \mathfrak{I}$, $\alpha, \beta \in \mathcal{L}^\sim$ and $t \in \mathbb{N}$. We assume that $I, t \models \Box(\alpha \rightarrow \beta)$ and suppose that $I, t \not\models \Box\alpha \rightarrow \Box\beta$. Since $I, t \models \Box(\alpha \rightarrow \beta)$, we have (1) $I, t' \models \alpha \rightarrow \beta$ for all $t' \in \min_\prec(t)$. Going back to the supposition, if $I, t \not\models \Box\alpha \rightarrow \Box\beta$, then $I, t \models \Box\alpha$ and $I, t \models \neg\Box\beta$. Using duality, we obtain $I, t \models \Box\alpha$ and $I, t \models \Diamond\neg\beta$. Since $I, t \models \Diamond\neg\beta$, there is a $t'' \in \min_\prec(t)$ where $I, t'' \models \neg\beta$. Moreover, since $t'' \in \min_\prec(t)$ and $I, t \models \Box\alpha$, we have $I, t'' \models \alpha$. Therefore, we have $I, t'' \models \alpha \wedge \neg\beta$. Thanks to De Morgan's law, we obtain $I, t'' \models \neg(\alpha \rightarrow \beta)$. The result of the supposition conflicts with the assumption (1), as t'' is also in $\min_\prec(t)$ and thus $\alpha \rightarrow \beta$ is true at t'' . Therefore, we have $I, t \models \Box\alpha \rightarrow \Box\beta$. We conclude that $\models \Box(\alpha \rightarrow \beta) \rightarrow (\Box\alpha \rightarrow \Box\beta)$. \square

The axiom of distributivity (K) can be stated in terms of our defeasible temporal operators. We can also verify the validity of these two statements $\models \Box(\alpha \wedge \beta) \leftrightarrow (\Box\alpha \wedge \Box\beta)$ and $\models (\Box\alpha \vee \Box\beta) \rightarrow \Box(\alpha \vee \beta)$.

Proof. • $\models \Box(\alpha \wedge \beta) \leftrightarrow (\Box\alpha \wedge \Box\beta)$. We take an arbitrary $I = (V, \prec) \in \mathfrak{I}$, $\alpha, \beta \in \mathcal{L}^\sim$ and $t \in \mathbb{N}$. For the if part, we assume that $I, t \models \Box\alpha \wedge \Box\beta$. For all $t' \in \min_\prec(t)$, we have $I, t' \models \alpha$ and $I, t' \models \beta$. Therefore, we have $I, t' \models \alpha \wedge \beta$ and thus $I, t \models \Box(\alpha \wedge \beta)$. For the only if part, we assume that $I, t \models \Box(\alpha \wedge \beta)$. For all $t' \in \min_\prec(t)$, we have $I, t' \models \alpha \wedge \beta$. Then, for all $t' \in \min_\prec(t)$, we have $I, t' \models \alpha$ and also $I, t' \models \beta$. Thus, we have $I, t \models \Box\alpha$ and $I, t \models \Box\beta$ and therefore $I, t \models \Box\alpha \wedge \Box\beta$.

• $\models (\Box\alpha \vee \Box\beta) \rightarrow \Box(\alpha \vee \beta)$. We take an arbitrary $I = (V, \prec) \in \mathfrak{I}$, $\alpha, \beta \in \mathcal{L}^\sim$ and $t \in \mathbb{N}$ such that $I, t \models \Box\alpha \vee \Box\beta$. We have $I, t \models \Box\alpha$ or $I, t \models \Box\beta$. We assume that $I, t \models \Box\alpha$. It follows that $I, t' \models \alpha$ for all $t' \in \min_\prec(t)$. Then, we have $I, t' \models \alpha \vee \beta$ for all $t' \in \min_\prec(t)$. Therefore, we have $\Box(\alpha \vee \beta)$. \square

Similarly to the operator \Box , the validity $\models \Box(\alpha \vee \beta) \rightarrow (\Box\alpha \vee \Box\beta)$ is not true. Assume that a preferential interpretation I satisfies $\Box(\alpha \vee \beta)$ at t . This means that for all $t' \in \min_\prec(t)$, either $I, t' \models \alpha$ or $I, t' \models \beta$. Let say that α is true for all $t' \in \min_\prec(t)$ except for one time point t'' which satisfies β instead. In this case, neither $\Box\alpha$ nor $\Box\beta$ are true in t .

Proposition 3.12 (Reflexivity). *Let $\alpha \in \mathcal{L}^\sim$. We have:*

$$(\tilde{T}) \quad \not\models \Box\alpha \rightarrow \alpha$$

The reflexivity axiom (\tilde{T}) for the classical operators does not hold in the case of defeasible modalities. We can easily find an interpretation $I = (V, \prec)$ where $I, t \not\models \Box\alpha \rightarrow \alpha$. Indeed, since we can have $t \notin \min_\prec(t)$ for a temporal point t , we can have $I, t \models \Box\alpha$ and $I, t \models \neg\alpha$. Case in point on the interpretation in Figure 3.1, the set of preferred future time points relative to 3 is $\min_\prec(3) = [4, \infty[$. We can see that $I, t' \models x_2 \rightarrow y_3$ for all $t' \in \min_\prec(3)$ and therefore $I, 3 \models \Box(x_2 \rightarrow y_3)$. However, we have $I, 3 \not\models x_2 \rightarrow y_3$.

Proposition 3.13 (Transitivity). *Let $\alpha \in \mathcal{L}^\sim$. We have:*

$$(\tilde{4}) \quad \not\models \Box\alpha \rightarrow \Box\Box\alpha$$

It is worth to point out that the set of future preferred time points changes dynamically as we move forward in time. Given three time points $t_1 \leq t_2 \leq t_3$, $t_3 \notin \min_\prec(t_1)$ whilst $t_3 \in \min_\prec(t_2)$ could be true in some cases. Hence, if $I, t \models \Box\alpha$ does not imply that for all $t' \in \min_\prec(t)$, $I, t' \models \Box\alpha$. Therefore, the transitivity axiom $(\tilde{4})$ does not hold in the case of defeasible modalities. On the other hand, given those three time points, $t_3 \notin \min_\prec(t_2)$ implies that $t_3 \notin \min_\prec(t_1)$.

We argue that since defeasible modalities are non-monotonic in nature, the reflexivity and transitivity axioms for these type of modalities do not hold. In the case of classical modalities, by combining both

(T) and (4) axioms, we obtain the validity $\models \Box\Box\alpha \leftrightarrow \Box\alpha$. Using duality, we also obtain $\models \Diamond\Diamond\alpha \leftrightarrow \Diamond\alpha$. And as discussed in Proposition 3.12 and 3.13, the two aforementioned validities are false in the case of defeasible modalities, i.e., $\not\models \Box\Box\alpha \leftrightarrow \Box\alpha$ and $\not\models \Diamond\Diamond\alpha \leftrightarrow \Diamond\alpha$. Therefore, there is no collapsing when it comes to defeasible temporal operators.

3.4 LTL^\sim sub-languages

In this memoir, we will focus on two subsets of the language, namely, \mathcal{L}_1 and \mathcal{L}^* . In the sub-language \mathcal{L}_1 , we omit \mathcal{U} and \Box from the set of modalities. Moreover, only Boolean sentences are allowed within the scope of \Box sentences. In the second subset \mathcal{L}^* , the language contains only Boolean connectives, the two defeasible operators \Box, \Diamond and their classical counterparts.

3.4.1 The fragment \mathcal{L}_1

The set of operators consists of $\wedge, \vee, \Diamond, \Box, \bigcirc, \Diamond$. We shall assume in \mathcal{L}_1 are in negation normal form, which means that negation is only applied to atomic propositions. Furthermore, only Boolean connectors are allowed within the scope of \Box sentences. Temporal operators, classical or non-monotonic, are not permitted in the scope of \Box sentences.

In what follows, we describe well formed sentences of \mathcal{L}_1 . In order to do that, we define first the set of Boolean sentences \mathcal{L}_{bool} . Let $p \in \mathcal{P}$, sentences $\alpha_{bool} \in \mathcal{L}_{bool}$ are defined recursively as such:

$$\alpha_{bool} ::= p \mid \neg p \mid \alpha_{bool} \wedge \alpha_{bool} \mid \alpha_{bool} \vee \alpha_{bool}$$

Next, let $\alpha_{bool} \in \mathcal{L}_{bool}$, sentences in \mathcal{L}_1 are recursively defined as such:

$$\alpha ::= \alpha_{bool} \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Diamond\alpha \mid \Box\alpha_{bool} \mid \bigcirc\alpha \mid \Diamond\alpha$$

While the expressivity of \mathcal{L}_1 is restricted, we can express a variety of properties using this language. For instance, we can check for the pertinent liveness property $\Diamond\alpha$, liveness property $\Diamond\alpha$ and the persistence property $\Diamond\Box\alpha$. We can also express another version of the defeasible persistence property $\Diamond\Box\alpha$ (after a normal time point, α holds in all future time points). Nevertheless, defeasible safety $\Box\alpha$ is not allowed and only safety of Boolean properties is allowed $\Box\alpha_{bool}$.

Example 3.14. Here are some examples of well-formed sentences in \mathcal{L}_1 . Let $p, q \in \mathcal{P}$:

$$p, \quad \neg p, \quad \Box(p \wedge q) \rightarrow \Diamond p, \quad \Diamond\Box p, \quad \bigcirc\Box(p \vee q)$$

The following sentences are not well-formed sentences in \mathcal{L}_1 :

$$\Box p, \quad \Box\Diamond(p \wedge q), \quad \Box\Diamond p$$

3.4.2 The fragment \mathcal{L}^*

The second fragment \mathcal{L}^* is a sub-language of \mathcal{L}^\sim on which only Boolean connectives and the temporal operators $\Box, \Box, \Diamond, \Diamond$ (the operators \bigcirc, \mathcal{U} are omitted) are allowed as connectives. Sentences in \mathcal{L}^* are recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha \mid \Box\alpha \mid \Diamond\alpha$$

The fragment \mathcal{L}^* is more expressive than \mathcal{L}_1 . All classical and defeasible properties that are discussed in Sections 2.1.1 and 3.1 can be expressed using this fragment. With the absence of \bigcirc , the inductive form of both \Box and \Diamond cannot be expressed, i.e., $\models \Box\alpha \leftrightarrow \alpha \wedge \bigcirc\Box\alpha$ and $\models \Diamond\alpha \leftrightarrow \alpha \vee \bigcirc\Diamond\alpha$.

Example 3.15. Here are some examples of well-formed sentences in \mathcal{L}^* . Let $p, q \in \mathcal{P}$:

$$p, \neg p, \Box(p \wedge q) \rightarrow \Diamond p, \Diamond \Box p, \Box \Diamond(p \vee q), \Box p, \Box \Diamond p, \Box \Diamond(p \rightarrow q)$$

The following sentences are not well-formed sentences in \mathcal{L}^* :

$$\bigcirc p, p \mathcal{U} q, \bigcirc \Box(p \wedge q)$$

We based the syntax of these fragments on the fragments discussed in Section 2.2.2. In regards to the fragment \mathcal{L}_1 , sentences in \mathcal{L}_1 follows a similar pattern to the $L_{NNF}(\Diamond, \bigcirc)$ fragments, with the addition of \Diamond and allowing \Box sentences only when they have α_{bool} sentences in their scope. For the fragment \mathcal{L}^* , we add our defeasible temporal operators \Box, \Diamond to the fragment $L(\Diamond)$. In the upcoming chapters, we discuss the the satisfiability problem of sentences in these two fragments.

3.5 Summary

We show a new aspect of defeasibility that can be formulated in temporal logic, which is the *normality* in a run. Normality in *LTL* indicates the importance of any time point during a run compared to others. The addition of the preference relation \prec over time points adds a new dimension to the verification of properties of runs by nuancing the desirability of the outcome in each time point. The well-foundness condition (see Definition 3.5) can be intuitively understood, in the sense that, any subset of time points can be categorized from the most desirable to the least desirable.

In this version of defeasible linear temporal logic, non-monotonic temporal operators express specifications similar to their classical counterparts, all whilst taking into account that exceptions might be tolerated at some time points. This gives the defeasible temporal operators the same elegance that classical modalities express when it comes to describing properties on entire runs. When reasoning about run that might have exceptional time points, the reasoner does not need to explicitly mention which are preferable and which are not. The \Box operator is a “weaker” version of \Box that skips time points that are preferable. On the other hand, \Diamond is a “stronger” version of \Diamond that targets preferable time points in the future. With this framework, other defeasible versions of temporal operator can be implemented, such as *until* \mathcal{U} , *release* \mathcal{R} and also past operators. The well-foundness ensures that any subset (past or future of a time point t) has minimal time points w.r.t. \prec , and thus having preferable elements. This will be a subject for future work.

Chapter 4

The satisfiability problem of LTL^\sim sentences

Contents

4.1 State-dependent preferential interpretations	57
4.2 Useful representations of preferential structures	59
4.3 The bounded-model property of the fragment \mathcal{L}_1	62
4.4 The satisfiability problem in \mathcal{L}_1	65
4.5 The bounded-model property of the fragment \mathcal{L}^*	68
4.6 The satisfiability problem in \mathcal{L}^*	74
4.7 Summary	77

With the \mathcal{L}^\sim language and preferential temporal interpretations defined, we address in this chapter an analysis of the satisfiability of \mathcal{L}^\sim sentences. The algorithmic problem is as follows: Given an input sentence $\alpha \in \mathcal{L}^\sim$, decide whether α is preferentially satisfiable. As highlighted in Section 2.2, Sistla and Clarke [SC85] provide, depending on the fragment of \mathcal{L} language, structures that are useful to prove the bounded model property. Then, they lay out the procedures for checking the satisfiability of the sentence within each of these fragments. In order to establish computational properties about the satisfiability problem in LTL extended with defeasible operators such as those we have considered so far. We introduce structures and LTL^\sim fragments inspired by the approach put forward by Sistla and Clarke [SC85]. A part of this work was published in Chafik et al. [CCACV20]. We shall expand it further in this chapter.

This chapter is divided into four parts: we shall discuss in Section 4.1 an interesting sub-class of LTL^\sim interpretations that is useful for establishing the bounded model property for a part of the language. Next, we investigate in Section 4.2 Sistla and Clarke's notations for preferential temporal structures. We proceed then to establish the bounded model property for two of \mathcal{L}^\sim fragments, namely \mathcal{L}_1 (Section 4.3) and \mathcal{L}^* (Section 4.5). Finally, we provide a procedure for checking the satisfiability of sentences within these fragments (Section 4.4 for \mathcal{L}_1 sentences and Section 4.6 for \mathcal{L}^* sentences). Proofs of propositions and lemmas that are not present in the main text of this chapter can be found in Appendix A.

4.1 State-dependent preferential interpretations

The complexity of the satisfiability problem for LTL has been investigated by Sistla and Clarke [SC85]. Since temporal structures are infinite by nature, finite representations of these structures were put in place

in order to check the computational properties of LTL . In the case of LTL^\sim , the preferential component of \mathfrak{I} interpretations could also be infinite. That is why in the study of the satisfiability problem of LTL^\sim , we define a well-behaved ordering relation \prec . In this section, we introduce a subclass of \mathfrak{I} -interpretations called state-dependent interpretations.

Definition 4.1 (State-dependent preferential interpretations). *Let $I = (V, \prec) \in \mathfrak{I}$. I is a state-dependent preferential interpretation iff for every $i, j, i', j' \in \mathbb{N}$, if $V(i') = V(i)$ and $V(j') = V(j)$, then $(i, j) \in \prec$ iff $(i', j') \in \prec$.*

The notation \mathfrak{I}^{sd} denotes the set of all state-dependent interpretations.

Example 4.2. *Let take the preferential temporal interpretation represented of the second run (see Figure 4.1 for a reminder).*

Recall that, time points where $x = 2$ and $y = 3$ are more preferred than time points where $x = 2$ and $y = 1$. In the previous interpretation $I = (V, \prec)$, we had $\prec := \{(5, 3), (1, 3)\}$. Note that for all $t > 5$, we have $V(t) = \{x_2, y_3\}$. Now that if all time points with valuations 1 and 5 are also more preferred than 3, We can use a state-dependent interpretation $I' \in \mathfrak{I}^{sd}$ to represent this case. The interpretation $I' = (V', \prec')$ has the same valuation function as the valuation function V in I . In addition, for all $(t, t') \in \mathbb{N}^2$ such that $V'(t) = \{x_2, y_3\}$ and $V'(t') = \{x_2, y_1\}$, we have $(t, t') \in \prec'$. In other words, the relation \prec' can be defined as such: $\prec' = \{(1, 3), (5, 3)\} \cup \{(t, t') \in \mathbb{N}^2 \mid V'(t) = \{x_2, y_3\} \text{ and } V'(t') = \{x_2, y_1\}\}$.

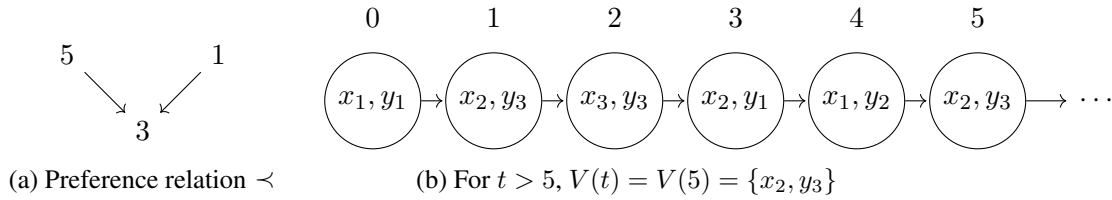


Figure 4.1: Preferential temporal interpretation $I = (V, \prec)$

The intuition behind setting up this restriction is to have a more compact form of expressing the ordering relation over time points. In general, time points with similar valuations are identical with regards to \prec , they express the same normality towards other time points. Moreover, we have an interesting property that does not hold in the general case.

Proposition 4.3. *Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and let $i, i', j, j' \in \mathbb{N}$ s.t. $i \leq i'$, $i' \leq j'$ and $j \in \min_{\prec}(i)$. If $V(j) = V(j')$, then $j' \in \min_{\prec}(i')$.*

Proof. Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and let i, j, i', j' be four time points s.t. $i \leq i'$, $i' \leq j'$ and $j \in \min_{\prec}(i)$. We assume that $V(j) = V(j')$ and we suppose that $j' \notin \min_{\prec}(i')$. Following our supposition, $j' \notin \min_{\prec}(i')$ means that there exists $k \in [i', \infty[$ where $(k, j') \in \prec$. From Definition 4.1, if $(k, j') \in \prec$ and $V(j) = V(j')$, then $(k, j) \in \prec$. Since $(k, j) \in \prec$, we have $j \notin \min_{\prec}(i)$. This conflicts with our assumption of $j \in \min_{\prec}(i)$. We conclude that if $V(j) = V(j')$ then $j' \in \min_{\prec}(i')$. \square

Proposition 4.3 states that whenever $j \in \min_{\prec}(i)$, i.e., j is a preferred future time point of i , then all $j' \geq i$ with the same valuation as j are preferred futures of all time points $i' \geq i$. This property is specific to the class of state-dependent interpretations. We add another property that holds on all interpretations $I \in \mathfrak{I}$.

Proposition 4.4. *Let $I = (V, \prec) \in \mathfrak{I}$ and let $i, j \in \mathbb{N}$ s.t. $j \in \min_{\prec}(i)$. For all $i \leq i' \leq j$, we have $j \in \min_{\prec}(i')$.*

Proof. Let $I = (V, \prec) \in \mathfrak{I}$ and let $i, i', j \in \mathbb{N}$ s.t. $j \in \min_{\prec}(i)$ and $i \leq i' \leq j$. Since $j \in \min_{\prec}(i)$, there is no $j' \in [i, \infty[$ s.t. $(j', j) \in \prec$. Moreover, we have $i \leq i'$, we conclude that there is no $j' \in [i', \infty[$ s.t. $(j', j) \in \prec$. Therefore, we have $j \in \min_{\prec}(i')$. \square

In the case of \mathfrak{I} -interpretations, when a time point j is a preferred time point of i , then the time point j remains a preferred time point of all time points between i and j . State-dependent interpretations are going to be used as the de facto interpretations for the fragment in \mathcal{L}^* . We shall study them in more depth in Sections 4.5 and 4.6.

4.2 Useful representations of preferential structures

Throughout this work, the term temporal sequence, or sequence in short, will denote a sequence of integer numbers in their natural order. A sequence represents a set of time points. They can either be in a continuous interval of integers, a discontinuous sequence or a combination of both. Sequences can also be finite or infinite. In what follows, we define formally the notion of sub-sequences.

Definition 4.5 (Sub-sequence). *Let N, N' be two sequences of natural number. N' is a subsequence of N (written as $N' \subseteq N$) iff for all $i \in N'$, we have $i \in N$.*

We introduce pseudo-interpretations next. A pseudo-interpretation I^N over a sequence N is the restriction of the valuation and the ordering relation of the interpretation $I = (V, \prec)$ to time points of N .

Definition 4.6 (Pseudo-interpretation over N). *Let $I = (V, \prec) \in \mathfrak{I}$ and N be a sequence of natural numbers. The pseudo-interpretation over N is the pair $I^N \stackrel{\text{def}}{=} (V^N, \prec^N)$ where:*

- $V^N : N \longrightarrow 2^{\mathcal{P}}$ is a valuation function over N , where for all $i \in N$, we have $V^N(i) = V(i)$,
- $\prec^N \subseteq N \times N$, where for all $(i, j) \in N^2$, we have $(i, j) \in \prec^N$ iff $(i, j) \in \prec$.

With pseudo-interpretations, we can check the truth values of sentences within sequences of the starting interpretation I . The truth values of \mathcal{L} -sentences in pseudo-interpretations are defined in a similar fashion as for preferential temporal interpretations. With $\models_{\mathcal{P}}$ we denote the truth values of sentences in a pseudo-interpretation.

- $I^N, t \models_{\mathcal{P}} p$ if $p \in V^N(t)$;
- $I^N, t \models_{\mathcal{P}} \neg\alpha$ if $I^N, t \not\models_{\mathcal{P}} \alpha$;
- $I^N, t \models_{\mathcal{P}} \alpha \wedge \beta$ if $I^N, t \models_{\mathcal{P}} \alpha$ and $I^N, t \models_{\mathcal{P}} \beta$;
- $I^N, t \models_{\mathcal{P}} \alpha \vee \beta$ if $I^N, t \models_{\mathcal{P}} \alpha$ or $I^N, t \models_{\mathcal{P}} \beta$;
- $I^N, t \models_{\mathcal{P}} \Box\alpha$ if $I^N, t' \models_{\mathcal{P}} \alpha$ for all $t' \in N$ s.t. $t' \geq t$;
- $I^N, t \models_{\mathcal{P}} \Diamond\alpha$ if $I^N, t' \models_{\mathcal{P}} \alpha$ for some $t' \in N$ s.t. $t' \geq t$;
- $I^N, t \models_{\mathcal{P}} \bigcirc\alpha$ if we have $t + 1 \in N$ and $I^N, t + 1 \models_{\mathcal{P}} \alpha$;
- $I^N, t \models_{\mathcal{P}} \Box\alpha$ if for all $t' \in N$ s.t. $t' \in \min_{\prec^N}(t)$, we have $I^N, t' \models_{\mathcal{P}} \alpha$;
- $I^N, t \models_{\mathcal{P}} \Diamond\alpha$ if $I^N, t' \models_{\mathcal{P}} \alpha$ for some $t' \in N$ s.t. $t' \in \min_{\prec^N}(t)$.

Another observation made by Sistla & Clarke in the case of finite sets of atomic proposition \mathcal{P} is that in every LTL interpretation, there is a time point t after which every t -successor's valuation occurs infinitely many times. This is an obvious consequence of having an infinite set of time points and a finite number of possible valuations. That is the case also for LTL^\sim interpretations.

Lemma 4.7. *Let $I = (V, \prec) \in \mathfrak{I}$. There exists a $t \in \mathbb{N}$ s.t. for all $l \in [t, \infty]$, there is a $k > l$ where $V(l) = V(k)$.*

Definition 4.8. *For an interpretation $I \in \mathfrak{I}$, we denote the first time point where the condition set in Lemma 4.7 is satisfied by t_I .*

With the delimiter t_I defined, we can split each temporal structure into two intervals: an initial and a final part.

Definition 4.9. *Let $I = (V, \prec) \in \mathfrak{I}$. We define:*

- $init(I) \stackrel{\text{def}}{=} [0, t_I]$;
- $final(I) \stackrel{\text{def}}{=} [t_I, \infty]$;
- $range(I) \stackrel{\text{def}}{=} \{V(i) \mid i \in final(I)\}$;
- $val(I) \stackrel{\text{def}}{=} \{V(i) \mid i \in \mathbb{N}\}$;
- $size(I) \stackrel{\text{def}}{=} length(init(I)) + card(range(I))$, where $length(\cdot)$ denotes the length of a sequence and $card(\cdot)$ set cardinality.

In the size of I , we count the number of time points in the initial part and the number of valuations contained in the final part. In the absence of \bigcirc and \mathcal{U} operators (such is the case of the fragment \mathcal{L}^*), the order of time points in final I does not matter [SC85]. In what follows, we show that it is the case if we use \mathcal{L}^* sentences and \mathfrak{I}^{sd} interpretations.

Proposition 4.10. *Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and let $i \leq j \leq i' \leq j'$ be time points in $final(I)$ s.t. $V(j) = V(j')$. Then we have $j \in \min_{\prec}(i)$ iff $j' \in \min_{\prec}(i')$.*

Lemma 4.11. *Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $i \leq i'$ be time points of $final(I)$ where $V(i) = V(i')$. Then for every $\alpha \in \mathcal{L}^*$, we have $I, i \models \alpha$ iff $I, i' \models \alpha$.*

What we have in Lemma 4.11 is that given an interpretation $I \in \mathfrak{I}^{sd}$, points of time in $final(I)$ having the same valuations satisfy exactly the same sentences.

Definition 4.12 (Faithful Interpretations). *Let $I = (V, \prec) \in \mathfrak{I}^{sd}$, $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two interpretations over the same set of atomic propositions \mathcal{P} . We say that I, I' are faithful interpretations if $val(I) = val(I')$ and, for all $i, j, i', j' \in \mathbb{N}$ s.t. $V'(i') = V(i)$ and $V'(j') = V(j)$, we have $(i, j) \in \prec$ iff $(i', j') \in \prec'$.*

Throughout this chapter, we write $init(I) \doteq init(I')$ as shorthand for the condition that states: $length(init(I)) = length(init(I'))$ and for each $i \in init(I)$ we have $V(i) = V'(i)$.

Lemma 4.13. *Let $I = (V, \prec) \in \mathfrak{I}^{sd}$, $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over \mathcal{P} such that $V'(0) = V(0)$ (in case $init(I)$ is empty), $init(I) \doteq init(I')$, and $range(I) = range(I')$. Then for all $\alpha \in \mathcal{L}^*$, we have that $I, 0 \models \alpha$ iff $I', 0 \models \alpha$.*

In the case of an empty initial part, we need to make sure that both of the interpretations start at the same temporal state $V(0)$. Hence, we add the constraint $V'(0) = V(0)$ when $init(I)$ is empty. Lemma 4.13 implies that the ordering of time points in $final(\cdot)$ does not matter, and what matters is the $range(\cdot)$ of valuations contained within it. It is worth to mention that Lemmas 4.11 and 4.13 hold only in \mathfrak{J}^{sd} interpretations and they are not always true in the general case.

Sistla & Clarke [SC85] introduced sequences that display a certain behaviour called *acceptable sequences*. We extend the notion of acceptable sequences for preferential temporal interpretations in \mathfrak{J} as follows:

Definition 4.14 (Acceptable sequence w.r.t. I). *Let $I = (V, \prec) \in \mathfrak{J}$ and N be a sequence of temporal time points. N is an acceptable sequence w.r.t. I iff for all $i \in N \cap final(I)$ and for all $j \in final(I)$ s.t. $V(i) = V(j)$, we have $j \in N$.*

The particularity we are looking for is that any picked time point in $init(\cdot)$ (resp. $final(\cdot)$) will remain in the initial (resp. final) part of the new pseudo-interpretation. It is worth pointing out that an acceptable sequence w.r.t. a preferential temporal interpretation can be either finite or infinite. Moreover, \mathbb{N} is an acceptable sequence w.r.t. any interpretation $I \in \mathfrak{J}$. The purpose behind the notion of acceptable sequence is to build new interpretations starting from an LTL^\sim interpretation.

Given N an acceptable sequence w.r.t. I , if N has a time point t in $final(I)$, then all time points t' that have the same valuation as t must be in N . Thus, we have an infinite sequence of time points in N . As such, we can define an initial part and a final part, in a similar way as LTL^\sim interpretations. We let $init(I, N)$ be the largest subsequence of N that is a subsequence of $init(I)$. Note that if N does not contain any time point of $final(I)$, then N is finite. Also, an empty sequence, by definition, is an acceptable sequence w.r.t. I .

We now define the notions $init(\cdot)$, $final(\cdot)$, $range(\cdot)$, and $size(\cdot)$ for acceptable sequences.

Definition 4.15. *Let $I = (V, \prec) \in \mathfrak{J}$, and let N be an acceptable sequence w.r.t. I . We define the following:*

$$\begin{aligned} init(I, N) &\stackrel{\text{def}}{=} N \cap init(I); \\ final(I, N) &\stackrel{\text{def}}{=} N \setminus init(I, N); \\ range(I, N) &\stackrel{\text{def}}{=} \{V(t) \mid t \in final(I, N)\}; \\ val(I, N) &\stackrel{\text{def}}{=} \{V(t) \mid t \in N\}; \\ size(I, N) &\stackrel{\text{def}}{=} length(init(I, N)) + card(range(I, N)). \end{aligned}$$

It is worth mentioning that the definition of $size(\cdot)$ is different between normal sequences and acceptable sequences. The reason behind it is that sequences do not have the same restriction as acceptable sequences. Thus it is difficult to define an initial part of the sequence that remains included in the initial part of the interpretation, same goes for the final part. That is why the size of pseudo-interpretations is the length of the sequence for normal sequences. In the case of a *finite* normal sequence N , The size of I^N is defined by $size(I, N) \stackrel{\text{def}}{=} length(N)$. Whereas for acceptable sequences, the size of the pseudo-interpretation is the length of the initial part plus the number of distinct valuations in the final part. Thanks to Definition 4.14, given an acceptable sequence w.r.t. I , we have $size(I, N) \leq size(I)$.

Let N_1, N_2 be two sequences of integers. The union of N_1 and N_2 , denoted by $N_1 \cup N_2$, is the sequence containing only and all elements of N_1 and N_2 . If N_1, N_2 are acceptable sequences, we have the following property:

Proposition 4.16. *Let $I = (V, \prec) \in \mathfrak{J}$, N_1, N_2 be two acceptable sequences w.r.t. I . Then $N_1 \cup N_2$ is an acceptable sequence w.r.t. I s.t. $size(I, N_1 \cup N_2) \leq size(I, N_1) + size(I, N_2)$.*

Proposition 4.17. *Let $I = (V, \prec) \in \mathfrak{I}$ and N be an acceptable sequence w.r.t. I . If for all distinct $t, t' \in N$, we have $V(t') = V(t)$ only when both $t, t' \in \text{final}(I, N)$, then $\text{size}(I, N) \leq 2^{|\mathcal{P}|}$.*

In the upcoming sections, we use sequences to establish the bounded model-property of the fragment \mathcal{L}_1 (Section 4.3) and we use acceptable sequences for the bounded-model property of the fragment \mathcal{L}^* (Section 4.5).

4.3 The bounded-model property of the fragment \mathcal{L}_1

The first contribution is to establish certain computational properties regarding the satisfiability problem in \mathcal{L}_1 (see Section 3.4.1). Let \mathcal{P} be a finite set of atomic propositions. Just as a remainder, sentences in \mathcal{L}_1 are recursively defined as follows:

$$\alpha ::= \alpha_{\text{bool}} \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Diamond \alpha \mid \Box \alpha_{\text{bool}} \mid \bigcirc \alpha \mid \Diamond \alpha$$

Where α_{bool} is a sentence that has only Boolean connectives. Next, we discuss the satisfiability of \mathcal{L}_1 sentences. Given an \mathfrak{I} -satisfiable sentence $\alpha \in \mathcal{L}_1$, there exists an interpretation $I \in \mathfrak{I}$ s.t. $I, 0 \models \alpha$. From I , we can find a finite sequence of integers N s.t. the pseudo-interpretation I^N satisfies α , i.e., $I^N, 0 \models_{\mathcal{P}} \alpha$. Then, we can transform the pseudo-interpretation I^N into an interpretation I' which has the same size and satisfies the sentence α . The first observation we make is that if an interpretation I satisfies a Boolean sentence $\alpha_{\text{bool}} \in \mathcal{L}_{\text{bool}}$ at a time point t , then for all pseudo-interpretations I^N over sequences N that contain t , we have $I^N, t \models_{\mathcal{P}} \alpha_{\text{bool}}$. We can extend it further and obtain Proposition 4.18.

Proposition 4.18. *Let $\alpha_{\text{bool}} \in \mathcal{L}_{\text{bool}}$, let $I = (V, \prec) \in \mathfrak{I}$ and N be a sequence containing a time point t s.t. $I^N, t \models_{\mathcal{P}} \alpha_{\text{bool}}$. Then for all $N' \subseteq N$ containing t , we have $I^{N'}, t \models_{\mathcal{P}} \alpha_{\text{bool}}$.*

Proof. Let $\alpha_{\text{bool}} \in \mathcal{L}_{\text{bool}}$, let $I = (V, \prec) \in \mathfrak{I}$ and N be a sequence containing t s.t. $I^N, t \models_{\mathcal{P}} \alpha_{\text{bool}}$. Let N' be a subsequence of N that contains t . We use structural induction based on α_{bool} .

- $\alpha_{\text{bool}} = p$. Since $I^N, t \models_{\mathcal{P}} p$, we know that $p \in V^N(t)$ and therefore $p \in V(t)$. On the other hand, since we have $t \in N'$ and $p \in V(t)$, then we have $p \in V^{N'}(t)$. Therefore, we have $I^{N'}, t \models_{\mathcal{P}} p$.
- $\alpha_{\text{bool}} = \neg p$. Since $I^N, t \models_{\mathcal{P}} \neg p$, we know that $p \notin V^N(t)$ and therefore $p \notin V(t)$. On the other hand, since we have $t \in N'$ and $p \notin V(t)$, then we have $p \notin V^{N'}(t)$. Therefore, we have $I^{N'}, t \models_{\mathcal{P}} \neg p$.
- $\alpha_{\text{bool}} = \alpha_1 \wedge \alpha_2$. We have $I^N, t \models_{\mathcal{P}} \alpha_1 \wedge \alpha_2$, which means $I^N, t \models_{\mathcal{P}} \alpha_1$ and $I^N, t \models_{\mathcal{P}} \alpha_2$. Since N' is a subsequence of N containing t , by the induction hypothesis on α_1 and α_2 , we have $I^{N'}, t \models_{\mathcal{P}} \alpha_1$ and $I^{N'}, t \models_{\mathcal{P}} \alpha_2$. Therefore, we have $I^{N'}, t \models_{\mathcal{P}} \alpha_1 \wedge \alpha_2$.
- $\alpha_{\text{bool}} = \alpha_1 \vee \alpha_2$. We have $I^N, t \models_{\mathcal{P}} \alpha_1 \vee \alpha_2$, which means either $I^N, t \models_{\mathcal{P}} \alpha_1$ or $I^N, t \models_{\mathcal{P}} \alpha_2$. We suppose that $I^N, t \models_{\mathcal{P}} \alpha_1$. Since N' is a subsequence of N containing t , by the induction hypothesis on α_1 , we have $I^{N'}, t \models_{\mathcal{P}} \alpha_1$. Therefore, we have $I^{N'}, t \models_{\mathcal{P}} \alpha_1 \vee \alpha_2$. Same reasoning applies when $I^N, t \models_{\mathcal{P}} \alpha_2$.

□

Next, let $I \in \mathfrak{I}$ be an interpretation, N be a sequence, $\alpha \in \mathcal{L}_1$ and $t \in N$ s.t. $I^N, t \models_{\mathcal{P}} \alpha$. We can show, using structural induction on α , that we can find a finite sequence M that contains t and such that $I^M, t \models_{\mathcal{P}} \alpha$. Moreover, for all sequences $M \subseteq Q \subseteq N$ we have $I^Q, t \models_{\mathcal{P}} \alpha$. We show in the following lemma that $\text{size}(I, M) \leq |\alpha|$ ($|\alpha|$ denotes the number of symbols within α).

Lemma 4.19. *Let $\alpha \in \mathcal{L}_1$, $I = (V, \prec) \in \mathfrak{I}$, $N \subseteq \mathbb{N}$ and $t \in N$ s.t. $I^N, t \models_{\mathcal{P}} \alpha$. Then there exists a finite sequence M containing t such that:*

1. $M \subseteq N$;
2. $\text{size}(I, M) \leq |\alpha|$;
3. for all sequences Q where $M \subseteq Q \subseteq N$, we have $I^Q, t \models_{\mathcal{P}} \alpha$.

Proof. Let $\alpha \in \mathcal{L}_1$, $I = (V, \prec) \in \mathfrak{I}$, $t \in N$ and $N \subseteq \mathbb{N}$ s.t. $I^N, t \models_{\mathcal{P}} \alpha$. We use structural induction on the length of α .

- $\alpha = p$. Let $M = (t)$ be a sequence containing only t . Then M is a finite sequence such that:
 1. since $t \in N$, then $M \subseteq N$; 2. we have $\text{size}(I, M) = 1 \leq |p|$; 3. since $I^N, t \models_{\mathcal{P}} p$. Then we have $p \in V(t)$. Let Q be a sequence s.t. $M \subseteq Q \subseteq N$, we have $t \in Q$. Therefore, we have $p \in V^Q(t)$ and $I^Q, t \models_{\mathcal{P}} p$.
- $\alpha = \neg p$. Let $M = (t)$ be a sequence containing only t . Then M is a finite sequence such that:
 1. since $t \in N$, then $M \subseteq N$; 2. we have $\text{size}(I, M) = 1 \leq |\neg p|$; 3. since $I^N, t \not\models_{\mathcal{P}} \neg p$, then we have $p \notin V(t)$. Let Q be a sequence where $M \subseteq Q \subseteq N$, we have $t \in Q$. Therefore, we have $p \notin V^Q(t)$ and $I^Q, t \models_{\mathcal{P}} \neg p$.
- $\alpha = \alpha_1 \wedge \alpha_2$. Since $I^N, t \models_{\mathcal{P}} \alpha_1 \wedge \alpha_2$, we then have $I^N, t \models_{\mathcal{P}} \alpha_1$ and $I^N, t \models_{\mathcal{P}} \alpha_2$. Using the induction hypothesis on α_1 , there exists a finite sequence M_1 containing t such that:
 1. we have $M_1 \subseteq N$; 2. we have $\text{size}(I, M_1) \leq |\alpha_1|$; 3. for all sequences Q where $M_1 \subseteq Q \subseteq N$, we have $I^Q, t \models_{\mathcal{P}} \alpha_1$.

Similarly, using the induction hypothesis on α_2 , there exists a finite sequence M_2 such that:

1. we have $M_2 \subseteq N$; 2. we have $\text{size}(I, M_2) \leq |\alpha_2|$; 3. for all sequences Q where $M_2 \subseteq Q \subseteq N$, we have $I^Q, t \models_{\mathcal{P}} \alpha_2$.

Let $M = M_1 \cup M_2$. Since M_1 and M_2 contain t , then M is a finite sequence that contains t such that:

1. since $M_1 \subseteq N$ and $M_2 \subseteq N$, then we have $M_1 \cup M_2 \subseteq N$; 2. we have $\text{size}(I, M) = \text{size}(M_1 \cup M_2) \leq \text{size}(I, M_1) + \text{size}(I, M_2) \leq |\alpha_1| + |\alpha_2| \leq |\alpha_1 \wedge \alpha_2|$; 3. let $M \subseteq Q \subseteq N$ be a sequence. Since $M_1 \subseteq Q \subseteq N$, then we have $I^Q, t \models_{\mathcal{P}} \alpha_1$. Similarly, since $M_2 \subseteq Q \subseteq N$, then we have $I^Q, t \models_{\mathcal{P}} \alpha_2$. Therefore, we have $I^Q, t \models_{\mathcal{P}} \alpha_1 \wedge \alpha_2$.

- $\alpha = \alpha_1 \vee \alpha_2$. We have either $I^N, t \models_{\mathcal{P}} \alpha_1$ or $I^N, t \models_{\mathcal{P}} \alpha_2$. We suppose that $I^N, t \models_{\mathcal{P}} \alpha_1$. Using the induction hypothesis on α_1 , there exists a finite sequence M_1 containing t such that:

1. we have $M_1 \subseteq N$; 2. we have $\text{size}(I, M_1) \leq |\alpha_1|$; 3. for all sequences Q where $M_1 \subseteq Q \subseteq N$, we have $I^Q, t \models_{\mathcal{P}} \alpha_1$.

Let $M = M_1$. Since M_1 contains t , then M is a finite sequence that contains t such that:

1. we have $M = M_1 \subseteq N$; 2. we have $\text{size}(I, M) = \text{size}(M_1) \leq |\alpha_1| \leq |\alpha_1 \vee \alpha_2|$; 3. for all sequences Q where $M_1 \subseteq Q \subseteq N$, we have $I^Q, t \models_{\mathcal{P}} \alpha_1$. Therefore, $I, t \models_{\mathcal{P}} \alpha_1 \vee \alpha_2$.

The reasoning is the same when $I^N, t \models_{\mathcal{P}} \alpha_2$.

- $\alpha = \bigcirc \alpha_1$. Since $I^N, t \models_{\mathcal{P}} \bigcirc \alpha_1$, then $t + 1 \in N$ and $I^N, t + 1 \models_{\mathcal{P}} \alpha_1$. Using the induction hypothesis on α_1 , there exists a finite sequence containing $t + 1$ such that:

1. we have $M_1 \subseteq N$; 2. we have $\text{size}(I, M_1) \leq |\alpha_1|$; 3. for all sequences Q where $M_1 \subseteq Q \subseteq N$, we have $I^Q, t + 1 \models_{\mathcal{P}} \alpha_1$.

Let $M = (t) \cup M_1$; then M is a finite sequence containing t such that:

1. since $M_1 \subseteq N$ and $t \in N$, then we have $M \subseteq N$; 2. we have $\text{size}(I, M) = 1 + \text{size}(I, M_1) \leq |\bigcirc \alpha_1|$; 3. let Q be a sequence such that $M \subseteq Q \subseteq N$, we have $t, t + 1 \in M$. Since $M_1 \subseteq Q \subseteq N$, then $I^Q, t + 1 \models_{\mathcal{P}} \alpha_1$. Therefore, we have $I^Q, t \models_{\mathcal{P}} \bigcirc \alpha_1$.

- $\alpha = \Diamond \alpha_1$. Since $I^N, t \models_{\mathcal{P}} \Diamond \alpha_1$, then there exists $t' \in N$ such that $I^N, t' \models_{\mathcal{P}} \alpha_1$. Using the induction hypothesis on α_1 , there exists a finite sequence containing t' such that:

1. we have $M_1 \subseteq N$; 2. we have $\text{size}(I, M_1) \leq |\alpha_1|$; 3. for all sequences Q where $M_1 \subseteq Q \subseteq N$, we have $I^Q, t' \models_{\mathcal{P}} \alpha_1$.

Let $M = (t) \cup M_1$; then M is a finite sequence containing t such that:

1. since $M_1 \subseteq N$ and $t \in N$, then we have $M \subseteq N$; 2. we have $\text{size}(I, M) = 1 + \text{size}(I, M_1) \leq |\Diamond \alpha_1|$; 3. let Q be a sequence such that $M \subseteq Q \subseteq N$. Then we have $t, t' \in M$. Since $M_1 \subseteq Q \subseteq N$ and $t' \in M_1$, then $I^Q, t' \models_{\mathcal{P}} \alpha_1$. Therefore, we have $I^Q, t \models_{\mathcal{P}} \Diamond \alpha_1$.

- $\alpha = \Diamond \alpha_1$. Since $I^N, t \models_{\mathcal{P}} \Diamond \alpha_1$, there exists $t' \in N$ s.t. $t' \in \min_{\prec_N}(t)$. Using the induction hypothesis on α_1 , there exists a finite sequence M_1 containing t' such that:

1. we have $M_1 \subseteq N$; 2. we have $\text{size}(I, M_1) \leq |\alpha_1|$; 3. for all sequences Q where $M_1 \subseteq Q \subseteq N$, we have $I^Q, t' \models_{\mathcal{P}} \alpha_1$.

Let $M = (t) \cup M_1$; then M is a finite sequence containing t such that:

1. since $M_1 \subseteq N$ and $t \in N$, then we have $M \subseteq N$; 2. we have $\text{size}(I, M) = 1 + \text{size}(I, M_1) \leq |\Diamond \alpha_1|$; 3. let Q be a sequence such that $M \subseteq Q \subseteq N$. Since we have $t, t' \in M$, $M_1 \subseteq M \subseteq Q \subseteq N$ and $t' \in M_1$, then (i) $I^Q, t' \models_{\mathcal{P}} \alpha_1$.

We suppose that $t' \notin \min_{\prec_Q}(t)$, there exists $t'' \in Q$ s.t. $(t'', t') \in \prec_Q$. Following this supposition, we have $(t'', t') \in \prec$. Since $t', t'' \in N$, we have $(t'', t') \in \prec^N$, thus $t' \notin \min_{\prec_N}(t)$. This supposition conflicts with our assumption that $t' \in \min_{\prec_N}(t)$. Therefore we have (ii) $t' \in \min_{\prec_Q}(t)$. From (i) and (ii), we conclude that $I^Q, t \models_{\mathcal{P}} \Diamond \alpha_1$.

- $\alpha = \Box \alpha_{bool}$. Since $I^N, t \models_{\mathcal{P}} \Box \alpha_{bool}$, we have $I^N, t \models_{\mathcal{P}} \alpha_{bool}$ for all $t' \in N$ s.t. $t' \geq t$. Let $M = (t)$ be a sequence containing only t . Then we have the following:

1. we have $M \subseteq N$; 2. we have $\text{size}(I, M) = 1 \leq |\Box \alpha_{bool}|$; 3. let $M \subseteq Q \subseteq N$ be a sequence. We need to prove that $I^Q, t \models_{\mathcal{P}} \Box \alpha_{bool}$. Suppose that $I^Q, t \not\models_{\mathcal{P}} \Box \alpha_{bool}$. This means that there exists $t' \in Q$ s.t. $t' \geq t$ and $I^Q, t' \not\models_{\mathcal{P}} \alpha_{bool}$.

On the other hand, since $t' \in Q$, and $Q \subseteq N$, we have $t' \in N$. We know that $I^N, t \models_{\mathcal{P}} \Box \alpha_{bool}$, and $t' \geq t$, therefore $I^N, t' \models_{\mathcal{P}} \alpha_{bool}$. Thanks to Proposition 4.18, since $\alpha_{bool} \in \mathcal{L}_{bool}$, $t' \in Q \subseteq N$ and $I^N, t' \models_{\mathcal{P}} \alpha_{bool}$, we have $I^Q, t' \models_{\mathcal{P}} \alpha_{bool}$, which raises a contradiction with our assumption. Thus, there is no $t' \in Q$ s.t. $t' \geq t$ and $I^Q, t' \not\models_{\mathcal{P}} \alpha_{bool}$. We conclude that $I^Q, t \models_{\mathcal{P}} \Box \alpha_{bool}$.

□

We can generalize the result of Lemma 4.19 in the following corollary.

Corollary 4.20. *Let $\alpha \in \mathcal{L}_1$ and $I = (V, \prec) \in \mathfrak{I}$ s.t. $I, t \models \alpha$. Then there exists a finite sequence M containing t s.t. $I^M, t \models_{\mathcal{P}} \alpha$ and $\text{size}(I, M) \leq |\alpha|$.*

So far, we showed that if we have an interpretation $I \in \mathfrak{I}$ where $I, t \models \alpha$, then we can find a finite sequence M that contains t s.t. $I^M, t \models_{\mathcal{P}} \alpha$. Next, an interpretation $I' \in \mathfrak{I}$ is induced from the pseudo-interpretation I^N which preserves the satisfaction of α . We define formally the construction below.

Definition 4.21 (Induced interpretation). *Let $I = (V, \prec) \in \mathfrak{I}$, let $N = (t_0, t_1, t_2, \dots, t_{n-1})$ where $t_0 < t_1 < t_2 < \dots < t_{n-1}$ be a finite sequence. The interpretation $I' \stackrel{\text{def}}{=} (V', \prec') \in \mathfrak{I}$ is induced from the pseudo-interpretation $I^N = (V^N, \prec^N)$ as follows:*

$$V' : \begin{cases} V'(i) := V^N(t_i), \text{ if } 0 \leq i < n; \\ V'(i) := V^N(n-1), \text{ otherwise.} \end{cases}$$

And for all $0 \leq i, j < n$ s.t. $(t_i, t_j) \in \prec^N$, we have $(i, j) \in \prec'$.

Let $I^N := (V^N, \prec^N)$ be a pseudo-interpretation and let $I = (V', \prec')$ be the I^N -induced interpretation. We can see that $\text{size}(I') \leq \text{size}(I, M)$. The size of the initial part of I' is the sequence N and the final part has one distinct valuation which is the valuation of the last element of the sequence N . We can also see that the truth values of sub-sentences are preserved in the induced interpretation I' . In other words, for every $\alpha \in \mathcal{L}_1$, if $I^N, t_i \models_{\mathcal{P}} \alpha$, then $I', i \models \alpha$.

Theorem 4.22 (Bounded-Model property). *Let $\alpha \in \mathcal{L}_1$ be \mathfrak{I} -satisfiable. Then there exists $I = (V, \prec) \in \mathfrak{I}$ s.t. $\text{size}(I) \leq |\alpha|$ and $I, 0 \models \alpha$.*

Proof. Let $\alpha \in \mathcal{L}_1$ be \mathfrak{I} -satisfiable and let $I = (V, \prec) \in \mathfrak{I}$ where $I, 0 \models \alpha$ be an interpretation that satisfies α . Thanks to Lemma 4.19, since \mathbb{N} is a sequence and $0 \in \mathbb{N}$ s.t. $I, 0 \models \alpha$, then there is a sequence $M \subseteq \mathbb{N}$ containing 0 where $\text{size}(I, M) \leq |\alpha|$ and $I^M, 0 \models \alpha$. We obtain I^N -induced interpretation $I' = (V', \prec')$ by changing the labels of M into a sequence of natural numbers and looping the valuation of the last element of M . We can see that $I', 0 \models \alpha$ and $\text{size}(I') \leq |\alpha|$. □

4.4 The satisfiability problem in \mathcal{L}_1

Thanks to Theorem 4.22, if a sentence $\alpha \in \mathcal{L}_1$ is \mathfrak{I} -satisfiable, then there exists an interpretation I such that $\text{size}(I) \leq |\alpha|$ that satisfies it. Otherwise, if there is no interpretation that satisfies α such that its size is less than the length of α , then the sentence is unsatisfiable. Based on the bounded-model property, we can make a non-deterministic guess for a bounded interpretation and check whether it satisfies the sentence α . Note that the induced \mathfrak{I} -interpretations for sentences in \mathcal{L}_1 sentences have final parts that consist of only one distinct valuation (see Definition 4.21). Not only that, but the preference relation concerns only time points of the initial part. To this purpose, we introduce a compact structure to represent the bounded interpretations obtained on the last section.

Definition 4.23 (Finite preferential structure). *A finite preferential structure is a tuple $S = (n, V_S, \prec_S)$ where: n is an integer such that $n \geq 0$ (where n is intended to be the size of the finite sequence); $V_S : [0, n-1] \rightarrow 2^{\mathcal{P}}$, and $\prec_S \subseteq [0, n-1]^2$ is a strict partial order.*

We define the size of the structure $\text{size}(S) \stackrel{\text{def}}{=} n$. Thanks to these structures, we can build the interpretation $\mathfrak{l}(S)$ in the following way:

Definition 4.24. Given a finite preferential structure $S = (n, V_S, \prec_S)$, let $I(S) \stackrel{\text{def}}{=} (V, \prec)$, $V(t) \stackrel{\text{def}}{=} V_S(t)$, if $t < n$, and $V(t) \stackrel{\text{def}}{=} V_S(n-1)$, otherwise; and $\prec \stackrel{\text{def}}{=} \{(t, t') \mid (t, t') \in \prec_S\}$.

Interpretations of Definition 4.24 are \mathcal{I} -interpretations such that:

- there is a time point n after which all time points $t \geq n$ have the same valuation $V(t) = V(n-1)$;
- the preference relation is only on time points within the initial sequence $[0, n-1]$.

Moreover, we have $\text{size}(I(S)) \leq n$, and thus $\text{size}(I(S)) \leq \text{size}(S)$. The interpretations of Definition 4.21 $I' = (V', \prec')$ can be viewed as an interpretation $I(S)$ issued from a finite preferential structure $S = (n, V_S, \prec_S)$. The structure S can be induced such that $n \stackrel{\text{def}}{=} |N|$ (where N is the finite sequence which I' was induced from, and $|N|$ is its cardinality), $V_S \stackrel{\text{def}}{=} V'(t)$ for all $t < n$ and $\prec_S \stackrel{\text{def}}{=} \{(t, t') \mid t, t' \in [0, n-1] \text{ and } (t, t') \in \prec'\}$. We can go from interpretations of Definition 4.21 to finite preferential structures S thanks to the intermediate interpretation $I(S)$ of Definition 4.24, and go the other way around. We extend also the notion of preferred time points to finite preferential structures S . The formal definition goes as follows: for $t < n$ we have $\min_{\prec_S}(t) \stackrel{\text{def}}{=} \{t' \in [t, n-1] \mid \text{there is no } t'' \in [t, n-1] \text{ with } (t'', t') \in \prec_S\}$. It is easy to show that for every $t, t' \in [0, n-1]$, we have $t' \in \min_{\prec_S}(t)$ iff $t' \in \min_{\prec}(t)$. Finite preferential structures are going to be useful in order to check the satisfiability of the guessed interpretations.

In order to check the satisfiability of \mathcal{L}_1 sentences using a finite preferential structure S , we introduce the notion of labelling sets in order to assign a set of sub-sentences of the original sentence α to each element of the sequence $[0, n-1]$ of S . The set of sub-sentences of α is denoted by $Sf(\alpha)$.

Definition 4.25 (Sub-sentences). Let $\alpha \in \mathcal{L}_1$. The set of all sub-sentences of α , denoted by $Sf(\alpha)$, is recursively defined as follows:

- $Sf(p) \stackrel{\text{def}}{=} \{p\}$;
- $Sf(\neg p) \stackrel{\text{def}}{=} \{\neg p\}$;
- $Sf(\alpha_1 \wedge \alpha_2) \stackrel{\text{def}}{=} Sf(\alpha_1) \cup Sf(\alpha_2) \cup \{\alpha_1 \wedge \alpha_2\}$;
- $Sf(\alpha_1 \vee \alpha_2) \stackrel{\text{def}}{=} Sf(\alpha_1) \cup Sf(\alpha_2) \cup \{\alpha_1 \vee \alpha_2\}$;
- $Sf(\Box \alpha_{bool}) \stackrel{\text{def}}{=} Sf(\alpha_{bool}) \cup \{\Box \alpha_{bool}\}$;
- $Sf(\Diamond \alpha_1) \stackrel{\text{def}}{=} Sf(\alpha_1) \cup \{\Diamond \alpha_1\}$;
- $Sf(\bigcirc \alpha_1) \stackrel{\text{def}}{=} Sf(\alpha_1) \cup \{\bigcirc \alpha_1\}$;
- $Sf(\Diamond \alpha_1) \stackrel{\text{def}}{=} Sf(\alpha_1) \cup \{\Diamond \alpha_1\}$.

With a proof by induction, we can show that the cardinality of the set $Sf(\alpha)$ is $|Sf(\alpha)| \leq |\alpha|$. We define for a structure $S = (n, V_S, \prec_S)$ and a sentence $\alpha \in \mathcal{L}_1$, labelling sets $lab_\alpha^M(\cdot)$ which link a set of sub-sentences of α that hold true in each $t \in [0, n-1]$.

Definition 4.26 (Labelling sets). Let $S = (n, V_S, \prec_S)$ be a structure, $\alpha \in \mathcal{L}_1$. The set of sub-sentences of α in a $t \in [0, n-1]$, denoted by $lab_\alpha^S(t)$, is defined as follows:

- $p \in lab_\alpha^S(t)$ iff $p \in V_S(t)$;
- $\neg p \in lab_\alpha^S(t)$ iff $p \notin V_S(t)$;

- $\alpha_1 \wedge \alpha_2 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1, \alpha_2 \in \text{lab}_\alpha^S(t)$;
- $\alpha_1 \vee \alpha_2 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1 \in \text{lab}_\alpha^S(t)$ or $\alpha_2 \in \text{lab}_\alpha^S(t)$;
- $\Diamond \alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1 \in \text{lab}_\alpha^S(t')$ for some $t' \in [t, n-1]$;
- $\Box \alpha_{bool} \in \text{lab}_\alpha^S(t)$ iff $\alpha_{bool} \in \text{lab}_\alpha^S(t')$ for all $t' \in [t, n-1]$;
- $\bigcirc \alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1 \in \text{lab}_\alpha^S(t+1)$ and $t+1 \leq n-1$;
- $\Diamond \alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1 \in \text{lab}_\alpha^S(t')$ for some $t' \in \min_{\prec_S}(t)$.

The labelling sets $\text{lab}_\alpha^M(\cdot)$ is used to check the satisfiability of the sub-sentences of α in each t in the interval $[0, n-1]$. As mentioned after Definition 4.24, we can represent the bounded interpretations found on the last section. As such, for any given bounded interpretation I' , there is a finite structure $S = (n, V_S, \prec_S)$ such that its $\text{l}(S)$ is the same as I' (same valuation for all time points and same preference relation). Given any induced bounded-interpretation I' , we show that for every $t \in [0, n-1]$ and every $\alpha_1 \in \text{Sf}(\alpha)$ we have $\alpha_1 \in \text{lab}_\alpha^S(t)$ iff $I', t \models \alpha_1$. The proof of Lemma 4.27 can be found on the Appendix A.

Lemma 4.27. *Let $\alpha \in \mathcal{L}_1$ be an \mathfrak{I} -satisfiable sentence and $I = (V, \prec) \in \mathfrak{I}$ be an interpretation such that $I, 0 \models \alpha$. Let I^N be the pseudo-interpretation of I over the finite sequence N such that $I^N, 0 \models_{\mathfrak{P}} \alpha$, and $I' = (V', \prec')$ be the induced interpretation from I^N . Let $S = (n, V_S, \prec_S)$ be the finite preferential structure where $n = |N|$, $V_S(t) = V'(t)$ for each $t \in [0, |N|-1]$ and $\prec_S = \prec'$. Let $\text{l}(S) = (V'', \prec'')$ be the induced interpretation from S . We have the following:*

- $\prec'' = \prec'$ and $V''(t) = V'(t)$ for each $t \in \mathbb{N}$;
- for every $\alpha_1 \in \text{Sf}(\alpha)$, we have $\alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\text{l}(S), t \models \alpha_1$.

The Lemma 4.27 has two interesting results. The first result is we can represent bounded interpretations of Section 4.3 as finite preferential structures. The second result is labelling sets can be used to check the satisfiability of sub-sentences of α within the finite part of the interpretations. Furthermore, we can generalize Lemma 4.27 and obtain this proposition. In fact, the following proposition is a special case of Lemma 4.27 when $t = 0$.

Proposition 4.28. *Given a finite preferential structure $S = (n, V_S, \prec_S)$ and $\alpha \in \mathcal{L}_1$, we have $\text{l}(S), 0 \models \alpha$ iff $\alpha \in \text{lab}_\alpha^S(0)$.*

We describe, in what follows, the algorithm that checks the \mathfrak{I} -satisfiability for \mathcal{L}_1 sentences. Let α be a sentence in \mathcal{L}_1 . If α is satisfiable, then there exists an interpretation $I \in \mathfrak{I}$ where $I, 0 \models \alpha$. Thanks to Theorem 4.22, a new interpretation I' can be induced from I where $I', 0 \models \alpha$, $\text{size}(I') \leq |\alpha|$, $\text{final}(I') = 1$ and the preferential relation \prec' is only on time points within the finite sequence. As discussed after Definition 4.24, a finite preferential structure S can be induced from I' . Therefore, we can make a non-deterministic guess for a finite preferential structure $S = (n, V_S, \prec_S)$ s.t. $\text{size}(S) \leq |\alpha|$. Next, for each $\alpha_1 \in \text{Sf}(\alpha)$ in the increasing order of $|\alpha_1|$ and for each $t \in [0, n-1]$, we update $\text{lab}_\alpha^S(t)$. At the end of this procedure, S is accepted as a model for α iff $\alpha \in \text{lab}_\alpha^S(0)$, otherwise, S is rejected (thanks to Proposition 4.28).

The procedure is polynomial-time bounded. Since the set $\text{Sf}(\alpha)$ is ordered by the increasing length of sub-sentences of α , then each time we want to add a sub-sentence α_1 to $\text{lab}_\alpha^S(t)$, the presence of all of the sub-sentences of α_1 in the labelling set $\text{lab}_\alpha^S(t)$ has already been checked for all $t \in [0, n-1]$. Therefore, checking whether said sub-sentences of α_1 are in a point t' is a simple “yes” or “no” question.

Thus, for each sub-sentence α_1 and $t \in [0, n-1]$, we check only once if $\alpha_1 \in lab_\alpha^S(t)$. We can see that the most costly sentence to check time wise is \Diamond sentences. Say that we check for a sentence $\Diamond\alpha_1 \in lab_\alpha^S(t)$ with $t \in [0, n-1]$. In this case, we need to check whether there is a $t' \in [t, n-1]$ s.t. $\alpha_1 \in lab_\alpha^S(t')$ and $t' \in \min_{\prec_S}(t)$. In the worst case scenario, this takes a time of $\mathcal{O}((t-n)^2)$. Since $size(S) = n$, checking whether $\Diamond\alpha_1 \in lab_\alpha^S(0)$ costs at most $\mathcal{O}(n^2)$, checking whether $\Diamond\alpha_1 \in lab_\alpha^S(1)$ costs at most $\mathcal{O}((n-1)^2)$, and so on. If we add them together, then for all $t \in [0, n-1]$, checking whether $\Diamond\alpha_1 \in lab_\alpha^S(t)$ is $\mathcal{O}(n^3)$. Checking whether \Box and \Diamond sentences for all $t \in [0, n-1]$ costs at most $\mathcal{O}(n^2)$. For \bigcirc and Boolean sentences, it costs at most $\mathcal{O}(n)$. Suppose that $|Sf(\alpha)| = k$, checking for all sub-sentences of α for all $t \in [0, n-1]$ costs $\mathcal{O}(k * n^3)$. Without loss of generality, since $k, n \leq |\alpha|$, then the full expansion of the labelling sets costs $\mathcal{O}(|\alpha|^4)$ at most.

Theorem 4.29. \mathcal{I} -satisfiability for \mathcal{L}_1 sentences is NP-complete.

Proof. \mathcal{I} -satisfiability for \mathcal{L}_1 sentences is at least NP-hard because the satisfiability of Boolean sentences is an NP-hard problem, and Boolean sentences are a subset of \mathcal{L}_1 .

Let $\alpha \in \mathcal{L}_1$. If α is \mathcal{I} -satisfiable, then there is an interpretation $I = (V, \prec) \in \mathcal{I}$ s.t. $I, 0 \models \alpha$. Thanks to Theorem 4.22, an interpretation $I' = (V', \prec)$ where $size(I') \leq |\alpha|$ and $I', 0 \models \alpha$ can be induced from I . The interpretation I' can be represented by a finite preferential structure $S = (n, V_S, \prec_S)$ where $l(S)$ is I' . We make a non-deterministic guess of a finite preferential structure $S = (n, V_S, \prec_S)$ where $size(S) \leq |\alpha|$ and use the labelling sets $lab_\alpha^S(t)$ to check for all sub-sentences of α_1 in each $t \in [0, n-1]$. Thanks to Proposition 4.28, if $\alpha \in lab_\alpha^S(0)$, then S is accepted as a model and therefore α is satisfiable. Otherwise, S is rejected. Using the aforementioned procedure, the labelling sets is polynomial-time bounded in $\mathcal{O}(|\alpha|^4)$. \mathcal{I} -satisfiability for \mathcal{L}_1 sentences is an NP problem. Therefore, \mathcal{I} -satisfiability for \mathcal{L}_1 sentences is NP-complete. \square

4.5 The bounded-model property of the fragment \mathcal{L}^*

The second contribution of our work is to show the decidability of the satisfiability problem of another fragment of defeasible LTL , namely \mathcal{L}^* (See Section 3.4.2). Just as a remainder, sentences in \mathcal{L}^* are recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha \mid \Box\alpha \mid \Diamond\alpha$$

Let $\alpha \in \mathcal{L}^*$ be a sentence. With $|\alpha|$ we denote the number of symbols within α . The main result of this section is summarized in the following theorem, of which the proof will be given in the remainder of the section.

Theorem 4.30 (Bounded-model property). *If $\alpha \in \mathcal{L}^*$ is \mathcal{I}^{sd} -satisfiable, then there is an interpretation $I \in \mathcal{I}^{sd}$ such that $I, 0 \models \alpha$ and $size(I) \leq |\alpha| \times 2^{|\mathcal{P}|}$.*

Hence, given a \mathcal{I}^{sd} -satisfiable sentence $\alpha \in \mathcal{L}^*$, there is an \mathcal{I}^{sd} -interpretation satisfying α of which the size is bounded. Since α is \mathcal{I}^{sd} -satisfiable, we know $I, 0 \models \alpha$. From I we can construct an interpretation I' also satisfying α , i.e., $I', 0 \models \alpha$, which is bounded on its size by $|\alpha| \times 2^{|\mathcal{P}|}$.

The goal of this section is to show how to build said bounded interpretation. Let $\alpha \in \mathcal{L}^*$ and let $I \in \mathcal{I}^{sd}$ be s.t. $I, 0 \models \alpha$. The first step is to characterize an acceptable sequence N w.r.t. I such that N is bounded first of all, and “keeps” the satisfiability of the sub-sentences α_1 contained in α , i.e., if $I, t \models \alpha_1$, then $I^N, t \models_{\mathcal{P}} \alpha_1$ (see Definition 4.6). We do so by building inductively a bounded pseudo-interpretation step by step by selecting what to take from the initial interpretation I for each sub-sentence α_1 contained in α to be satisfied. In what follows, we introduce the notion of *Anchors*(\cdot) as a strategy for picking out the desired time points from I . Definitions 4.33–4.37 tell us how to pick said time points.

Definition 4.31 (Induced acceptable sequence). Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and let N be a sequence of time points. Let N' be the sequence of all time points t' in $\text{final}(I)$ for which there is $t \in N \cap \text{final}(I)$ with $V(t') = V(t)$. With $AS(I, N) \stackrel{\text{def}}{=} N \cup N'$ we denote the induced acceptable sequence of N w.r.t. I .

Example 4.32. Let $I = (V, \prec)$ be the interpretation represented in Figure 4.2 and N be a sequence such that $N = (t_0, t_1, t_2)$ (marked with black circles on the figure). In order to obtain $AS(I, N)$, we look for time points of the sequence N that are in $\text{final}(I)$. The only time point in $\text{final}(I)$ is t_2 and has the valuation $V(t_2) = V_1$. In addition of t_0, t_1, t_2 , the induced acceptable sequence of N w.r.t. I , denoted by $AS(I, N)$, contains all time points in $\text{final}(I)$ that have the same valuation as t_1 (marked with green circles on the figure).

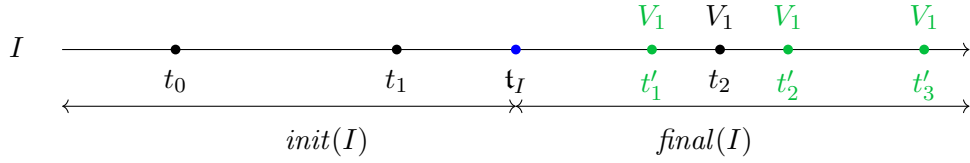


Figure 4.2: Induced acceptable sequence

In the previous definition, N' is the sequence of all time points t' having the same valuation as some time point $t \in N$ that is in $\text{final}(I)$. It is also worth to point out that N' can be empty in the case of there being no time point $t \in N$ that is in $\text{final}(I)$. N is then a finite acceptable sequence w.r.t. I where $AS(I, N) = N$. This notation is mainly used to ensure that we are using the acceptable version of any sequence.

Definition 4.33 (Chosen occurrence w.r.t. α). Let $I = (V, \prec) \in \mathfrak{I}^{sd}$, $\alpha \in \mathcal{L}^*$ and N be an acceptable sequence w.r.t. I s.t. there exists a time point t in N with $I, t \models \alpha$. The chosen occurrence satisfying α in N , denoted by $t_{\alpha}^{I, N}$, is defined as follows:

$$t_{\alpha}^{I, N} \stackrel{\text{def}}{=} \begin{cases} \min_{<} \{t \in \text{final}(I, N) \mid I, t \models \alpha\}, & \text{if } \{t \in \text{final}(I, N) \mid I, t \models \alpha\} \neq \emptyset; \\ \max_{<} \{t \in \text{init}(I, N) \mid I, t \models \alpha\}, & \text{otherwise.} \end{cases}$$

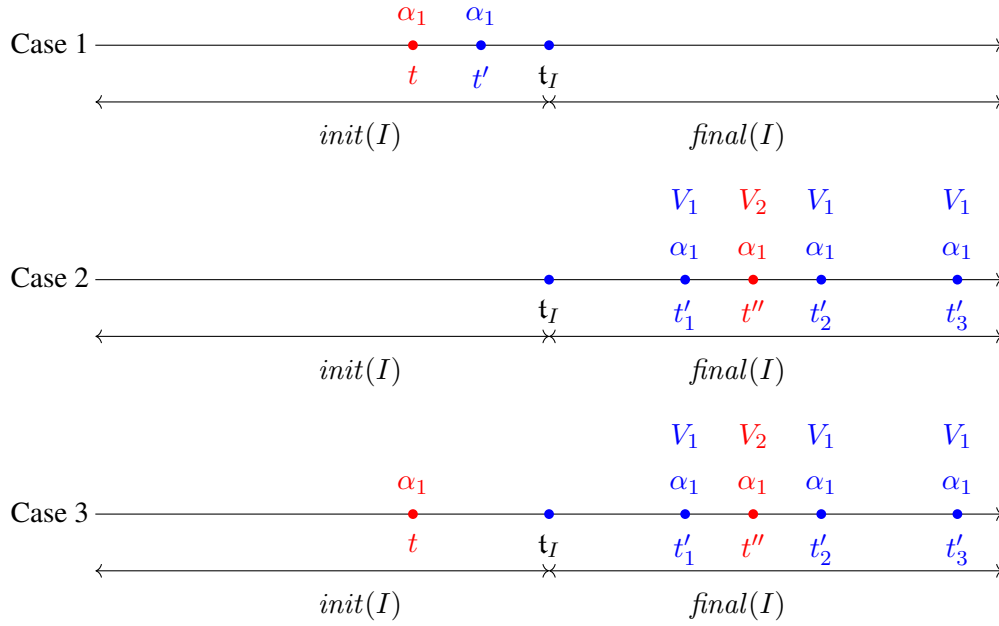
Notice that $<$ above denotes the natural ordering of the underlying temporal structure. The strategy to pick out a time point satisfying a given sentence α in N is as follows. If said sentence is in the final part, we pick the first time point that satisfies it, since we have the guarantee to find infinitely many time points having the same valuation as $t_{\alpha}^{I, N}$ that also satisfy α (see Lemma 4.11). If not, we pick the last occurrence in the initial part that satisfies α . Thanks to Definition 4.33, we can limit the number of time points taken that satisfy the same sentence.

Example 4.34. To highlight the notion of chosen occurrence, we illustrate it in Figure 4.3. On the figure, the coloured circles points are the time points of the sequence N that satisfy α_1 .

In Case 1, both t and t' are in $\text{init}(I)$. We pick the last occurrence which is t' (coloured in blue) as the chosen occurrence $t_{\alpha_1}^{I, N} = t'$.

In Case 2, all of the time points of N that satisfy α are in $\text{final}(I)$. We pick the first occurrence in $N \cap \text{final}(I)$, which is t'_1 as the chosen occurrence $t_{\alpha_1}^{I, N}$.

In Case 3, even when time points of N are both in $\text{init}(I)$ and $\text{final}(I)$, the chosen occurrence $ST(I, N, \alpha_1)$ is the first time point in $N \cap \text{final}(I)$ that satisfies α_1 .


 Figure 4.3: Selected time points of α in $AS(I, N)$

Next, we define the sequence $ST(\cdot)$ as the induced acceptable sequence of the sequence that contains only the chosen occurrence.

Definition 4.35 (Selected time points). Let $I = (V, \prec) \in \mathfrak{I}^{sd}$, N be an acceptable sequence w.r.t. I and $\alpha \in \mathcal{L}^*$ s.t. there is t in N s.t. $I, t \models \alpha$. With $ST(I, N, \alpha) \stackrel{\text{def}}{=} AS(I, (t_{\alpha}^{I, N}))$ we denote the selected time points of N and α w.r.t. I . (Note that $(t_{\alpha}^{I, N})$ is a sequence of only one element.)

Example 4.36. In Example 4.34, we obtained the chosen occurrences for each of the cases represented in Figure 4.3. The next step is to compute $ST(I, N, \alpha_1)$.

In Case 1, since $ST(I, N, \alpha_1)$ is the induced acceptable sequence of $(t_{\alpha_1}^{I, N})$ and $t_{\alpha_1}^{I, N} \in init(I)$, then $ST(I, N, \alpha_1) = (t_{\alpha_1}^{I, N}) = (t')$.

In Case 2, now that $t_{\alpha_1}^{I, N}$ is in $final(I)$, the sequence $ST(I, N, \alpha_1)$ is the acceptable sequence w.r.t. I that contains all time points $final(I)$ with the same valuation as $t_{\alpha_1}^{I, N}$ (coloured in blue in Figure 4.3), i.e., $ST(I, N, \alpha_1) = (t'_1, t'_2, t'_3, \dots)$ with $V(t'_i) = V_1$ for all $i \geq 1$.

In Case 3 and following the same line of reasoning as in Case 2, since we have $t_{\alpha_1}^{I, N} = t'_1$ and $t_{\alpha_1}^{I, N} \in final(I)$, then $ST(I, N, \alpha_1)$ is the sequence $(t'_1, t'_2, t'_3, \dots)$ with $V(t'_i) = V_1$ for all $i \geq 1$ (coloured in blue in Figure 4.3).

Given a sentence $\alpha \in \mathcal{L}^*$ and an acceptable sequence N w.r.t. I s.t. there is at least one time point $t \in N$ where $I, t \models \alpha$, the sequence $ST(I, N, \alpha)$ is the induced acceptable sequence of the sequence $(t_{\alpha}^{I, N})$. If $t_{\alpha}^{I, N} \in init(I)$, the sequence $ST(I, N, \alpha)$ is the sequence $(t_{\alpha}^{I, N})$. Otherwise, the sequence $ST(I, N, \alpha)$ is the sequence of all time points t in $final(I)$ that have the same valuation as $t_{\alpha}^{I, N}$. In both cases, we can see that $size(I, ST(I, N, \alpha)) = 1$.

Given an interpretation $I = (V, \prec)$ and N an acceptable sequence w.r.t. I , the representative sentence of a valuation v is formally defined as $\alpha_v \stackrel{\text{def}}{=} \bigwedge \{p \mid p \in v\} \wedge \bigwedge \{\neg p \mid p \notin v\}$.

Definition 4.37 (Distinctive reduction). Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and let N be an acceptable sequence w.r.t. I . With $DR(I, N) \stackrel{\text{def}}{=} \bigcup_{v \in val(I, N)} ST(I, N, \alpha_v)$ (The definition of $val(I, N)$ can be found in Definition 4.15) we denote the distinctive reduction of N .

Given an acceptable sequence N w.r.t. I , $DR(I, N)$ is the sequence containing the chosen occurrence $t_{\alpha_v}^{I, N}$ that satisfies the representative α_v in N for each $v \in \text{val}(I, N)$. In other words, we pick the selected time points for each possible valuation in $\text{val}(I, N)$. There are two interesting results with regard to $DR(I, N)$. The first one is that $DR(I, N)$ is an acceptable sequence w.r.t. I . This can easily be proven since $ST(I, N, \alpha_v)$ is also an acceptable sequence w.r.t. I , and the union of all $ST(I, N, \alpha_v)$ is an acceptable sequence w.r.t. I (see Proposition 4.16). The second result is that $\text{size}(I, DR(I, N)) \leq 2^{|\mathcal{P}|}$. Indeed, thanks to Proposition 4.16, we can see that $\text{size}(I, DR(I, N)) \leq \sum_{v \in \text{val}(I, N)} \text{size}(I, ST(I, N, \alpha_v))$. Moreover, we have $\text{size}(I, ST(I, N, \alpha_v)) = 1$ for each $v \in \text{val}(I, N)$. On the other hand, there are at most $2^{|\mathcal{P}|}$ possible valuations in $\text{val}(I, N)$. Thus, we can assert that $\sum_{v \in \text{val}(I, N)} \text{size}(I, ST(I, N, \alpha_v)) \leq 2^{|\mathcal{P}|}$, and then we have $\text{size}(I, DR(I, N)) \leq 2^{|\mathcal{P}|}$.

Definition 4.38 (Anchors). *Let $\alpha \in \mathcal{L}^*$ be of the form $\mathcal{O}\alpha_1$ where $\mathcal{O} \in \{\Diamond, \Box, \Diamond, \Box\}$ and $\alpha_1 \in \mathcal{L}^*$. Let $I = (V, \prec) \in \mathcal{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$. The sequence $\text{Anchors}(I, T, \alpha)$ is defined as:*

$$\begin{aligned} \text{Anchors}(I, T, \Diamond\alpha_1) &\stackrel{\text{def}}{=} ST(I, \mathbb{N}, \alpha_1); \\ \text{Anchors}(I, T, \Box\alpha_1) &\stackrel{\text{def}}{=} \emptyset; \\ \text{Anchors}(I, T, \Diamond\alpha_1) &\stackrel{\text{def}}{=} \bigcup_{t \in T} ST(I, AS(I, \min_{\prec}(t)), \alpha_1); \\ \text{Anchors}(I, T, \Box\alpha_1) &\stackrel{\text{def}}{=} DR(I, \bigcup_{t \in T} AS(I, \min_{\prec}(t))). \end{aligned}$$

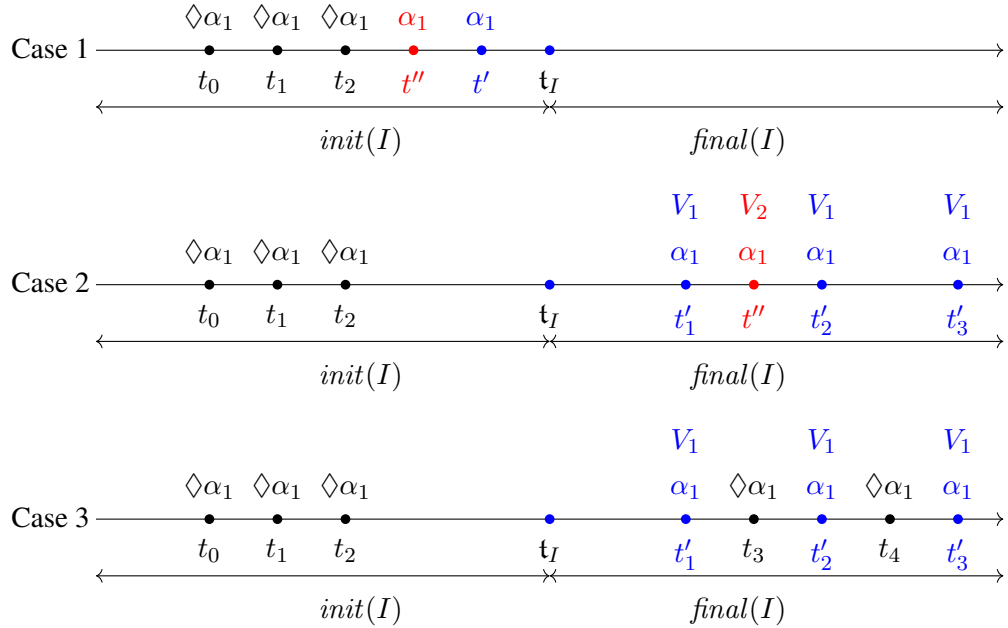
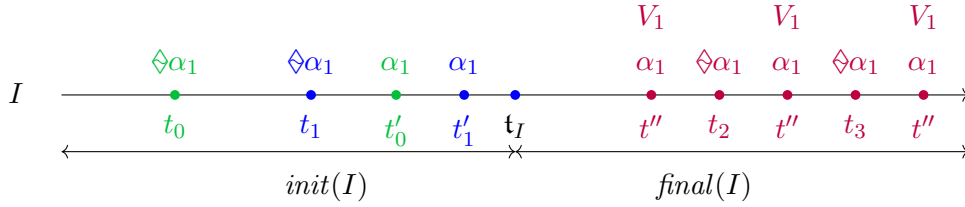


Figure 4.4: Anchors for \Diamond -sentences

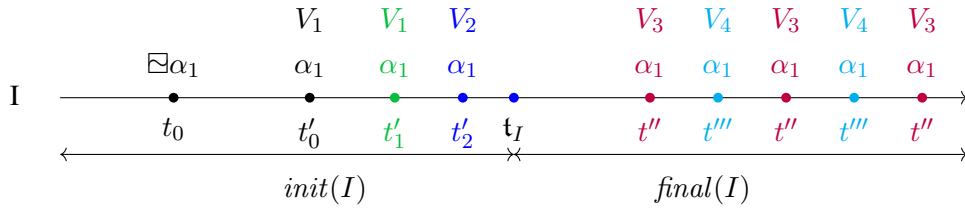
Given an acceptable sequence T w.r.t. $I \in \mathcal{I}^{sd}$ where all of its time points satisfy $\mathcal{O}\alpha_1$ (where $\mathcal{O} \in \{\Diamond, \Box, \Diamond, \Box\}$), $\text{Anchors}(I, T, \mathcal{O}\alpha_1)$ is an acceptable sequence w.r.t. I such that all of its elements have the sub-sentence α_1 . The goal here if we inductively select the time points that satisfy α_1 in $\text{Anchors}(I, T, \mathcal{O}\alpha_1)$, all of the $\mathcal{O}\alpha_1$ sentences in T would then be satisfied. We shall start with $\text{Anchors}(\cdot)$ for $\Diamond\alpha_1$ sentences (see Figure 4.4). Let T be an acceptable sequence w.r.t. I such that all of its elements have the sentence $\Diamond\alpha_1$. In case 1 of Figure 4.4, let N be an acceptable sequence that contains t_0, t_1, t_2 and t' such that $I^N, t' \models_p \alpha_1$. The sentence $\Diamond\alpha_1$ is then satisfied in I^N , i.e., $I^N, t_0 \models_p \Diamond\alpha_1$.

(same goes for t_1, t_2). We can see that t'' is also a candidate that keeps the satisfiability of $\Diamond\alpha_1$ in t_0, t_1 and t_2 . However, in order to have a bounded number of elements, we use the selected time points $ST(\cdot)$ function. If all time points that satisfy α_1 are in the $init(I)$, we pick the last one (t' in case 1). Otherwise, we choose the first candidate that satisfies α_1 in $final(I)$ (t'_1 in case 2) and pick all time points of $final(I)$ that have a similar valuation. Even if said candidate comes before t_i with $\Diamond\alpha_1$ (case 3), there is always a time point with the same valuation that comes after t_i that satisfies α_1 (Thanks to Lemma 4.10, time points with valuation in $final(I)$ satisfy the same sentences). By choosing the first when it comes to $final(I)$ and the last when it comes to $init(I)$, the picked can overlap with each other. This is the essence of $ST(\cdot)$ function and our strategy for picking time points.

It is worth to point out that the choice of $anchors(I, T, \Box\alpha_1) = \emptyset$ is due to the fact α_1 is satisfied starting from the first time point t_0 in T and onwards, i.e., for all $t \geq t_0$, we have $I, t \models \alpha_1$. We need to make sure that the sentence $\Box\alpha_1$ remains satisfied in $t_i \in T$ for all pseudo-interpretations I^N where $T \subseteq N$.


 Figure 4.5: Anchors for \Diamond -sentences

Moving on to $anchors(\cdot)$ for $\Diamond\alpha_1$ sentences (see Figure 4.5), each time point of t_i in T is represented by different color in the figure. For each time point $t_i \in T$, we shall pick the selected time points (using the $ST(\cdot)$ function) in $\min_{\prec}(t_i)$ that satisfy α_1 . In Figure 4.5, each time point t_i and its selected candidate have the same color. Similarly to $\Diamond\alpha_1$, if there is an acceptable sequence N that satisfies α_1 in all of the chosen occurrences. Then, the sentence $\Diamond\alpha_1$ is also satisfied. Later on this section, we will show $anchors(\cdot)$ for $\Diamond\alpha_1$ sentences is an acceptable sequence w.r.t. I that is bounded in its size.


 Figure 4.6: Anchors for \Box -sentences

Finally, $anchors(\cdot)$ for $\Box\alpha_1$ sentences is represented in Figure 4.6. For each time t_i (for simplicity sake, there is only one time point t_0 in Figure 4.6), the selected time points are the chosen occurrence for each distinct valuation in $\min_{\prec}(t_i)$, i.e., V_1, V_2, V_3, V_4 (each distinct valuation is represented by a different color in Figure 4.6). Note that the all the selected time points have the sentence α_1 . These candidates have a particular property that we shall motivate in Proposition 4.41.

The following are some properties of $anchors(\cdot)$ sequence:

Lemma 4.39. *Let $\alpha_1 \in \mathcal{L}^*$ be a sentence, $I = (V, \prec) \in \mathcal{I}^{sd}$ and let T be a non-empty acceptable sequence w.r.t. I where for all $t_i \in T$ we have $I, t_i \models \Diamond\alpha_1$. Then for all $t, t' \in anchors(I, T, \Diamond\alpha_1)$ s.t. $V(t) = V(t')$ and $t \neq t'$, we have $t, t' \in final(I, anchors(I, T, \Diamond\alpha_1))$.*

Proposition 4.40. *Let $\alpha \in \mathcal{L}^*$ be of the form $\mathcal{O}\alpha_1$, where $\mathcal{O} \in \{\Diamond, \Box, \Diamond, \Box\}$ and $\alpha_1 \in \mathcal{L}^*$. Let $I = (V, \prec) \in \mathfrak{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I where for all $t \in T$ we have $I, t \models \alpha$. Then, we have:*

$$\text{size}(I, \text{Anchors}(I, T, \alpha)) \leq 2^{|P|}.$$

Proposition 4.41. *Let $\alpha_1 \in \mathcal{L}^*$, $I = (V, \prec) \in \mathfrak{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \Box\alpha_1$, with $\alpha_1 \in \mathcal{L}^*$. For all acceptable sequences N w.r.t. I s.t. $\text{Anchors}(I, T, \Box\alpha_1) \subseteq N$ and for all $t_i \in N \cap T$, let $I^N = (V^N, \prec^N)$ be the pseudo-interpretation over N and $t' \in N$. We have the following:*

$$\text{If } t' \notin \min_{\prec}(t_i), \text{ then } t' \notin \min_{\prec^N}(t_i).$$

When trying to build the bounded pseudo-interpretation I^N that satisfies α , one problem we encountered is that the set $\min_{\prec^N}(t)$ may include a time point t' that is not in $\min_{\prec}(t)$. This becomes an issue when checking truth values of defeasible sentences in I^N . In order to solve this issue, we defined $\text{Anchors}(\cdot)$ to pick only time points such that we keep truth values of defeasible sentences. In the case of \Diamond -sentences, the sequence $\text{Anchors}(I, T, \Diamond\alpha_1)$ contains the selected time point t'_i that satisfies α_1 and is minimal to t_i w.r.t. \prec for each $t_i \in T$. This is sufficient to preserve the truth $\Diamond\alpha_1$ for each $t_i \in T$. As for \Box -sentences and for each $t_i \in T$, the sequence $\text{Anchors}(I, T, \Box\alpha_1)$ contains selected time points t'_i for each distinct valuation in $\min_{\prec}(t_i)$. As showed in Proposition 4.41, any time point t'_i that is not originally in $\min_{\prec}(t_i)$, is therefore not in $\min_{\prec^N}(t_i)$.

With $\text{Anchors}(\cdot)$ defined, we introduce the notion of $\text{Keep}(\cdot)$. The sequence $\text{Keep}(\cdot)$ will help us to compute recursively, starting from the initial satisfiable sentence α down to its literals, the selected time points to pick in order to induce the pseudo-interpretation I^N that is bounded in size and satisfies α .

Definition 4.42 (Keep). *Let $\alpha \in \mathcal{L}^*$ be in NNF, $I = (V, \prec) \in \mathfrak{I}^{sd}$, and let T be an acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$. The sequence $\text{Keep}(I, T, \alpha)$ is defined as \emptyset , if $T = \emptyset$; otherwise it is recursively defined as follows:*

- $\text{Keep}(I, T, \ell) \stackrel{\text{def}}{=} \emptyset$, where ℓ is a literal;
- $\text{Keep}(I, T, \alpha_1 \wedge \alpha_2) \stackrel{\text{def}}{=} \text{Keep}(I, T, \alpha_1) \cup \text{Keep}(I, T, \alpha_2)$;
- $\text{Keep}(I, T, \alpha_1 \vee \alpha_2) \stackrel{\text{def}}{=} \text{Keep}(I, T_1, \alpha_1) \cup \text{Keep}(I, T_2, \alpha_2)$, where $T_1 \subseteq T$ (resp. $T_2 \subseteq T$) is the sequence of all $t_1 \in T$ (resp. $t_2 \in T$) s.t. $I, t_1 \models \alpha_1$ (resp. $I, t_2 \models \alpha_2$);
- $\text{Keep}(I, T, \Diamond\alpha_1) \stackrel{\text{def}}{=} \text{Anchors}(I, T, \Diamond\alpha_1) \cup \text{Keep}(I, \text{Anchors}(I, T, \Diamond\alpha_1), \alpha_1)$;
- $\text{Keep}(I, T, \Box\alpha_1) \stackrel{\text{def}}{=} \text{Keep}(I, T, \alpha_1)$;
- $\text{Keep}(I, T, \Diamond\alpha_1) \stackrel{\text{def}}{=} \text{Anchors}(I, T, \Diamond\alpha_1) \cup \text{Keep}(I, \text{Anchors}(I, T, \Diamond\alpha_1), \alpha_1)$;
- $\text{Keep}(I, T, \Box\alpha_1) \stackrel{\text{def}}{=} \text{Anchors}(I, T, \Box\alpha_1) \cup \text{Keep}(I, T', \alpha_1)$, where $T' = \bigcup_{t_i \in T} AS(I, \min_{\prec}(t_i))$.

With $\mu(\alpha)$ we denote the number of classical and non-monotonic modalities in α .

Proposition 4.43. *Let $\alpha \in \mathcal{L}^*$ be in NNF, $I = (V, \prec) \in \mathfrak{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$. Then, we have $\text{size}(I, \text{Keep}(I, T, \alpha)) \leq \mu(\alpha) \times 2^{|P|}$.*

Given an acceptable sequence N w.r.t. I , we need to make sure that for each added time point t in the induced pseudo-interpretation I^N , we keep the truth values of the sub-sentences in t , i.e., if $I, t \models \alpha$, then $I^N, t \models_{\mathcal{P}} \alpha$. The sequence $\text{Keep}(I, T, \alpha)$ is the acceptable sequence of time points s.t. if $\text{Keep}(I, T, \alpha) \subseteq N$ and $t \in T$, then said condition is met. We prove this in Lemma 4.44.

Lemma 4.44. *Let $\alpha \in \mathcal{L}^*$ be in NNF, $I = (V, \prec) \in \mathfrak{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$. For all acceptable sequences N w.r.t. I , if $\text{Keep}(I, T, \alpha) \subseteq N$, then for every $t \in N \cap T$, we have $I^N, t \models_{\mathcal{P}} \alpha$.*

Since we build our pseudo-interpretation I^N by adding selected time points for each sub-sentence α_1 of α , we need to make sure that said sub-sentence remains satisfied in I^N . Lemma 4.44 ensures that.

Definition 4.45 (Induced interpretation). *Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and let N be an infinite acceptable sequence w.r.t. I and $t_i, t_j \in N$. The interpretation $I' = (V', \prec') \in \mathfrak{I}^{sd}$ is induced from the pseudo-interpretation $I^N = (V^N, \prec^N)$ as follows:*

- for all $i \geq 0$, we have $V'(i) = V^N(t_i)$;
- for all $i, j \geq 0$, $t_i, t_j \in N$, we have $(t_i, t_j) \in \prec^N$ iff $(i, j) \in \prec'$.

It is worth mentioning that the I^N -induced interpretation I' is a state-dependent interpretation. Moreover, we have $\text{size}(I') = \text{size}(I^N)$. We notice also that if $I^N, t_0 \models \alpha$, then $I', 0 \models \alpha$. We can now prove our bounded-model theorem.

Proof of Theorem 4.30. We assume that $\alpha \in \mathcal{L}^*$ is \mathfrak{I}^{sd} -satisfiable. The first thing we notice is that $|\alpha| \geq \mu(\alpha) + 1$. Let α' be the NNF of the sentence α . As a consequence of the duality rules of \mathcal{L}^* , we can deduce that $\mu(\alpha') = \mu(\alpha)$. Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ s.t. $I, 0 \models \alpha'$. Let $T_0 = AS(I, (0))$ be an acceptable sequence w.r.t. I . We can see that $\text{size}(I, T_0) = 1$. Since for all $t \in T_0$ we have $I, t \models \alpha'$ (see Lemma 4.11), we can compute recursively $U = \text{Keep}(I, T_0, \alpha')$. Thanks to Proposition 4.43, we conclude that U is an acceptable sequence w.r.t. I s.t. $\text{size}(I, U) \leq \mu(\alpha') \times 2^{|\mathcal{P}|}$. Let $N = T_0 \cup U$ be the union of T_0 and U and let $I^N = (V^N, \prec^N)$ be its pseudo-interpretation over N . Thanks to Proposition 4.16, we have $\text{size}(I, N) \leq 1 + \mu(\alpha') \times 2^{|\mathcal{P}|}$. Thanks to Lemma 4.44, since $0 \in N \cap T_0$ and $\text{Keep}(I, T_0, \alpha') \subseteq N$, we have $I^N, 0 \models_{\mathcal{P}} \alpha'$. In case N is finite, we replicate the last time point t_n infinitely many times. Notice that $\text{size}(I, N)$ does not change if we replicate the last element. We obtain the I^N -induced interpretation $I' \in \mathfrak{I}^{sd}$ by changing the labels of N into a sequence of natural numbers minding the order of time points in N (see Definition 4.45). We can see that $\text{size}(I') = \text{size}(I, N)$ and $I', 0 \models \alpha$. Consequently, we have $\text{size}(I') \leq 1 + \mu(\alpha') \times 2^{|\mathcal{P}|}$. Hence, from a given interpretation I s.t. $I, 0 \models \alpha$ we can build an interpretation I' s.t. $I', 0 \models \alpha$ and $\text{size}(I') \leq 1 + \mu(\alpha') \times 2^{|\mathcal{P}|}$. Since $|\alpha| \geq \mu(|\alpha|) + 1$ and $\mu(|\alpha'|) = \mu(|\alpha|)$, we conclude that $\text{size}(I') \leq |\alpha| \times 2^{|\mathcal{P}|}$. \square

4.6 The satisfiability problem in \mathcal{L}^*

We discuss in this section the satisfiability checking problem of the fragment \mathcal{L}^* . We use a similar procedure as described in Section 4.4. Given a sentence α , a bounded structure is non-deterministically guessed, then the labeling sets $\text{lab}_\alpha^S(\cdot)$ are used to update the set of sub-sentences in each element of the sequence and checking whether the sentence α is satisfied. Thanks to Theorem 4.30, if a sentence $\alpha \in \mathcal{L}^*$ is \mathfrak{I}^{sd} -satisfiable, then there exists an interpretation $I \in \mathfrak{I}^{sd}$ s.t. $\text{size}(I) \leq |\alpha| \times 2^{|\mathcal{P}|}$ that satisfies the sentence. We use a compact structure to represent state-dependent interpretations. For this purpose, we focus on particular interpretations of the class \mathfrak{I}^{sd} , namely the ultimately periodic interpretations (UPI in short). We show that any interpretation $I \in \mathfrak{I}^{sd}$ has an equivalent UPI, and the other way round. As we will see in the second part of this section, we define a finite representation of UPIs, called finite preferential structures.

Definition 4.46 (UPI). *Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and let $\pi = \text{card}(\text{range}(I))$. We say I is an ultimately periodic interpretation if:*

- for every $t, t' \in [t_I, t_I + \pi[$ s.t. $t \neq t'$, we have $V(t) \neq V(t')$,
- for every $t \in [t_I, \infty[$, we have $V(t) = V(t_I + (t - t_I) \bmod \pi)$.

A UPI I is a state-dependent interpretation s.t. each time point's valuation in $final(I)$ is replicated periodically. Given a UPI, $\pi = card(range(I))$ denotes the length of the period and the interval $[t_I, t_I + \pi[$ is the first period which is replicated periodically throughout the final part. It is worth pointing out that for every $t \in final(I)$, we have $V(t) \in \{V(t') \mid t' \in [t_I, t_I + \pi[\}$, which is one of the consequences of the definition above. Thanks to Lemma 4.13, we can prove the following proposition.

Proposition 4.47. *Let \mathcal{P} be a set of atomic propositions, $I = (V, \prec) \in \mathfrak{I}^{sd}$, $i = length(init(I))$ and $\pi = card(range(I))$. There exists an ultimately periodic interpretation $I' = (V', \prec') \in \mathfrak{I}^{sd}$ s.t. I, I' are faithful interpretations over \mathcal{P} (see Definition 4.12), $init(I') \doteq init(I)$, $range(I') = range(I)$ (see Definition 4.12 and Lemma 4.13 for reference) and $V'(0) = V(0)$. Moreover, for all $\alpha \in \mathcal{L}^*$, we have $I, 0 \models \alpha$ iff $I', 0 \models \alpha$.*

It is worth to point out that the size of an interpretation and that of its UPI are the same. It can easily be seen that these interpretations have the same initial part and the same range of valuations in the final part. I' from the aforementioned proposition is obtained from I by keeping the same initial part, and placing each distinct valuation of $range(I)$ in the interval $[t_I, t_I + \pi[$ and finally replicating this interval infinitely many times. Moreover, the preference relation \prec' arranges valuations in the same way as \prec . We can see that I and I' are faithful and that $init(I') \doteq init(I)$, $range(I') = range(I)$ and $V'(0) = V(0)$. Therefore, I and its UPI I' satisfy the same sentences.

We showed that, starting from any interpretation $I \in \mathfrak{I}^{sd}$, the equivalent UPI can be induced. Next, we introduce next a compact representation for ultimately periodic structures. The structure used for checking the satisfiability of \mathcal{L}^* sentences is defined in the following way:

Definition 4.48 (Periodical preferential structure). *A periodical preferential structure is a tuple $S = (i, \pi, V_S, \prec_S)$ where: i, π are two integers such that $i \geq 0$ and $\pi > 0$ (where i is intended to be the starting point of the period, π is the length of the period); $V_S : [0, i + \pi[\rightarrow 2^{\mathcal{P}}$, and $\prec_S \subseteq 2^{\mathcal{P}} \times 2^{\mathcal{P}}$ is a strict partial order. Moreover, (I) for all $t \in [i, i + \pi[$, we have $V_S(t) \neq V_S(i - 1)$; and (II) for all distinct $t, t' \in [i, i + \pi[$, we have $V_S(t) \neq V_S(t')$.*

The periodical preferential structure is split into two intervals. The interval $[0, i[$ represents the initial part of a UPI I , and the interval $[i, i + \pi[$ is the first period of the final part of I . Each element in the interval $[i, i + \pi[$ has a unique valuation, they represent all valuations in the range of I . We suppose that the elements of $[i, i + \pi[$ are inter-connected. Since this interval is infinitely replicated in the final part of the interpretation, then every time point with a valuation in $[i, i + \pi[$ is a successor of all time points with valuations in $[i, i + \pi[$. The added properties (I) and (II) make sure that we can build a periodical preferential structure S from a UPI I , and back (the initial part of I coincides with $[0, i[$ and the valuations in $[i, i + \pi[$ are the range of final part of I). Starting from a periodical preferential structure $S = (i, \pi, V_S, \prec_S)$, we can build a UPI $I(S)$ as follows:

Definition 4.49. *Given a periodical preferential structure $S = (i, \pi, V_S, \prec_S)$, let $I(S) \stackrel{\text{def}}{=} (V, \prec)$, where for every $t \geq 0$, $V(t) \stackrel{\text{def}}{=} V_S(t)$, if $t < i$, and $V(t) \stackrel{\text{def}}{=} V_S(i + (t - i) \bmod \pi)$, otherwise. The ordering relation \prec is defined as $\prec \stackrel{\text{def}}{=} \{(t, t') \mid (V(t), V(t')) \in \prec_S\}$.*

Given a periodical preferential structure $S = (i, \pi, V_S, \prec_S)$, the interval $[0, i[$ of the structure corresponds to the initial temporal part of the underlying interpretation $I(S)$ and $[i, i + \pi[$ represents a temporal period that is infinitely replicated in order to determine the final temporal part of the interpretation. The

ordering relation \prec of $l(S)$ is the projection of \prec_S over the time points in the sequence. It follows directly from \prec_S being an ordering relation on valuations that the relation \prec of $l(S)$ satisfies the condition of state-dependent interpretations, i.e., $l(S) \in \mathcal{I}^{sd}$. In addition, we have $size(l(S)) = i + \pi$. We define the size of the periodical preferential structure as $size(S) \stackrel{\text{def}}{=} i + \pi$.

Definition 4.50 (Minimality). *Let $S = (i, \pi, V_S, \prec_S)$ be a periodical preferential structure and t be a time point s.t. $t \in [0, i + \pi[$. The set of preferred time points of t w.r.t. S , denoted by $\min_{\prec_S}(t)$, is defined as follows: $\min_{\prec_S}(t) \stackrel{\text{def}}{=} \{t' \in [\min_{\prec}\{t, i\}, i + \pi[\mid \text{there is no } t'' \in [\min_{\prec}\{t, i\}, i + \pi[\text{ with } (V_S(t''), V_S(t')) \in \prec_S\}$.*

The definition of minimality in periodical preferential structures follows the principle of future preferred time points in the preferential interpretations. Given a $t \in [0, i + \pi[$, the set $\min_{\prec_S}(t)$ contains the minimal elements that come after t . Notice that in the case of $t \in [0, i[$, the minimal set starts with t and finishes with $i + \pi - 1$. Whereas in the case of $t \in [i, i + \pi[$, we recall that in Definition 4.48 the interval $[i, i + \pi[$ is a finite representation of the final part of an interpretation where the elements within this interval are successors of each other, then the set $\min_{\prec_S}(t)$ contains all minimal elements of $[i, i + \pi[$.

Proposition 4.51. *Let $S = (i, \pi, V_S, \prec_S)$ be a periodical preferential structure, $l(S) = (V, \prec)$ be its corresponding interpretation and $t, t', t_S, t'_S \in \mathbb{N}$ s.t.:*

$$t_S = \begin{cases} t, & \text{if } t < i; \\ i + (t - i) \bmod \pi, & \text{otherwise.} \end{cases} \quad t'_S = \begin{cases} t', & \text{if } t' < i; \\ i + (t' - i) \bmod \pi, & \text{otherwise.} \end{cases}$$

We have the following: $t' \in \min_{\prec}(t)$ iff $t'_S \in \min_{\prec_S}(t_S)$.

With the periodical preferential structures S introduced, we move to the procedure for checking the satisfiability of \mathcal{L}^* sentences. We use a similar procedure to the one described in Section 4.4. Let $\alpha \in \mathcal{L}^*$ be a sentence, we define first the ordered set of sub-sentences of α_1 .

Definition 4.52 (Sub-sentences). *Let $\alpha \in \mathcal{L}^*$, the set $Sf(\alpha)$ is recursively defined as follows:*

- $Sf(p) := \{p\}; Sf(\neg p) := \{\neg p\};$
- $Sf(\alpha_1 \wedge \alpha_2) := Sf(\alpha_1) \cup Sf(\alpha_2) \cup \{\alpha_1 \wedge \alpha_2\};$
- $Sf(\alpha_1 \vee \alpha_2) := Sf(\alpha_1) \cup Sf(\alpha_2) \cup \{\alpha_1 \vee \alpha_2\};$
- $Sf(\Box \alpha_1) := Sf(\alpha_1) \cup \{\Box \alpha_1\};$
- $Sf(\Diamond \alpha_1) := Sf(\alpha_1) \cup \{\Diamond \alpha_1\};$
- $Sf(\Box \alpha_1) := Sf(\alpha_1) \cup \{\Box \alpha_1\};$
- $Sf(\Diamond \alpha_1) := Sf(\alpha_1) \cup \{\Diamond \alpha_1\}.$

Next, the labelling set $lab_\alpha^S(\cdot)$ is defined accordingly.

Definition 4.53 (Labelling sets). *Let $S = (i, \pi, V_S, \prec_S)$ be a periodical preferential structure, $\alpha \in \mathcal{L}^*$ and $t \in [0, i + \pi[$. The set of sub-sentences of α that hold in t , denoted by $lab_\alpha^S(t)$, is defined as follows:*

- $p \in lab_\alpha^S(t)$ iff $p \in V_S(t); \neg \alpha_1 \in lab_\alpha^S(t)$ iff $\alpha_1 \notin lab_\alpha^S(t);$
- $\alpha_1 \wedge \alpha_2 \in lab_\alpha^S(t)$ iff $\alpha_1, \alpha_2 \in lab_\alpha^S(t); \quad \alpha_1 \vee \alpha_2 \in lab_\alpha^S(t)$ iff $\alpha_1 \in lab_\alpha^S(t)$ or $\alpha_2 \in lab_\alpha^S(t);$

- $\Diamond\alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1 \in \text{lab}_\alpha^S(t')$ for some $t' \in [\min_{<}\{t, i\}, i + \pi[$;
- $\Box\alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1 \in \text{lab}_\alpha^S(t')$ for all $t' \in [\min_{<}\{t, i\}, i + \pi[$;
- $\Diamond\alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1 \in \text{lab}_\alpha^S(t')$ for some $t' \in \min_{<_S}(t)$;
- $\Box\alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\alpha_1 \in \text{lab}_\alpha^S(t')$ for all $t' \in \min_{<_S}(t)$.

The set $\text{lab}_\alpha^S(t)$ contains the set of all sub-sentences of α that hold in t . For each sub-sentence, we can see that the labelling sets mimic the definition for that sentence's semantics. Moreover, we have the following property.

Proposition 4.54. *Given a periodical preferential structure S and $\alpha \in \mathcal{L}^*$, we have $\text{I}(S), 0 \models \alpha$ iff $\alpha \in \text{lab}_\alpha^S(0)$.*

Checking the \mathcal{I}^{sd} -satisfiability for \mathcal{L}^* sentences uses the same procedure described in Section 4.4. Let α be a sentence in \mathcal{L}^* and thanks to Theorem 4.30, if α is \mathcal{I}^{sd} -satisfiable, then there exists an interpretation $I \in \mathcal{I}^{sd}$ s.t. $\text{size}(I) \leq 2^{|\mathcal{P}|} \times |\alpha|$ that satisfies it. A UPI interpretation I' can be induced from I thanks to Proposition 4.47. Since UPI interpretations can be represented by periodical interpretations, we can make a non-deterministic guess for an $S = (i, \pi, V_S, <_S)$ s.t. $\text{size}(S) \leq 2^{|\mathcal{P}|} \times |\alpha|$. Next, for each $\alpha_1 \in \text{Sf}(\alpha)$ in the increasing order of $|\alpha_1|$ and for each $t \in [0, i + \pi[$, we update $\text{lab}_\alpha^S(t)$ accordingly. At the end of this procedure, S is accepted as a structure for α if $\alpha \in \text{lab}_\alpha^S(0)$, otherwise, S is rejected.

Proposition 4.55. *Let $\alpha \in \mathcal{L}^*$. We have that α is \mathcal{I}^{sd} -satisfiable iff there exists a periodical preferential structure S such that $\text{I}(S), 0 \models \alpha$ and $\text{size}(\text{I}(S)) \leq |\alpha| \times 2^{|\mathcal{P}|}$.*

Hence, to decide the satisfiability of a sentence $\alpha \in \mathcal{L}^*$, we can first guess a structure S bounded by $|\alpha| \times 2^{|\mathcal{P}|}$. Next, using the labelling sets of S , we check the satisfiability of α by the UPI $\text{I}(S)$.

Theorem 4.56. *\mathcal{I}^{sd} -satisfiability problem for \mathcal{L}^* sentences is decidable.*

Proof. Let $\alpha \in \mathcal{L}^*$. Thanks to Theorem 4.30 and 4.47, if α is \mathcal{I}^{sd} -satisfiable, there exists a UPI interpretation I s.t. $\text{I}, 0 \models \alpha$ and $\text{size}(I) \leq |\alpha| \times 2^{|\mathcal{P}|}$. We make a non-deterministic guess of a structure $S = (i, \pi, V_S, <_S)$ where $\text{size}(S) \leq |\alpha| \times 2^{|\mathcal{P}|}$ and use the labelling sets $\text{lab}_\alpha^S(t)$ to check for all sub-sentences of α_1 in each $t \in [0, i]$. If $\alpha \in \text{lab}_\alpha^S(0)$, S is accepted as a structure and therefore α is satisfiable. Otherwise, S is rejected. Therefore, the \mathcal{I}^{sd} -satisfiability for \mathcal{L}^* sentences is a decidable problem. \square

4.7 Summary

By following the line of reasoning of Sistla and Clarke [SC85], we established the bounded model property of the preferential temporal interpretations for some fragments of the LTL^\sim language. The fragment \mathcal{L}_1 is based on the work done on $L_{NNF}(\Diamond, \Box)$ and \mathcal{L}^* is based on the work done on the fragment $L(\Diamond)$ of LTL (a brief presentation of both fragments was presented in Section 2.2). We extended their observation and notations in the case of preferential interpretations. In the investigation of the decidability of the defeasible temporal language, we defined a sub-class of \mathcal{I} interpretations called state-dependent interpretations \mathcal{I}^{sd} . In case of the fragment \mathcal{L}_1 , the upper-bound in the size of \mathcal{I} -interpretations is the same as in the $L_{NNF}(\Diamond, \Box)$. However, we notice an exponential blowup when adding the \Box to the syntax in the case of the fragment \mathcal{L}^* (for a sentence α , the upper-bound is $|\alpha| \times 2^{|\mathcal{P}|}$ compared with $|\alpha|$ in the fragment $L(\Diamond)$ in Sistla and Clarke's work [SC85]).

Unlike the fragment \mathcal{L}^* , Lemmas 4.11 and 4.13 do not hold in the general case of \mathcal{L}^\sim . In the case of temporal interpretations for sentences in \mathcal{L} , Sistla and Clarke [SC85] showed that they can be represented as ultimately periodic structures (see Definition 2.10) that are bounded in size (see Lemma 4.5 and Lemma 4.6 in [SC85]). However, the stakes are to show under what circumstances or restrictions preferential interpretations where a similar version of the aforementioned Lemmas can be obtained.

Chapter 5

A one-pass tableau for LTL^\sim

Contents

5.1	Tableau method for \mathcal{L}_1	80
5.1.1	Static rules	81
5.1.2	Dynamic rules	83
5.2	Soundness and completeness	85
5.2.1	Soundness	85
5.2.2	Completeness	89
5.3	Illustrations of tableaux for \mathcal{L}_1 sentences	94
5.4	Summary	97

With the bounded model property and decidability of the two fragments \mathcal{L}_1 and \mathcal{L}^* shown in Chapter 4, the next task we worked on is to develop tools for checking the satisfiability of sentences within these fragments. In LTL , common techniques include automata-based approaches [BCM⁺92, Var96, RV11] as well as tableaux approaches [Wol85, Rey16b]. On the other hand, Britz and Varzinczak [BV18] proposed a tableau for defeasible modal logic **K**, on which they show that it is possible to build Kripke structures with preferential relation \prec on possible worlds. Rules for defeasible modalities \Diamond, \Box (see Section 1.2.4 for their definition) were also laid in the paper. Moreover, the rules for all modalities (whether classical or defeasible) take into account whether the created accessible nodes are preferred or not during the expansion of the tableau. We shall follow this principle for temporal sentences of LTL^\sim . For more details about the approach for defeasible modal logic, we refer the reader to Britz and Varzinczak’s paper [BV18].

As a first step, we proposed a tableau for sentences in the fragment \mathcal{L}_1 [CCACV21]. We based our method on Reynolds’ one-pass tree-shaped tableau [Rey16a]. A brief description of the approach is provided in Section 2.3. We opted for the one-pass tree-shaped tableau for two reasons. First of which is its tree-like structure (the expansion only goes from a parent node to its children), which resembles traditional tableaux. In the literature, tableaux for formalisms extended with preferential semantics *à-la* KLM [GGOP09, BV18] have a similar structure. As such, Reynolds’ approach is a good starting point for investigating preferential semantics for linear temporal logic. Secondly, each branch of the tree is considered a potential model for the sentence independently of other branches. Therefore, it is easier to integrate a preference relation \prec for each branch without worrying for a potential conflicts with other branches. The tableau for defeasible modal logic **K** is similar in this regard [BV18], i.e., a successful branch by itself is a model.

We highlight, in this chapter, the tableau for \mathcal{L}_1 sentences. We discuss the structure of the tableau which accommodates preferential semantics. We provide a rule for \Diamond operator and how to handle defeasible eventualities. We introduce a version of dynamics rules, i.e., prune and loop. We show the soundness and completeness of the method in Section 5.2 alongside some examples in Section 5.3.

5.1 Tableau method for \mathcal{L}_1

Just as a remainder, the fragment \mathcal{L}_1 considers that sentences are in NNF (negation is only allowed on the level of atomic propositions). On the other hand, the non-monotonic operator \Box is omitted from \mathcal{L}_1 . Furthermore, only Boolean sentences are permitted within the scope of \Box . Sentences in \mathcal{L}_1 are recursively defined as such:

$$\alpha ::= \alpha_{bool} \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Diamond \alpha \mid \Box \alpha_{bool} \mid \bigcirc \alpha \mid \Diamond \alpha$$

Sentences of the form $\Diamond \alpha$ are called *eventualities*, because its truth depends on α being true in the future. Similarly, sentences of the form $\Diamond \alpha$ are called *non-monotonic eventualities*. Their truth depends not only on α being true in some future, but it depends also on this future being preferred to the other future time points.

A tableau for $\alpha \in \mathcal{L}_1$ is a tree of nodes. Each node has a positive integer n as a *label*. It has also two sets: one we denote as Γ and the other as *une* (which stands for unfulfilled non-monotonic eventualities, a notion to be detailed below). The set Γ is a subset of \mathcal{L}_1 which contains the sentences in the node. The set *une* is a set of pairs $(n_k, \Diamond \alpha_k)$, where n_k is a label and $\Diamond \alpha_k$ is a non-monotonic eventuality.

Definition 5.1 (Labelled node). *A labelled node is a triple of the form $n : (\Gamma, \text{une})$ where $n \in \mathbb{N}$, $\Gamma \subseteq \mathcal{L}_1$ and $\text{une} \subseteq [0, n] \times \mathcal{L}_1$.*

It is worth to mention that different nodes can have the same label. Intuitively, the nodes labelled by a same integer n represent the set of sentences that are satisfied at the time point associated with n . Hence, these nodes correspond with a given temporal state.

A branch B is a sequence of nodes, we introduce also a strict partial ordering relation \prec_B on the labels of the nodes within the branch. The relation \prec_B is used to infer a preference relation on the temporal states of the branch B . In addition to \prec_B , each branch B has also has a set min_B . It represents the set constraints that the final preference relation issued from B must satisfy. More precisely, each pair (n, n') in min_B indicates that n' represents a preferred temporal state compared to all $n'' \geq n$. We shall see their role on the rule application section.

Definition 5.2 (Branch). *A branch is a tuple $B \stackrel{\text{def}}{=} (\langle x_0, x_1, x_2, \dots \rangle, \prec_B, \text{min}_B)$ where the first element is a sequence of labelled nodes $x_i := n_i : (\Gamma_i, \text{une}_i)$, \prec_B is a strict partial order ($\prec_B \subseteq \mathbb{N} \times \mathbb{N}$) on labels within the branch, and min_B is a set of pairs of labels ($\text{min}_B \subseteq \mathbb{N} \times \mathbb{N}$).*

Let $B := (\langle x_0, x_1, x_2, \dots \rangle, \prec_B, \text{min}_B)$ be a branch, x_n, x_m be two labelled nodes in B . If x_m comes after x_n in the sequence, then x_m is a *successor* of x_n , and x_n is a *predecessor* of x_m . We denote it by $x_n \leq x_m$. Moreover, if x_m is not the same labelled node as x_n , we say that x_m is a *proper successor* of x_n (same goes for a *proper predecessor*). We denote it by $x_n < x_m$. The last node of a branch is called a *leaf node*. When a leaf node is ticked with \checkmark , we say that the branch is a *successful branch*. On the other hand, when a leaf node is crossed with \times , we say that the branch is a *failed branch*.

A tree is a set of branches $\mathcal{T} \stackrel{\text{def}}{=} \{B_0, B_1, B_2, B_3, \dots, B_k\}$ where $k \geq 0$. A tableau \mathcal{T} for α is the limit of a sequence of trees $\langle \mathcal{T}^0, \mathcal{T}^1, \mathcal{T}^2, \dots \rangle$ where the initial tree is $\mathcal{T}^0 := \{(\langle 0 : (\alpha, \emptyset) \rangle, \emptyset, \emptyset)\}$ and every \mathcal{T}^{i+1} is obtained from \mathcal{T}^i by applying a rule on one of its branches. We say that a tableau \mathcal{T} for α is *saturated* if no more rules can be applied after a certain tree \mathcal{T}^j .

5.1.1 Static rules

We have two types of rules, static and dynamic rules. We introduce static rules first. Let \mathcal{T} be a tree, and let B be a branch of \mathcal{T} that has a leaf $n : (\Gamma, une)$. We say that a static rule (ρ) is applicable at the leaf $n : (\Gamma, une)$ if a sentence in Γ or a pair in une instantiates the pattern ρ . A static rule is a rule of the form:

$$(\rho) \frac{n : (\Gamma, une), \prec_B, min_B}{n : (\Gamma_1, une_1), \prec_{B_1}, min_{B_1} \mid \dots \mid n : (\Gamma_k, une_k), \prec_{B_k}, min_{B_k}}$$

In a tree \mathcal{T}^i , after applying the static rule (ρ) , we obtain the tree \mathcal{T}^{i+1} by replacing the branch $B := (\langle x_0, x_1, x_2, \dots, n : (\Gamma, une) \rangle, \prec_B, min_B)$ either by the branch $B_1 := (\langle x_0, x_1, x_2, \dots, n : (\Gamma, une), n : (\Gamma_1, une_1) \rangle, \prec_{B_1}, min_{B_1})$, or the branch $B_2 := (\langle x_0, x_1, x_2, \dots, n : (\Gamma, une), n : (\Gamma_2, une_2) \rangle, \prec_{B_2}, min_{B_2})$, and so on. The symbol ‘ \mid ’ indicates the occurrence of a split in the branch, i.e., a non-deterministic choice of possible outcomes, each of which needs to be explored. It is worth to mention that after applying a static rule on $n : (\Gamma, une)$, the leaf nodes of all the new branches keep the same label n .

In what follows, we show the rules for Boolean connectives and the operators (\Box, \Diamond) . We also show two stopping conditions, namely, (Empty) and (Contradiction). We chose to omit \prec_B and min_B to lighten these rules. The crucial detail to remember is that they do not change after applying the rules below, i.e., $\prec_{B_i} = \prec_B$ and $min_{B_i} = min_B$ for all resulting branches. The symbol \cup is the union of two sets. The symbol \uplus represents the union between disjoint sets.

$$\begin{array}{ll}
\text{(Contradiction)} \quad \frac{n : (\{\alpha, \neg\alpha\} \uplus \Sigma, une)}{(\mathbf{X})} & \text{(Empty)} \quad \frac{n : (\emptyset, \emptyset)}{(\checkmark)} \\
\\
(\wedge) \quad \frac{n : (\{\alpha_1 \wedge \alpha_2\} \uplus \Sigma, une)}{n : (\{\alpha_1, \alpha_2\} \cup \Sigma, une)} & (\vee) \quad \frac{n : (\{\alpha_1 \vee \alpha_2\} \uplus \Sigma, une)}{n : (\{\alpha_1\} \cup \Sigma, une) \mid n : (\{\alpha_2\} \cup \Sigma, une)} \\
\\
(\Box) \quad \frac{n : (\{\Box\alpha_1\} \uplus \Sigma, une)}{n : (\{\alpha_1, \Box\alpha_1\} \cup \Sigma, une)} & (\Diamond) \quad \frac{n : (\{\Diamond\alpha_1\} \uplus \Sigma, une)}{n : (\{\alpha_1\} \cup \Sigma, une) \mid n : (\{\Box\Diamond\alpha_1\} \cup \Sigma, une)}
\end{array}$$

Before introducing the rule for the non-monotonic operator \Diamond , we discuss first-hand the notion of *fulfilment* for classical and non-monotonic eventualities. Following Reynolds’ tableau [Rey16b], let an eventuality $\Diamond\alpha$ be in a node with a label n . If the sentence α appears in a proper successor node x with the label $m \geq n$, we say that $\Diamond\alpha$ at the position n is *fulfilled* in m . In a similar fashion, we define the fulfilment for non-monotonic eventualities as follows:

Definition 5.3 (Fulfilment of non-monotonic eventualities). *Let a non-monotonic eventuality $\Diamond\alpha$ be in a node with a label n in a branch B . If α appears in a proper successor node x with a label $m \geq n$, and $(n, m) \in min_B$, we say $\Diamond\alpha$ at the position n is fulfilled in m .*

The truth value $\Diamond\alpha$ in a temporal state n depends on α being true on a future temporal state m and m being minimal to all temporal states that come after n w.r.t. \prec_B . We say m is minimal to n as shorter way to say that m is minimal to all temporal states that come after n . Unfulfilled non-monotonic eventualities in a node x with the label n are stored in the set $une \stackrel{\text{def}}{=} \{(n_1, \Diamond\alpha_1), (n_2, \Diamond\alpha_2), \dots\}$, each pair $(n_k, \Diamond\alpha_k)$ represents a non-monotonic eventuality $\Diamond\alpha_k$ at a position n_k that needs to be fulfilled. Therefore each node x has three components: n is a label indicating the temporal state, Γ is the set of sentences within the node and une is the set of non-monotonic eventualities at x that need to be fulfilled. With all of our notions introduced, here is the rule for the \Diamond operator:

$$(\Diamond) \frac{n : (\{\Diamond\alpha_1\} \uplus \Sigma, une), \prec_B, \min_B}{n : (\{\alpha_1\} \cup \Sigma, une), \prec_B, \min_B \cup \{(n, n)\} \mid n : (\Sigma, une \cup \{(n, \Diamond\alpha_1)\}), \prec_B, \min_B}$$

For the rule (\Diamond) , we explore two outcomes. The first outcome is when the non-monotonic eventuality $\Diamond\alpha_1$ at n is fulfilled in n . We then add α_1 to the set of sentences Γ of the leaf node and add $(n, n) \in \min$ of the branch. The second outcome is when $\Diamond\alpha_1$ is not fulfilled in n , then we add the pair to $(n, \Diamond\alpha_1)$ to une of the leaf node as a non-monotonic eventuality that needs to be fulfilled. Example 5.4 shows the application of (\Diamond) rule.

Example 5.4. Let a branch B have \prec_B, \min_B and a leaf node $5 : (\{p, q, \Box(p \wedge q), \Diamond r\}, \emptyset)$. After applying (\Diamond) rule on $\Diamond r$, we have two new branches B_1 and B_2 . The branch B_1 has a leaf node where the sentence r is in Γ of the leaf node and $(5, 5) \in \min_{B_1}$. The branch B_2 has $(5, \Diamond r)$ in une of the leaf node.

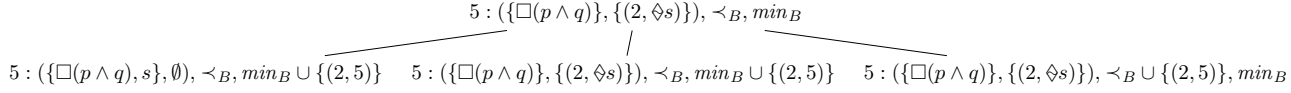
$$\begin{array}{c} 5 : (\{p, q, \Box(p \wedge q), \Diamond r\}, \emptyset), \prec_B, \min_B \\ \swarrow \quad \searrow \\ 5 : (\{p, q, \Box(p \wedge q), r\}, \emptyset), \prec_B, \min_B \cup \{(5, 5)\} \quad 5 : (\{p, q, \Box(p \wedge q)\}, \{(5, \Diamond r)\}), \prec_B, \min_B \end{array}$$

The next static rule we discuss is the rule (une) . Let n, n' be two labels such that $n' < n$, for each label n and a pair $(n', \Diamond\alpha_1)$, the rule (une) is applied one and only one time. The rule goes as follows:

$$\begin{array}{c} (une) \frac{n : (\Gamma, \{(n', \Diamond\alpha_1)\} \uplus U), \prec_B, \min_B}{n : (\{\alpha_1\} \cup \Gamma, U), \prec_B, \min_B \cup \{(n', n)\} \mid} \\ \hline n : (\Gamma, \{(n', \Diamond\alpha_1)\} \cup U), \prec_B, \min_B \cup \{(n', n)\} \mid \\ \hline n : (\Gamma, \{(n', \Diamond\alpha_1)\} \cup U), \prec_B \cup \{(n', n)\}, \min_B \end{array}$$

For the rule (une) , we explore three outcomes. The first outcome is when $\Diamond\alpha_1$ at the position n' is fulfilled at n . We remove $(n', \Diamond\alpha_1)$ from une , then we add α_1 in Γ of the leaf node and (n', n) in \min of the branch. In the second and third branches, we explore the outcome of $\Diamond\alpha_1$ not being fulfilled yet in n , we keep the pair $(n', \Diamond\alpha_1)$ on the leaves of two branches. The second branch explore the outcome of n being minimal to n' w.r.t. to \prec of the branch. We then add (n', n) to the \min of the branch. In the third branch, we explore the outcome of n not being minimal to n' w.r.t. \prec of the branch. It means that there exists a temporal state m' in the future of n' where m' is preferred to n w.r.t. to \prec of the branch, we add the pair (n', n) in \prec of the branch to represent this case. It is worth to mention that the rule (une) does not apply when the label of the node n is the same as $(n, \Diamond\alpha_1)$. The reason behind this is that we have already explored the case when the eventuality is fulfilled in n thanks to (\Diamond) rule. Example 5.5 shows the application of (une) rule.

Example 5.5. Let a branch B have \prec_B , \min_B and a leaf node $5 : (\{\Box(p \wedge q)\}, \{(2, \Diamond s)\})$. In this leaf node, we have $\Gamma = \{\Box(p \wedge q)\}$ and $\text{une} = \{(2, \Diamond s)\}$. After the application of une on $(2, \Diamond s)$, we have three branches B_1 , B_2 and B_3 . B_1 has the sentence s in Γ of its leaf node, it has also $(2, 5)$ in \min_{B_1} . B_2 keeps $(2, \Diamond s)$ in the une of its leaf node, with $(2, 5) \in \min_{B_2}$. B_3 keeps also $(2, \Diamond s)$ in une of its leaf node, with $(2, 5) \in \prec_{B_3}$.

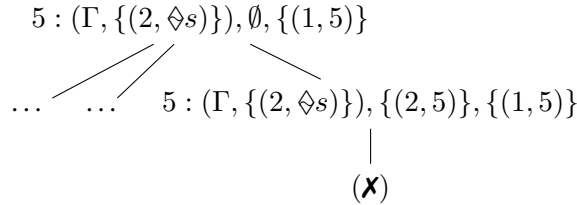


With the (une) and (\Diamond) introduced, we need to check the consistency of \prec of all the new branches. We apply this check each time we apply (une) or (\Diamond) rule. Let $B := (\langle x_0, x_1, x_2, \dots \rangle, \prec_B, \min_B)$ be a branch, the rule goes as follows:

- (\prec -inconsistency) : if $(n, n') \in \min_B$ and there exists $n'' \geq n$ s.t. $(n'', n') \in \prec_B$, then the branch is crossed (\times).

In a branch B , if $(n, n') \in \min_B$, then we are currently exploring a branch where n' is minimal to n w.r.t. \prec_B . Therefore there should be no $n'' \geq n$ where $(n'', n') \in \prec_B$. Each time we explore a branch where this inconsistency arises, we close the branch.

Example 5.6. Let B be a branch where \prec_B is empty, \min_B has $(1, 5)$ in it, and a leaf node $5 : (\Gamma, \{(2, \Diamond s)\})$. After applying une rule on $(2, \Diamond s)$, we have three branches B_1 , B_2 and B_3 . The relation \prec_{B_1} is empty, and \min_{B_1} has the pairs $(1, 5)$ and $(2, 5)$. In this case, there is no inconsistency w.r.t. \prec_{B_1} so far. The same goes for B_2 . However, we add $(2, 5)$ to \prec_{B_3} . Since we already have $(1, 5) \in \min_{B_3}$, then we cannot have $(2, 5) \in \prec_{B_3}$. We close B_3 .



In a branch B of a tree \mathcal{T} with a leaf node x_i , after applying every static rule aforementioned (the application order of these rules is non-deterministic) that can be applied, all leaf nodes of the generated branches contain only sentences of the form p , $\neg p$ or $\Box \alpha$ in their Γ . When no more static rules can be applied in a node, this node is called a *state-labelled node*. State-labelled nodes mark the full expansion of all sentences that hold in a state n .

5.1.2 Dynamic rules

Once we are in a state-labelled node, in order to go from a temporal state to the next, we need a transition rule (a rule to go from a temporal state n to the next $n+1$). In a branch B with a leaf state-labelled node, the rule (Transition) goes the following way:

$$(\text{Transition}) \quad \frac{n : (\{\Box \alpha_1, \Box \alpha_2, \Box \alpha_3, \dots, \Box \alpha_k\} \uplus \Sigma, \text{une}), \prec_B, \min_B}{n+1 : (\{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_k\}, \text{une}), \prec_B, \min_B}$$

After the (Transition) rule is applied to a state-labelled node $n : (\Gamma, une)$, we add a node with the label $n + 1$. It marks the start of a new temporal state $n + 1$. We carry over to $n + 1$ only sentences within the scope of $\bigcirc\alpha_i$ sentences. The set une gets transferred as well to the next temporal state. Any pair $(n', \Diamond\alpha_1) \in une$ remaining in the state node with the label n indicates that the current branch explores an outcome where the non-monotonic eventuality $\Diamond\alpha_1$ at n' is not yet fulfilled in n . Therefore, these non-monotonic eventualities need to be fulfilled in $n'' \geq n + 1$.

Before applying the (Transition) rule, we need to add a set of checks to prevent branches from expanding indefinitely. These checks are called (Loop) and (Prune) rules. These rules, together with the (Transition) rule, are called *dynamic rules*.

Let $B := (\langle x_0, x_1, x_2, \dots, v \rangle, \prec_B, min_B)$ be a branch where v is a state-labelled node $n : (\Gamma_v, une_v)$. Let u be the last state-labelled node $n - 1 : (\Gamma_u, une_u)$ that comes before v in the branch B . Before applying the transition rule on v , we check for these rules:

- (Loop) Let v be a state-labelled node such that it has at least one sentence of the form $\bigcirc\Box\alpha_{bool}$ in Γ_v but has no $\bigcirc\alpha_{bool}, \bigcirc\Diamond\beta, \bigcirc\Diamond\beta$ in Γ_v and $une_v = \emptyset$. If for all $\bigcirc\Box\alpha_{bool}$ in Γ_v , there exists $u < s \leq v$ such that $\Box\alpha_{bool} \in \Gamma_s$, then the branch B is ticked (\checkmark).

Notice that once an eventuality is fulfilled, it does not appear any longer in the successors of the node. In this case, we say that the sentence is *consumed*. On the other hand, sentences of the form $\Box\alpha_{bool}$ never get consumed and get replicated indefinitely. Once a branch has no eventuality left, $\Box\alpha_{bool}$ sentences give rise to an infinite tableau with repetitive nodes. Nevertheless, we can represent this by looping nodes of the last temporal state. We can, in this case, stop the branch from ever going infinite. The (Loop) rule states that when the leaf state node v has no eventualities (classical or non-monotonic), has only $\bigcirc\Box\alpha_{bool}$ as sentences with the pattern \bigcirc , and each $\bigcirc\Box\alpha_{bool}$ is a result from applying the \Box rule to a node in B with label n , the branch is ticked and marked as a successful branch.

- (Prune) Let $u < v$ be two consecutive state-labelled nodes s.t. $\Gamma_v = \Gamma_u$ and $une_v = une_u$ and such that there is at least one eventuality in x_u (either $\bigcirc\Diamond\beta \in \Gamma_u$ or $(n', \Diamond\beta) \in une_u$), then the branch is crossed (\times).

The (Prune) rule states that when the last two state nodes u and v have the same set of classical and non-monotonic eventualities that need to be fulfilled, and there is at least one eventuality in u , the branch is then crossed and marked as an unsuccessful branch. Any branch that does not fulfill at least one eventuality between the current and the last temporal state is closed, to prioritize the exploration of branches that fulfill one or more eventuality of the last temporal state. If neither (Prune or (Loop)) apply on v , we apply the (Transition) rule on the node v . Note that the (Loop) and (Prune) rules are fundamentally different from the ones proposed in Reynolds' tableau [Rey16a]. These rules are tailored to the restrictions of the fragment \mathcal{L}_1 , in particular, the restriction of not allowing temporal sentences inside the \Box operator. We argue that when eventualities (either classical or non-monotonic) are not infinitely replicated inside *globally* operators, we only need to check the current state node with the last one that comes beforehand. It is the reason why we also omit also the operator \mathcal{U} , since the right part of a \mathcal{U} -sentence can also replicate eventualities.

Once we are in a state-labelled node, we check for the loop and prune within the branch before applying the transition rule. If the transition rule is applied on a state node with a label n , we obtain a new node with the label $n + 1$. We can then expand the tree from this node by applying static rules until we find ticked branches (thanks to the (Empty) rule), closed branches (thanks to the (Contradiction) or (\prec -inconsistency) rules), or branches with a state node that has the label $n + 1$. We then repeat the cycle between static and dynamic rules. We can see that the tableau method does not go indefinitely. Thanks

to prune rule, we close any branch (\times) that does not fulfil any eventuality in the current temporal state. Anytime we apply a transition rule (from n to $n + 1$), we need to fulfil at least one eventuality in n . Therefore, as long as a branch is not closed with prune rule, eventuality sentences (either classical or non-monotonic) get consumed one by one over the execution of the method. Thus any branch that is not closed with prune has no eventualities left to fulfil. Note that if a branch contains at least one sentence of the form $\Box\alpha_{bool}$, it is then ticked thanks to the loop rule ($\Box\alpha_{bool}$ sentences do not get consumed). Otherwise, it is ticked thanks to the empty rule. Therefore any tableau \mathcal{T} for a sentence in \mathcal{L}_1 is a *saturated* tableau.

5.2 Soundness and completeness

We show the soundness and completeness of the tableau method for sentences in \mathcal{L}_1 . We based the proofs of this section on the soundness and completeness of one-pass tree-shaped tableaux proposed by Reynolds [Rey16a].

5.2.1 Soundness

Here we prove that the tableau method is sound, that is, when a tableau \mathcal{T} of a sentence $\alpha \in \mathcal{L}_1$ has a successful branch, then α is satisfiable. As a first step, we show that we can extract an interpretation $I \in \mathcal{I}$ from the successful branch. Let $B := (\langle x_0, x_1, x_2, \dots, x_n, (\checkmark) \rangle, \prec_B, \min_B)$ be a successful branch of a tableau \mathcal{T} for α , the sequence of nodes contains normal and state-labelled nodes. Each state-labelled node, denoted by x_{j_i} , within this sequence has a distinct label i . Figure 5.1 shows an example of the branch B .

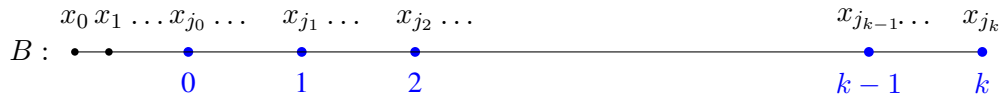


Figure 5.1: Illustration of the branch B .

From the aforementioned branch B , we can build an interpretation $I_B = (V, \prec)$. In this section, k denotes the label of the last state node. The function V is defined as follows:

$$V(i) := \begin{cases} \{p \in \mathcal{P} \mid p \in \Gamma_{x_{j_i}}\}, & \text{if } 0 \leq i \leq k; \\ V(k), & \text{otherwise.} \end{cases}$$

The ordering relation \prec is defined as follows $\prec := \{(n, n') \mid (n, n') \in \prec_B\}$. We can see that the resulting \prec is irreflexive, since there is no rule application that adds (n, n) to \prec_B . The relation \prec does not contradict the transitive property of strict-partial orders, since for all (n_1, n_2) and (n_2, n_3) in \prec_B , there is no rule application that adds $(n_3, n_1) \in \prec_B$ knowing that n_3 temporally comes after n_1 . Finally, since \prec_B has no infinitely descending chains, then we can conclude that \prec preserves the well-foundedness condition over \mathbb{N} . Therefore, the interpretation I_B is a preferential interpretation in \mathcal{I} .

With the model construction introduced, we can move on to the second part of the proof of soundness. We need to show that the model I satisfies the sentence α . In order to do so, we introduce a mapping function, denoted by Δ_B , that links each time point $i \in \mathbb{N}$ to a set of sentences that are true in said i . These sentences come from the branch B . Depending on how the branch is ticked, the function Δ_B is defined in the following way.

If the branch was ticked with the empty rule:

$$\Delta_B(i) := \begin{cases} \bigcup_{x_0 \leq x \leq x_{j_0}} \Gamma_x, & \text{if } i = 0; \\ \bigcup_{x_{j_{i-1}} < x \leq x_{j_i}} \Gamma_x, & \text{if } 1 \leq i \leq k-1; \\ \emptyset, & \text{otherwise.} \end{cases}$$

If the branch was ticked with the loop rule:

$$\Delta_B(i) := \begin{cases} \bigcup_{x_0 \leq x \leq x_{j_0}} \Gamma_x, & \text{if } i = 0; \\ \bigcup_{x_{j_{i-1}} < x \leq x_{j_i}} \Gamma_x, & \text{if } 1 \leq i \leq k; \\ \Delta_B(k), & \text{otherwise.} \end{cases}$$

For a time point $0 \leq i \leq k$, $\Delta_B(i)$ contains the set of all sentences in Γ of the node between the two consecutive state nodes $x_{j_{i-1}}$ and x_{j_i} , $x_{j_{i-1}}$ not included. If B is ticked thanks to the empty rule, then $\Delta_B(i)$ is empty for all $i \geq k$. If B is ticked thanks to the loop rule, then $\Delta_B(i)$ has the same set of sentences as $\Delta_B(k)$ for all $i \geq k$. We can show next that if a sentence α_1 is in $\Delta_B(i)$, then $I_B, i \models \alpha_1$. In what follows, let B be a successful branch of a tableau \mathcal{T} , let k be the label of the last state node in B , and let I_B, Δ_B be the interpretation and the mapping function of sentences extracted from B .

Lemma 5.7. *Let B be a successful branch, and $i \in \mathbb{N}$. If $\bigcirc\alpha_1 \in \Delta_B(i)$, then $\alpha_1 \in \Delta_B(i+1)$.*

Proof. Let B be a ticked branch of the tableau, k be the label of the last node and $i \in \mathbb{N}$. We discuss two possibilities:

- When the branch B is ticked with empty rule. We can see that when $i \geq k$, $\Delta_B(i) = \{\}$ and therefore $\bigcirc\alpha_1 \notin \Delta_B(i)$. We also know that since $\Delta_B(k) = \{\}$, then there is no $\bigcirc\alpha_1 \in \Gamma_{x_{j_{k-1}}}$ (otherwise, $\Delta_B(k)$ would at least contain α_1). Furthermore, there is no static rule that removes $\bigcirc\alpha_1$, we can conclude that there is no $\bigcirc\alpha_1 \in \Delta_B(k-1)$.

Otherwise, in the case of $0 \leq i < k-1$, if $\bigcirc\alpha_1 \in \Delta_B(i)$, then $\bigcirc\alpha_1 \in \Gamma_x$ where $x_{j_{i-1}} < x \leq x_{j_i}$. Since there is no static rule that removes $\bigcirc\alpha_1$, we have $\bigcirc\alpha_1 \in \Gamma_{x_{j_i}}$. Furthermore, after applying the transition rule on the node x_{j_i} , we have $\alpha_1 \in \Gamma_{x_{j_{i+1}}}$. Thus, we have $\alpha_1 \in \Delta_B(i+1)$.

- When the branch B is ticked with loop rule. In the case of $0 \leq i < k$, the proof is analogous to the case of empty rule. When $i = k$, if $\bigcirc\alpha_1 \in \Delta_B(k)$, then $\bigcirc\alpha_1$ is subsequently in $\Gamma_{x_{j_k}}$. Since B is ticked with loop, then α_1 is a sentence of the form $\Box\alpha_{bool}$ and $\Box\alpha_{bool} \in \Gamma_x$ ($x_{j_{k-1}} < x \leq x_{j_k}$) and therefore $\Box\alpha_{bool} \in \Delta_B(k)$. Moreover, we have $\Delta_B(k) = \Delta_B(k+1)$. Therefore, we have $\Box\alpha_{bool} \in \Delta_B(k+1)$ and thus $\alpha_1 \in \Delta_B(k+1)$.

In the case where $i \geq k$. If $\bigcirc\alpha_1 \in \Delta_B(i)$, then $\bigcirc\alpha_1 \in \Delta_B(k-1)$. As mentioned before, since $\bigcirc\alpha_1 \in \Delta_B(k-1)$, then α_1 is $\Box\alpha_2$ and $\Box\alpha_2 \in \Delta_B(k-1)$. Since $\Box\alpha_2 \in \Delta_B(k-1)$, then $\Box\alpha_2 \in \Delta_B(i+1)$ and therefore $\alpha_1 \in \Delta_B(i+1)$.

□

Lemma 5.8. *Let B be a successful branch, and $i \in \mathbb{N}$. If $\Box\alpha_1 \in \Delta_B(i)$, then for all $f \geq i$, we have $\{\alpha_1, \Box\alpha_1, \bigcirc\Box\alpha_1\} \subseteq \Delta_B(f)$.*

Proof. Let B be a ticked branch of the tableau, k be the label of the last node and $i \in \mathbb{N}$.

For all $0 \leq i \leq k$, whenever $\Box\alpha_1 \in \Delta_B(i)$, then both α_1 and $\bigcirc\Box\alpha_1$ is in $\Delta_B(i)$. By Lemma 5.7, since $\bigcirc\Box\alpha_1 \in \Delta_B(i)$, then we have $\Box\alpha_1 \in \Delta_B(i+1)$. By successive applications of Lemma 5.7, we

have $\{\alpha_1, \Box\alpha_1, \Box\Box\alpha_1\} \subseteq \Delta_B(f)$ for all $i \leq f \leq k$. Note that in the case of a branch ticked with empty rule, since $\Delta_B(k) = \{\}$, $\Box\alpha_1$ cannot be in any $\Delta_B(i)$ where $0 \leq i \leq k$. In other words, if a branch contains $\Box\alpha_1$, it can only be ticked with loop rule.

Since $\{\alpha_1, \Box\alpha_1, \Box\Box\alpha_1\} \subseteq \Delta_B(k)$, and for all $f \geq k$, we have $\Delta_B(f) = \Delta_B(k)$, then $\{\alpha_1, \Box\alpha_1, \Box\Box\alpha_1\} \subseteq \Delta_B(f)$. Thus, the lemma holds when $0 \leq i \leq k$.

In the case of $i > k$, since $\Box\alpha_1 \in \Delta_B(i)$ and $\Delta_B(i) = \Delta_B(k-1)$. Thanks to \Box -rule, $\{\alpha_1, \Box\alpha_1, \Box\Box\alpha_1\} \subseteq \Delta_B(k-1)$. Thus, we have $\{\alpha_1, \Box\alpha_1, \Box\Box\alpha_1\} \subseteq \Delta_B(f)$ for all $f \geq k$ and subsequently $\{\alpha_1, \Box\alpha_1, \Box\Box\alpha_1\} \subseteq \Delta_B(f)$ for all $f \geq i$. \square

Lemma 5.9. *Let B be a successful branch, and $i \in \mathbb{N}$. If $\Diamond\alpha_1 \in \Delta_B(i)$, then there exists $d \geq i$ s.t. $\alpha_1 \in \Delta_B(d)$ and for all $i \leq f < d$, we have $\{\Diamond\alpha_1, \Box\Diamond\alpha_1\} \subseteq \Delta_B(f)$.*

Proof. Let B be a ticked branch of the tableau, k be the label of the last node and $i \in \mathbb{N}$. We discuss two possibilities:

- When the branch B is ticked with empty rule. In the case of $0 \leq i \leq k-1$, whenever $\Diamond\alpha_1 \in \Delta_B(i)$, then either α_1 is in $\Delta_B(i)$ or $\Box\Diamond\alpha_1$ is in $\Delta_B(i)$. If $\alpha_1 \in \Delta_B(i)$, the lemma holds. Otherwise, by Lemma 5.7, if $\Box\Diamond\alpha_1 \in \Delta_B(i)$ then $\Diamond\alpha_1 \in \Delta_B(i+1)$. By successive applications of Lemma 5.7, $\{\Diamond\alpha_1, \Box\Diamond\alpha_1\}$ is in $\Delta_B(f)$ for $i \leq f \leq k-1$, unless we find $i \leq d \leq f$ with $\alpha_1 \in \Delta_B(d)$. It remains to show that there is a time point d where $\alpha_1 \in \Delta_B(d)$. Since the branch is closed thanks to the empty rule, it means that $\Box\Diamond\alpha_1 \notin \Delta_B(k-1)$. Therefore, there is a state $i \leq d \leq k-1$ where $\alpha_1 \in \Delta_B(d)$.
- When the branch B is ticked with loop rule and in the case of $0 \leq i \leq k$, the proof is analogous to the case of empty rule (notice that $\Box\Diamond\alpha_1 \notin \Delta_B(k)$ also in the case of branches ticked with loop). In the case of $i > k$, since $\Diamond\alpha_1 \in \Delta_B(i)$, then we have $\Diamond\alpha_1 \in \Delta_B(k-1)$. Furthermore, since the branch is ticked with loop rule, we know that $\Box\Diamond\alpha_1 \notin \Delta_B(k)$. Therefore $\alpha_1 \in \Delta_B(k)$, thus $\alpha_1 \in \Delta_B(i)$. \square

Lemma 5.7 to 5.9 are analogous to Reynolds' method [Rey16a]. We show that a similar property can be obtained for \Diamond sentences.

Proposition 5.10. *Let B be a successful branch. If $(i, i') \in \min_B$, then there is no i'' such that $i \leq i''$ and $(i'', i') \in \prec_B$.*

Proof. Let B be a successful branch s.t. $(i, i') \in \min_B$. Since the branch is successful, then it is not closed with \prec -inconsistency and therefore there is no i'' such that $i \leq i''$ and $(i'', i') \in \prec_B$. \square

Lemma 5.11. *Let B be a successful branch and $0 \leq i \leq k$. If $\Diamond\alpha_1 \in \Delta_B(i)$, then there exists $d \geq i$ s.t. $(i, d) \in \min_B$ and $\alpha_1 \in \Delta_B(d)$.*

Proof. Let B be a ticked branch of the tableau, k be the label of the last state node and $i \in \mathbb{N}$. We discuss two possibilities:

- When the branch B is ticked with empty rule, whenever $\Diamond\alpha_1 \in \Delta_B(i)$, then we have $0 \leq i \leq k-1$. Since $\Diamond\alpha_1 \in \Delta_B(i)$, then $\Diamond\alpha_1 \in \Gamma_x$ where $x_{j_{i-1}} < x_k \leq x_{j_i}$. Let x be the node where we apply the rule (\Diamond) on $\Diamond\alpha_1$, then we either have α_1 in Γ of the next node with $(i, i) \in \min_B$ or we have $(i, \Diamond\alpha_1) \in \text{une}$ of the next node. If α_1 is in Γ of the next node, then the lemma holds. If

$(i, \Diamond \alpha_1) \in \text{une}$ of the next node, then we find $(i, \Diamond \alpha_1) \in \text{une}_{x_{j_i}}$. Thanks to the transition rule, we have $(i, \Diamond \alpha_1) \in \text{une}_{x_{j_{i+1}}}$ (transition rule propagates all unfulfilled eventualities to the next temporal state). By applying the rule une on a node with the label $i + 1$, then we either have α_1 in Γ of the next node with $(i, i + 1) \in \text{min}_B$ or we have $(i, \Diamond \alpha_1) \in \text{une}$ (the two remaining branches) of the next node. In a similar way as in i , we can conclude that either $\alpha_1 \in \Delta_B(i + 1)$ with $(i, i + 1) \in \text{min}_B$ (the lemma holds) or $(i, \Diamond \alpha_1) \in \text{une}_{x_{j_{i+1}}}$. Without loss of generality, $(i, \Diamond \alpha_1)$ is in $\text{une}_{x_{j_f}}$ for $i \leq f \leq k - 1$ unless we find $i \leq d \leq f$ with $\alpha_1 \in \Delta_B(d)$ and $(i, d) \in \text{min}_B$. Since the branch is ticked thanks to the empty rule, it means that $(i, \Diamond \alpha_1) \notin \text{une}_{x_{j_{k-1}}}$. Therefore, there is a state $i \leq d \leq k - 1$ where $\alpha_1 \in \Delta_B(d)$ with $(i, d) \in \text{min}_B$.

- When the branch B is ticked with loop rule, the proof is analogous to the case of the empty rule (notice that we also have $(i, \Diamond \alpha_1) \notin \text{une}_{x_{j_k}}$).

□

Theorem 5.12. *Let B be a ticked branch from a saturated tableau, and $I_B = (V, \prec)$ be the model built from the branch B . For all $\alpha \in \mathcal{L}_1$, for all $i \geq 0$, if $\alpha \in \Delta_B(i)$ then $I_B, i \models \alpha$.*

Proof. We prove this lemma using structural induction on the size of the sentence α . Let B be a successful branch for a tableau \mathcal{T} , and $I_B = (V, \prec)$ be the model built from B .

- $\alpha = p$. Let $p \in \Delta_B(i)$. By construction of the model I_B , we have $p \in V(i)$. Therefore, we have $I_B, i \models p$.
- $\alpha = \neg p$. Let $\neg p \in \Delta_B(i)$. Since B is a ticked branch, then it was not closed with the contradiction rule, therefore we have $p \notin V(i)$. Therefore, we have $I_B, i \models \neg p$.
- $\alpha = \alpha_1 \wedge \alpha_2$. Let $\alpha_1 \wedge \alpha_2 \in \Delta_B(i)$. By \wedge -rule, we have $\alpha_1, \alpha_2 \in \Delta_B(i)$. By induction hypothesis on α_1, α_2 , we have $I_B, i \models \alpha_1$ and $I_B, i \models \alpha_2$. Thus, we have $I_B, i \models \alpha_1 \wedge \alpha_2$.
- $\alpha = \alpha_1 \vee \alpha_2$. Let $\alpha_1 \vee \alpha_2 \in \Delta_B(i)$. By \vee -rule, we either have α_1 or α_2 in $\Delta_B(i)$. Suppose that $\alpha_1 \in \Delta_B(i)$, by induction hypothesis on α_1 , we have $I_B, i \models \alpha_1$. Therefore, we have $I_B, i \models \alpha_1 \vee \alpha_2$. Same reasoning applies when $\alpha_2 \in \Delta_B(i)$.
- $\alpha = \bigcirc \alpha_1$. Let $\bigcirc \alpha_1 \in \Delta_B(i)$. Thanks to Lemma 5.7, we have $\alpha_1 \in \Delta_B(i + 1)$. By induction hypothesis on α_1 , we have $I_B, i + 1 \models \alpha_1$. Therefore, we have $I_B, i \models \bigcirc \alpha_1$.
- $\alpha = \Box \alpha_1$. Let $\Box \alpha_1 \in \Delta_B(i)$. Thanks to Lemma 5.8, we have $\alpha_1 \in \Delta_B(f)$ for all $f \geq i$. By induction hypothesis on α_1 , we have $I_B, f \models \alpha_1$ for all $f \geq i$. Therefore, we have $I_B, i \models \Box \alpha_1$.
- $\alpha = \Diamond \alpha_1$. Let $\Diamond \alpha_1 \in \Delta_B(i)$. Thanks to Lemma 5.9, we have $\alpha_1 \in \Delta_B(d)$ for some $d \geq i$. By induction hypothesis on α_1 , we have $I_B, d \models \alpha_1$. Therefore, we have $I_B, i \models \Diamond \alpha_1$.
- $\alpha = \Diamond \alpha_1$. Let $\Diamond \alpha_1 \in \Delta_B(i)$. Depending on where i is, we have two cases:
 - In the case of $i > k$, since $\Diamond \alpha_1 \in \Delta_B(i)$, then the branch is ticked with loop and we have $\Diamond \alpha_1 \in \Delta_B(k)$. Furthermore, since the branch is ticked with loop rule, we know that $(i, \Diamond \alpha_1) \notin \text{une}_{x_{j_k}}$. With $\Diamond \alpha_1 \in \Delta_B(k)$ and $(i, \Diamond \alpha_1) \notin \text{une}_{x_{j_k}}$, we can infer that $\alpha_1 \in \Delta_B(k)$, thus $\alpha_1 \in \Delta_B(i)$. Furthermore, since $\prec := \prec_B$, and there is no $f \geq i$ such $(f, i) \in \prec_B$, then $i \in \text{min}_{\prec}(i)$, and therefore, $I_B, i \models \Diamond \alpha_1$.

- $0 \leq i \leq k$. Thanks to Lemma 5.11, there exists $d \geq i$ s.t. $\alpha_1 \in \Delta_B(d)$ and $(i, d) \in \min_B$. By induction hypothesis on α_1 , we have $I_B, d \models \alpha_1$. Thanks to Proposition 5.10, there is no $i \leq f \leq k$ where $(f, d) \in \prec_B$ and therefore there is no $i \leq f \leq k$ where $(f, d) \in \prec$. Furthermore, by the construction of the model I_B , there is no $f \geq k$ where $(f, d) \in \prec$. Therefore, we have $d \in \min_{\prec}(i)$. Thus, we have $I_B, i \models \Diamond \alpha_1$.

□

Let $\alpha \in \mathcal{L}_1$, B be a ticked branch from a saturated tableau for α , $I_B = (V, \prec)$ be a model built from B . Since we have $\alpha \in \Delta_B(0)$, then we have $I_B, 0 \models \alpha$. Hence, we have proved the soundness of the tableau method for \mathcal{L}_1 sentences.

5.2.2 Completeness

In order to prove the completeness of the tableau method for \mathcal{L}_1 sentences, we need to check whether a tableau \mathcal{T} for a satisfiable sentence α has a successful branch (✓). Recall that since the order of applying static rules is non-deterministic, multiple tableaux for the same sentence α can be obtained. The strong result we show in this section is the Theorem 5.13.

Theorem 5.13. *Let $\alpha \in \mathcal{L}_1$ be a satisfiable sentence of LTL^\sim . Then any tableau for α has a successful branch.*

Given an \mathcal{I} -satisfiable sentence α , we prove that any given tableau \mathcal{T} for α contains a successful branch B with a ticked node (✓). Since α is \mathcal{I} -satisfiable, there exists an interpretation $I \in \mathcal{I}$ such that $I, 0 \models \alpha$. The goal is to show that the interpretation I is represented in the tableau \mathcal{T} . Moreover, that said representation leads to a ticked node (✓).

In order to prove the aforementioned theorem, we use an intermediate sequence s that serves as a link between an interpretation I that satisfies the sentence α and a tableau \mathcal{T} for α . The sequence s is a tuple $s := (\langle x_0, x_1, x_2, \dots \rangle, \prec_s, \min_s)$ where each x_i is a pair (Γ, une) , \prec_s, \min_s are the set of constraints that the sequence s must follow in order to be coherent with \prec of the interpretation. The set \prec_s is not an ordering relation, it records instances of points of time not being minimal to other points of time w.r.t. the ordering relation \prec . Remember that when we apply the *une* rule, we add a pair (n', n) to \prec in order to symbolize the outcome of n not being minimal to n' . The set \min_s records the instances of points of time being minimal to other points of time w.r.t. the ordering relation \prec .

We link each node of the sequence x_i to a time point $J(x_i)$ of the interpretation I and a labelled node $f(x_i)$ of the tableau \mathcal{T} . Depending on I , we can build the sequence s using the tableau, we then show the sequence s ends up with a tick (✓). We make sure that for each node x_i with the index time point $J(x_i)$ of the sequence, we have the following invariant:

$$Inv(x_i, J(x_i)) \left\{ \begin{array}{l} (1) \text{ For each } \alpha \in \Gamma_{x_i}, \text{ we have } I, J(x_i) \models \alpha. \\ (2) \text{ For each } (J_1, \Diamond \alpha_1) \in \text{une}_{x_i}, \text{ there exists } J_2 \geq J(x_i) \text{ where } J_2 \in \min_{\prec}(J_1) \text{ and } I, J_2 \models \alpha_1. \\ (3) \text{ For each } (J_1, J_2) \in \min_s, \text{ we have } J_2 \in \min_{\prec}(J_1). \\ (4) \text{ For each } (J_1, J_2) \in \prec_s, \text{ there exists } J_3 \geq J_1 \text{ s.t. } (J_3, J_2) \in \prec \\ \text{(in other words } J_2 \notin \min_{\prec}(J_1)). \end{array} \right.$$

We start by putting the root node $0 : (\{\alpha\}, \emptyset)$ with the index time point $J(x_0) := 0$ at the start of the sequence. For the first node x_0 with the index time point 0 (since there is no rule applied before

the root node, the sets min_s and \prec_s are empty at the start), we have $I, 0 \models \alpha$. Therefore the invariant $Inv(x_0, 0)$ holds. Suppose that the invariant holds up to x_i , and a rule was applied to x_i , we then add a new node x_{i+1} to the sequence depending on which outcome of the rule represents the interpretation I . We then move to the outcome node in the tableau, and see which rule is applied to it, and so on and so forth. Each time we add a new node x_{i+1} to the sequence s , we need to make sure that the invariant $Inv(x_{i+1}, J(x_{i+1}))$ holds. In general, the sequence will head from the parent node to a child node but it might occasionally jump backwards (only in the case of the parent being a prune node, more on that later). It is worth to point out that since we might be jumping back and forth between nodes of \mathcal{T} , each time we are adding a new node x_{i+1} to the sequence s , we are going to rename labels within the sets une_x , \prec_B and min_B by their respective indexed time points J . The function f links each node x_i of the sequence s to a labelled node $f(x_i)$ of the tableau \mathcal{T} . It is worth to mention that, since we are only renaming labels of other sets, then we have $\Gamma_{x_i} = \Gamma_{f(x_i)}$.

We discuss the case of each rule that is applied to x_i . Suppose that the sequence s is built up to x_i and the invariant holds for all the nodes in the sequence.

Rules (Empty) and (Loop):

If we end up with a ticked node in the sequence s , the theorem holds.

Rule (Contradiction):

If the sequence s is closed, then we have p and $\neg p$ in Γ_{x_i} . Since we have $Inv(x_i, J(x_i))$, then we $I, J(x_i) \models p$ and $I, J(x_i) \models \neg p$. This cannot happen in a interpretation $I \in \mathcal{I}$.

Rule (\wedge):

Suppose that the rule (\wedge) is applied to the sentence $\alpha_1 \wedge \alpha_2$ on the node $f(x_i)$ of the tableau \mathcal{T} . Let y be the child node of the node $f(x_i)$ in the branch. We have $\Gamma_y = (\Gamma_{f(x_i)} \setminus \{\alpha_1 \wedge \alpha_2\}) \cup \{\alpha_1, \alpha_2\}$. We define the next node in the sequence x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_y$, $une_{x_{i+1}} = une_{x_i}$, and the sets min_s , \prec_s remain unchanged. Since we have $Inv(x_i, J(x_i))$ and $\alpha_1 \wedge \alpha_2 \in \Gamma_{x_i}$, then $I, J(x_i) \models \alpha_1$ and $I, J(x_i) \models \alpha_2$. For the node x_{i+1} , we have $\Gamma_{x_{i+1}} = (\Gamma_{x_i} \setminus \{\alpha_1 \wedge \alpha_2\}) \cup \{\alpha_1, \alpha_2\}$ and $une_{x_{i+1}} = une_{x_i}$. Therefore the first and second conditions of $Inv(x_{i+1}, J(x_{i+1}))$ are met. Moreover, since min_s , \prec_s remain unchanged and we have $Inv(x_i, J(x_i))$, then the third and forth conditions of $Inv(x_{i+1}, J(x_{i+1}))$ are met. Consider that $J(x_{i+1}) = J(x_i)$, the invariant $Inv(x_{i+1}, J(x_{i+1}))$ holds.

We can see that by applying a static rule of the from ($\wedge, \vee, \Box, \Diamond$) on the node $f(x_i)$, we do not add in either une , \prec_B or min_B while applying these rules nor add a new non-monotonic eventuality to be fulfilled in the outcome nodes. In order to lighten the proof, we skip the check for the second, third and fourth conditions of Inv up until \Diamond and une rules.

Rule (\vee):

Suppose that the rule (\vee) is applied to the sentence $\alpha_1 \vee \alpha_2$ on the node $f(x_i)$ of the tableau \mathcal{T} . We obtain two children nodes y and z of $f(x_i)$. We have $\Gamma_y = (\Gamma_{f(x_i)} \setminus \{\alpha_1 \vee \alpha_2\}) \cup \{\alpha_1\}$ and $\Gamma_z = (\Gamma_{f(x_i)} \setminus \{\alpha_1 \vee \alpha_2\}) \cup \{\alpha_2\}$. Since we have $Inv(x_i, J(x_i))$, and $\alpha_1 \vee \alpha_2 \in \Gamma_{x_i}$, then we either have $I, J(x_i) \models \alpha_1$ or $I, J(x_i) \models \alpha_2$. Consider that $J(x_{i+1}) = J(x_i)$, we discuss two cases:

- Case 1: If $I, J(x_i) \models \alpha_1$, then we define the next node x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_y$ and $une_{x_{i+1}} = une_{x_i}$. We know that $\Gamma_{x_{i+1}} = (\Gamma_{x_i} \setminus \{\alpha_1 \vee \alpha_2\}) \cup \{\alpha_1\}$. Therefore for all $\gamma \in \Gamma_{x_{i+1}}$, we have

$I, J(x_i) \models \gamma$. Thus, the invariant $Inv(x_{i+1}, J(x_i))$ holds.

- Case 2: Otherwise, when $I, J(x_i) \models \alpha_2$, then we define the node x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_z$ and $une_{x_{i+1}} = une_{x_i}$. We know that $\Gamma_{x_{i+1}} = (\Gamma_{x_i} \setminus \{\alpha_1 \vee \alpha_2\}) \cup \{\alpha_2\}$. Therefore for all $\gamma \in \Gamma_{x_{i+1}}$, we have $I, J(x_i) \models \gamma$. Thus, the invariant $Inv(x_{i+1}, J(x_i))$ holds.

Rule (\Diamond):

Suppose that the rule (\Diamond) is applied to the sentence $\Diamond\alpha_1$ on the node $f(x_i)$ of the tableau \mathcal{T} . We obtain two children nodes y and z of $f(x_i)$. We have $\Gamma_y = (\Gamma_{f(x_i)} \setminus \{\Diamond\alpha_1\}) \cup \{\alpha_1\}$ and $\Gamma_z = (\Gamma_{f(x_i)} \setminus \{\Diamond\alpha_1\}) \cup \{\Diamond\alpha_1\}$. Since we have $Inv(x_i, J(x_i))$, and $I, J(x_i) \models \Diamond\alpha_1$, then we have $I, J(x_i) \models \alpha_1 \vee \Diamond\alpha_1$. Therefore, we either have $I, J(x_i) \models \alpha_1$ or $I, J(x_i) \models \Diamond\alpha_1$. Consider that $J(x_{i+1}) = J(x_i)$, we discuss two cases:

- Case 1: If $I, J(x_i) \models \alpha_1$, then we define the next node x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_y$ and $une_{x_{i+1}} = une_{x_i}$. We know that $\Gamma_{x_{i+1}} = (\Gamma_{x_i} \setminus \{\Diamond\alpha_1\}) \cup \{\alpha_1\}$. Therefore for all $\gamma \in \Gamma_{x_{i+1}}$, we have $I, J(x_i) \models \gamma$. Thus, the invariant $Inv(x_{i+1}, J(x_i))$ holds.
- Case 2: When $I, J(x_i) \models \Diamond\alpha_1$, then we define the next node x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_z$ and $une_{x_{i+1}} = une_{x_i}$. We know that $\Gamma_{x_{i+1}} = (\Gamma_{x_i} \setminus \{\Diamond\alpha_1\}) \cup \{\Diamond\alpha_1\}$. Therefore for all $\gamma \in \Gamma_{x_{i+1}}$, we have $I, J(x_i) \models \gamma$. Thus, the invariant $Inv(x_{i+1}, J(x_i))$ holds.

Rule (\Box):

Suppose that the rule (\Box) is applied to the sentence $\Box\alpha_1$ on the node $f(x_i)$ of the tableau \mathcal{T} . Let y be the child node of the node $f(x_i)$ in the branch. We have $\Gamma_y = (\Gamma_{f(x_i)} \setminus \{\Box\alpha_1\}) \cup \{\alpha_1, \Box\alpha_1\}$. We define the next node x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_y$ and $une_{x_{i+1}} = une_{x_i}$ and $I, J(x_i) \models \Box\alpha_1$, then we have $I, J(x_i) \models \alpha_1 \wedge \Box\alpha_1$. Therefore, we have $I, J(x_i) \models \alpha_1$ and $I, J(x_i) \models \Box\alpha_1$. We know that $\Gamma_{x_{i+1}} = (\Gamma_{x_i} \setminus \{\Box\alpha_1\}) \cup \{\alpha_1, \Box\alpha_1\}$. Therefore for all $\gamma \in \Gamma_{x_{i+1}}$, we have $I, J(x_i) \models \gamma$. Consider that $J(x_{i+1}) = J(x_i)$, the invariant $Inv(x_{i+1}, J(x_i))$ holds.

Rule (\Diamond):

When the rule (\Diamond) is applied to $\Diamond\alpha_1$ on the node $f(x_i)$ of \mathcal{T} , we explore two outcomes. Let n be the label of the node $f(x_i)$ in the branch. In the first outcome, we have a child y with $\Gamma_y = (\Gamma_{f(x_i)} \setminus \{\Diamond\alpha_1\}) \cup \{\alpha_1\}$ and (n, n) in min of the branch. In the second outcome, we have a child node z with $\Gamma_z = (\Gamma_{f(x_i)} \setminus \{\Diamond\alpha_1\})$ and $une_z = une_{f(x_i)} \cup (n, \Diamond\alpha_1)$. Since we have $Inv(x_i, J(x_i))$, and $\Diamond\alpha_1 \in \Gamma_{x_i}$, then we have $I, J(x_i) \models \Diamond\alpha_1$. It means that there exists $J_1 \geq J(x_i)$ s.t. $J_1 \in min_{\prec}(J(x_i))$ and $I, J_1 \models \alpha_1$. Consider that $J(x_{i+1}) = J(x_i)$, we discuss two cases:

- Case 1: If $J_1 = J(x_i)$, then we have $I, J(x_i) \models \alpha_1$ and $J(x_i) \in min_{\prec}(J(x_i))$. We then define the next node x_{i+1} of the sequence with $\Gamma_{x_{i+1}} = \Gamma_y$, $une_{x_{i+1}} = une_{x_i}$ and add the pair $(J(x_i), J(x_i))$ to min_s . Notice that we swapped the labels of nodes with the position of their indexed time point $J(x_i)$, we will be using indexed time point J instead of labels throughout this proof. We know that $\Gamma_{x_{i+1}} = (\Gamma_{x_i} \setminus \{\Diamond\alpha_1\}) \cup \{\alpha_1\}$ with $I, J(x_i) \models \alpha_1$. Additionally, we have $min_s := min_s \cup \{(J(x_i), J(x_i))\}$ with $J(x_i) \in min_{\prec}(J(x_i))$. The sets $une_{x_{i+1}}, \prec_s$ remains unchanged. Therefore, the invariant $Inv(x_{i+1}, J(x_i))$ holds.
- Case 2: when $J_1 > J(x_i)$, then we define the next node x_{i+1} of the sequence with $\Gamma_{x_{i+1}} = \Gamma_z$, $une_{x_{i+1}} = une_{x_i} \cup \{(J(x_i), \Diamond\alpha_1)\}$. We also know that $J_1 > J(x_i)$ and $J_1 \in min_{\prec}(J(x_i))$ and

$I, J_1 \models \alpha_1$. Therefore, the second condition of $Inv(x_{i+1}, J(x_i))$ holds on the pair $(J(x_i), \Diamond\alpha_1)$. The sets min_s and \prec_s remain unchanged. The invariant $Inv(x_{i+1}, J(x_i))$ holds.

Rule (*une*):

When the rule (*une*) is applied on a pair $(n_1, \Diamond\alpha_1)$ in *une* of $f(x_i)$. Let n be the label of the node $f(x_i)$. Let x be the predecessor of x_i in s where the rule (\Diamond) was applied on $\Diamond\alpha_1$, let $J(x)$ be the indexed time point of x . Note that the label of $f(x)$ is n_1 . In the first outcome, we have a child y where $\Gamma_y = \Gamma_{f(x_i)} \cup \{\alpha_1\}$, $une_y = une_{f(x_i)} \setminus \{(n_1, \Diamond\alpha_1)\}$ and (n_1, n) in *min* of the branch. In the second outcome, we have a child z where $\Gamma_z = \Gamma_{f(x_i)}$, $une_z = une_{f(x_i)}$ and (n_1, n) in *min* of the branch. In the third outcome, we have a child v where $\Gamma_v = \Gamma_{f(x_i)}$, $une_v = une_{f(x_i)}$ and (n_1, n) in \prec of the branch.

On the other hand, since x is a predecessor of x_i in s , then we have $Inv(x, J(x))$. Furthermore, since we have $(n_1, \Diamond\alpha_1) \in une_{f(x_i)}$, it means that when the rule (\Diamond) is applied on the node $f(x)$, the branch where $(n_1, \Diamond\alpha_1) \in une_{f(x+1)}$ is the path that corresponds with the interpretation I . By (\Diamond) rule, since we have $Inv(x+1, J(x+1))$, $(n_1, \Diamond\alpha_1) \in une_{f(x+1)}$ and we know that $J(x+1) = J(x)$, then we have $(J(x), \Diamond\alpha_1) \in une_{x+1}$. Furthermore, since no rule application consumed $(n_1, \Diamond\alpha_1)$ up to $f(x_i)$, then the pair $(J(x), \Diamond\alpha_1)$ remains also in une_{x_i} . Also, we have $Inv(x_i, J(x_i))$, then there is $J' \geq J(x_i)$ where $J' \in min_{\prec}(J(x))$ and $I, J' \models \alpha_1$. Consider that $J(x_{i+1}) = J(x_i)$, we discuss all possibilities below:

- Case 1: If $J' = J(x_i)$, then we have $J(x_i) \in min_{\prec}(J(x))$ and $I, J(x_i) \models \alpha_1$. We define the next node x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_y$, $une_{x_{i+1}} = une_{x_i} \setminus \{(J(x), \Diamond\alpha_1)\}$ and add $(J(x), J(x_i))$ to min_s . We have $\Gamma_{x_{i+1}} = \Gamma_{x_i} \cup \{\alpha_1\}$ with $I, J(x_i) \models \alpha_1$. Additionally, we have $(J(x), J(x_i)) \in min_s$ with $J(x_i) \in min_{\prec}(J(x))$. The set \prec_s remains unchanged. Thus, the invariant $Inv(x_{i+1}, J(x_i))$ holds.
- Case 2: when $J' > J(x_i)$, we have two possibilities:
 - Case 2.1: If $J(x_i) \in min_{\prec}(J(x))$, then we define the next node x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_z$, $une_{x_{i+1}} = une_{x_i}$ and add $(J(x), J(x_i))$ to min_s . We have $(J(x), J(x_i)) \in min_s$ with $J(x_i) \in min_{\prec}(J(x))$. The sets $\Gamma_{x_{i+1}}$, $une_{x_{i+1}}$ and \prec_s remain unchanged. Thus, the invariant $Inv(x_{i+1}, J(x_i))$ holds.
 - Case 2.2: If $J(x_i) \notin min_{\prec}(n_1)$, then there exists $J'' \geq J(x)$ s.t. $(J'', J(x_i)) \in \prec$. We define the next node x_{i+1} with $\Gamma_{x_{i+1}} = \Gamma_v$, $une_{x_{i+1}} = une_{x_i}$ and add $(J(x), J(x_i))$ to \prec_s . We have $(J(x), J(x_i)) \in \prec_s$ with $J(x_i) \notin min_{\prec}(n_1)$. The sets $\Gamma_{x_{i+1}}$, $une_{x_{i+1}}$ and min_s remain unchanged. Thus, the invariant $Inv(x_{i+1}, J(x_i))$ holds.

Rule (Transition):

Suppose that the transition rule is applied on the state node $f(x_i)$. Let y be the child node of the node x_i in the branch. We have $\Gamma_y = \{\alpha_1 \mid \bigcirc\alpha_1 \in \Gamma_{f(x_i)}\}$ and $une_y = une_{f(x_i)}$. We define the next node x_{i+1} in s with $\Gamma_{x_{i+1}} = \Gamma_y$ and $une_{x_{i+1}} = une_{x_i}$. We consider that $J(x_{i+1}) = J(x_i) + 1$.

Since we have $Inv(x_i, J(x_i))$, then for all $\bigcirc\alpha_1 \in \Gamma_{x_i}$, we have $I, J(x_i) \models \bigcirc\alpha_1$ and therefore $I, J(x_i) + 1 \models \alpha_1$. The first condition of the invariant $Inv(x_{i+1}, J(x_i) + 1)$ is met.

Secondly, since x_i is a state node, then for each remaining $(n_1, \Diamond\alpha_1) \in une_{f(x_i)}$, either the rule (\Diamond) or (*une*) was applied to a node $f(x'_i)$ with the index $J(x'_i) = J(x_i)$ and $(n_1, \Diamond\alpha_1)$ was carried over to $f(x_i)$. In both rules, for each $(n_1, \Diamond\alpha_1) \in une_{f(x_i)}$, we have $(J(x_1), \Diamond\alpha_1) \in une_{x_i}$ s.t. $f(x_1)$ is the node where the rule (\Diamond) was applied to $\Diamond\alpha_1$ (see Case 2 for (\Diamond) and (*une*) rules). Furthermore, since we have

$Inv(x_i, J(x_i))$ and $f(x_i)$ is a state node, then for each $(J(x_1), \diamond\alpha_1) \in une_{x_i}$, there exists $J_2 > J(x_i)$ where $J_2 \in \min_{\prec}(J(x_1))$ and $I, J_2 \models \alpha_1$. Without loss of generality, there exists $J_2 \geq J(x_i) + 1$ where $J_2 \in \min_{\prec}(J(x_1))$ and $I, J_2 \models \alpha_1$. The second condition of the invariant $Inv(x_{i+1}, J(x_i) + 1)$ is met. Since \min_s and \prec_s remain unchanged, the invariant $Inv(x_{i+1}, J(x_i) + 1)$ holds.

Rule (\prec -inconsistency):

Suppose that the (\prec -inconsistency) arises on the node $f(x_i)$, and let n be the label of the $f(x_i)$ on the branch B . If this inconsistency rises, we have (n_1, n) in \min_B and (n_2, n) in \prec_B where $n_1 \leq n_2 \leq n$. These two pairs come from applying (\diamond) or (une) rule on two predecessors $f(x)$, $f(x')$ of $f(x_i)$ with the same label n and the same indexed time point $J(x) = J(x') = J(x_i)$.

Let J_1 be the time point corresponding to the node $f(x_1)$ with the label n_1 , and let J_2 be the time point corresponding to the node $f(x_2)$ with the label n_2 . It is worth to mention that $J_1 \leq J_2 \leq J(x_i)$. Since x, x' are predecessors of x , we have $Inv(x, J(x))$, $Inv(x', J(x'))$ and $Inv(x_i, J(x_i))$. Therefore, when the rules are applied on x and x' , we end up with $(J_1, J(x_i)) \in \min_s$ and $(J_2, J(x_i)) \in \prec_s$. Since $(J_1, J(x_i)) \in \min_s$, then we have $J(x_i) \in \min_{\prec}(J_1)$. On the other hand, since $(J_2, J(x_i)) \in \prec_s$, then there exists $J_3 \geq J_2$ s.t. $(J_3, J(x_i)) \in \prec$. Moreover, we have $J_1 \leq J_2$, this entails that there exists $J_3 \geq J_1$ s.t. $(J_3, J(x_i)) \in \prec$. This contradicts Definition 3.4 of minimality w.r.t. to the relation \prec . Therefore this cannot happen in an interpretation $I \in \mathcal{I}$.

Rule (Prune):

Let $f(x_i)$ be a state node where the prune condition is met. There is a sequence within s that goes the following way, $x_h = u, x_{h+1}, x_{h+2}, \dots, v = x_i$. The node u or x_h is the state node that comes before x_i and the node v is the current state node. Since v is a prune node, we have $\Gamma_v = \Gamma_u$ and $une_u = une_v$. We can see that if we apply the transition rule to the node x_i , we will have $\Gamma_{x_{i+1}} = \Gamma_{x_{h+1}}$ and $une_{x_{i+1}} = une_{x_{h+1}}$. Therefore, we can proceed with the construction of s as if x_i was linked to $f(u)$ instead of $f(v)$. Thanks to the transition, since we have $Inv(x_u, J(x_u))$, then we have $Inv(x_{i+1}, J(x_i) + 1)$.

Each time we find a pair (u, v) in the sequence s , we call it a *jump*. These jumps may occur once or many times (and it may go infinite) in s . In a sequence s , if a pair (u, v) jumps repeatedly in succession, we call the pair a *recurring jump*. It is worth to point out that, each time we jump backwards because of a node closed with prune, we return to the state labelled node that comes before. In general, the sequence s explores one branch B of \mathcal{T} , and it deviates sometimes to a prune node and goes back to B . Furthermore, since no eventuality is fulfilled within a prune loop, eventualities and their fulfilment are in the same branch B .

What we showed so far is that for an interpretation I and its corresponding sequence s , we have $Inv(x_i, J(x_i))$ for each $i \geq 0$. Going back to the start of the proof, we need to prove that the sequence finishes with a ticked node (such is the case when we end up in (Loop) or (Empty) node). We can see that if the sequence s is on a [prune] node, we jump back to the state node that comes before it. Theoretically, this jump can recur infinitely many times. This means that sequence goes infinite on this case (and never find a ticked node). We need to prove that this case cannot happen in the sequence s of I . Suppose that is the case, that means the last jump (u_k, v_k) in the sequence s is a recurring jump that goes infinitely many times. The jumps (u_j, v_j) that come before may recur many times but not infinitely many times (otherwise, (u_k, v_k) would not be the last jump). In the recurring jump (u_k, v_k) , no eventuality is fulfilled (whether it is classical or non-monotonic). This entails that when we are in a parent node $u_k < x_l < v_k$ that applies either (\diamond) or (une) rule, we move to the child node that delays the propagation of the eventuality (we are in Case 2 for both rules).

It is worth to point out that we have at least one eventuality in u_k . Let us take $\bigcirc\alpha_1 \in \Gamma_{u_k}$ for

example, since we have $Inv(u_k, J(u_k))$, that means that $I, J(u_k) \models \Diamond \alpha_1$. Thus, we take the *first* time point $J_{\alpha_1} > J(u_k)$ s.t. $I, J_{\alpha_1} \models \alpha_1$. We also have $I, J_{\alpha_1} \models \Diamond \alpha_1$. On the other hand, for all $J(u_k) < J < J_{\alpha_1}$, we have $I, J \models \Diamond \alpha_1$ $I, J \models \Diamond \alpha_1$. In other words, each time we encounter $\Diamond \alpha_1 \in \Gamma_{x_{l-1}}$ within our jumps (keep in mind we have $Inv(x_{l-1}, J)$), we pick the node in Case 2 of the (\Diamond) rule, i.e., $\Diamond \alpha_1 \in \Gamma_{x_l}$. However, in the node indexed with J_{α_1} , when we encounter $\Diamond \alpha_1 \in \Gamma_{x_{l'-1}}$ (keep in mind we have $Inv(x_{l'-1}, J_{\alpha_1})$), we pick the node in Case 1 of the (\Diamond) rule, i.e., $\alpha_1 \in \Gamma_{x'_l}$. This raises a contradiction, because the node $x_{l'}$ is not present within the jump (u_k, v_k) . Thus breaking the infinite recurring jump (u_k, v_k) .

If the eventuality is a non-monotonic one, namely $(J_1, \Diamond \alpha_1) \in une_{u_k}$. Since we have $Inv(u_k, J(u_k))$ with u_k being a state node, there exists $J' > J(u_k)$ s.t. $J' \in min_{\prec}(J_1)$ and $I, J' \models \alpha_1$. Let J_{α_1} be the first time point that met these criteria. For all $J(u_k) < J < J_{\alpha_1}$, each time we encounter $(J_1, \Diamond \alpha_1) \in une_{x_{l-1}}$ with the index J , we have $J_{\alpha_1} > J$, $J_{\alpha_1} \in min_{\prec}(J_1)$ and $I, J_{\alpha_1} \models \alpha_1$. Therefore, we pick Case 2 of (une) rule, i.e., $(J_1, \Diamond \alpha_1) \in une_{x_l}$. However, when we encounter $(J_1, \Diamond \alpha_1) \in une_{x_{l'-1}}$ with the index J_{α_1} , we have $J_{\alpha_1} \in min_{\prec}(J_1)$ and $I, J_{\alpha_1} \models \alpha_1$, then we pick the node in Case 1 of (une) rule, i.e., $\alpha_1 \in x_{l'}$. This raises a contradiction, because the node $x_{l'}$ is not present within the jump (u_k, v_k) .

We proved that since $I, 0 \models \alpha$, then the corresponding sequence s cannot finish on a contradiction, \prec -inconsistency or a prune jump. Therefore it must finish with a ticked node. Hence, the tableau \mathcal{T} of α has a ticked node and therefore a successful branch.

5.3 Illustrations of tableaux for \mathcal{L}_1 sentences

In this section, we highlight a tableau for three sentences. The first tableau in Figure 5.2 is for the sentence $\Box(\neg p \vee q) \wedge \Diamond p$. The tableau in Figure 5.3 is for the sentence $\Box p \wedge \Diamond \neg p$. Finally, the tableau in Figure 5.4 is for the sentence $\Diamond p \wedge \Diamond q$. In all of the figures, nodes have the form $n : (\Gamma, une)$. The state-labelled nodes are underlined. Each time there is an update in min_B or \prec_B in a branch, they are marked under the node. In Figure 5.2 and 5.4, the children nodes after the application of the (une) rule are stacked vertically instead of horizontally to fit it in the page (they are meant to be displayed the same way as in Figure 5.3). Otherwise, the tableaux go from the top to the bottom.

The first sentence $\Box(\neg p \vee q) \wedge \Diamond p$ is satisfiable. We show that its tableau contains a successful branch. The second sentence $\Box p \wedge \Diamond \neg p$. Therefore, all the branches of the fully expanded tableau of the second sentence are crossed. The third sentence $\Diamond p \wedge \Diamond q$ is a satisfiable sentence that has two defeasible eventualities. The presented tableaux for this sentence highlights how defeasible eventualities are handled.

The tableau for $\Box(\neg p \vee q) \wedge \Diamond p$ features a lot of interesting rule applications. Alongside the Boolean rules and (\Box) operator, Figure 5.2 shows the rules (\Diamond) and (une) rules applied on the sentence $\Diamond p$. It shows also branches closed with (Prune) and (Contradiction) rules. Notice in the pruned branches, there are no eventualities (there is only one in the example, which is $\Diamond p$) fulfilled between the state node with the label 0 and the state node with the label 1. Otherwise, the successful branch is ticked thanks to the (Loop) rule. Note that the state node of said branch contains no eventualities, i.e., the only sentences left are p, q and $\Box(\neg p \vee q)$.

The sentence $\Box p \wedge \Diamond \neg p$ in Figure 5.3 is unsatisfiable. This sentence shows a similar pattern as $\Box p \wedge \Diamond \neg p$. Each time the sentence $\Diamond p$ is fulfilled in a branch, it clashes with the sub-sentence in the scope of $\Box \neg p$. The fulfilment of the defeasible $\Diamond p$ (similar to $\Diamond p$) can be delayed in the future state. However, it will end up clashing with $\Box \neg p$. That is why the (Prune) rule is quite effective in stopping the tree from expanding indefinitely. While closing the branches that do not fulfil $\Diamond p$ in 1 (the left two branches), the only branch that fulfils the eventuality closes due to the (Contradiction) in the node

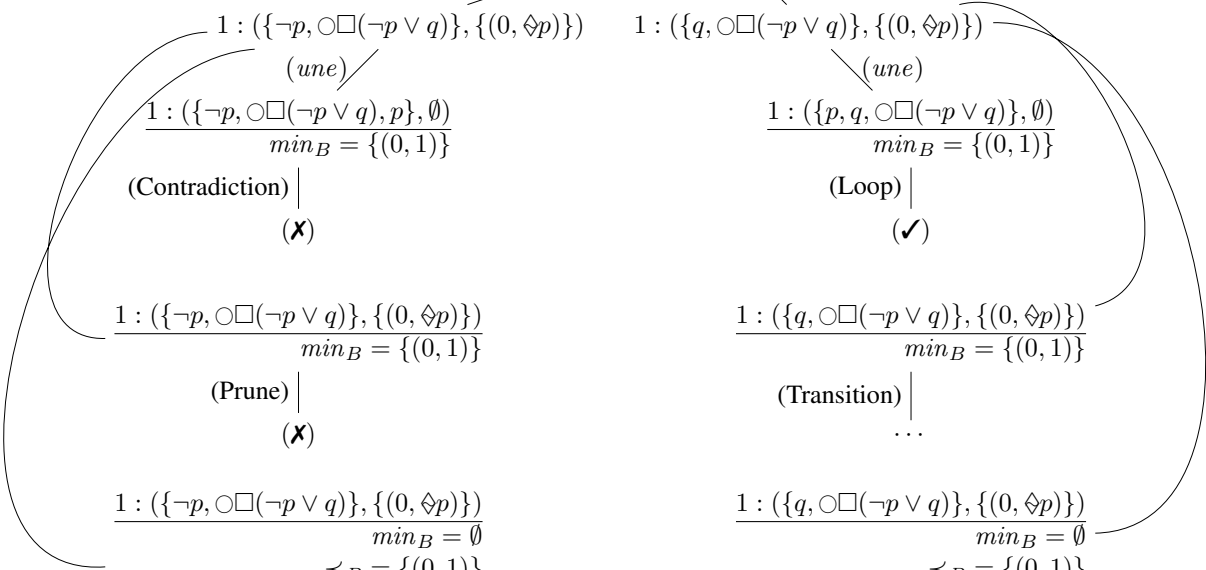


Figure 5.2: Tableau for $\Box(\neg p \vee q) \wedge \Diamond p$

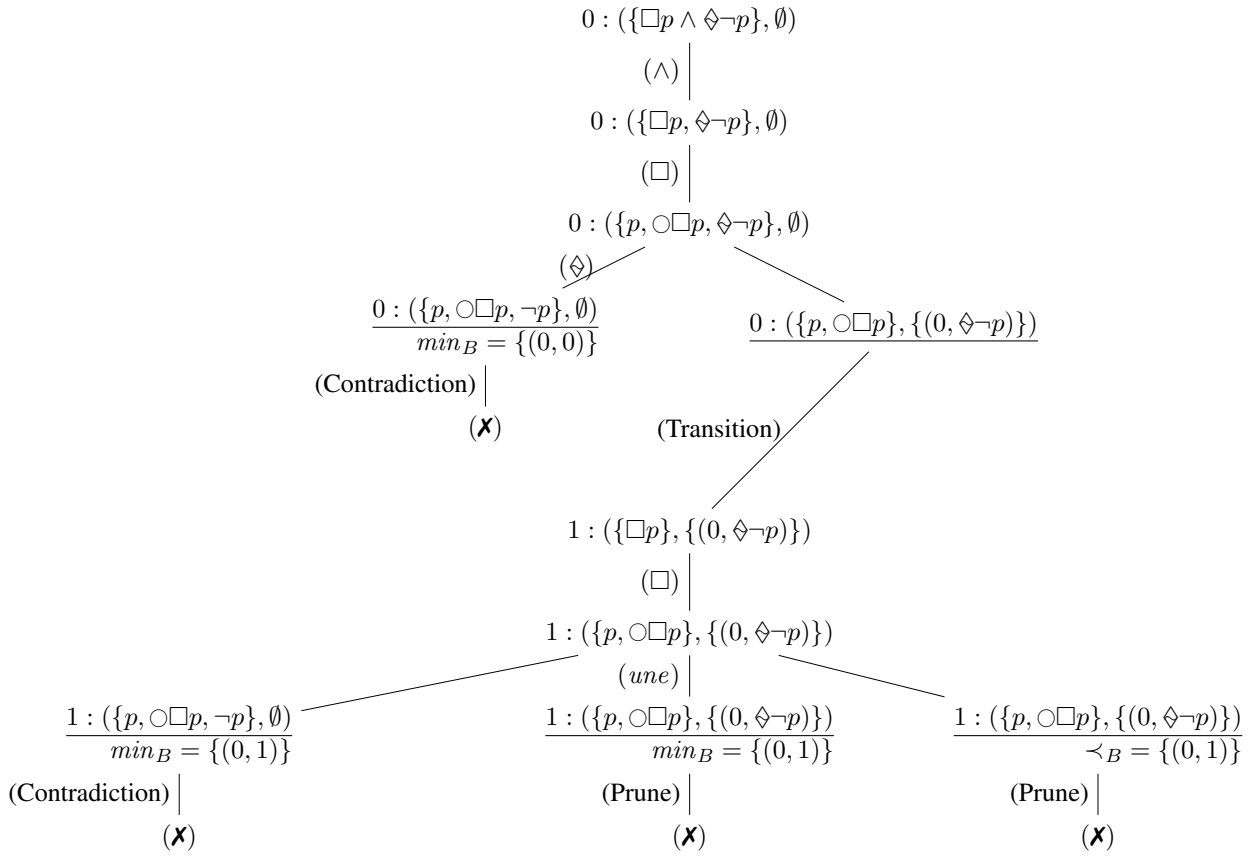


Figure 5.3: Tableau for $\Box p \wedge \Diamond \neg p$

1 : ($\{p, \circ\Box p, \neg p\}, \emptyset$).

The sentences in Figure 5.4 contains two defeasible eventualities $\Diamond p$ and $\Diamond q$. The branches on the right explore the outcome of $\Diamond p$ being fulfilled in 1 (meaning that 1 is a preferred future of 0). The outcome of $\Diamond q$ not being fulfilled in 1 because it is not a preferred future of 0 shall not be explored (the branch on the bottom right side). And vice-versa, branches on the right explore $\Diamond p$ not being fulfilled in 1 it is not a preferred future of 0. Therefore, the fulfilment of $\Diamond q$ in 1 shall not be explored (the first branch closed with (\prec -inconsistency) rule on the left side).

5.4 Summary

We showed how preferential semantics work in a one-pass tree-shaped tableau *à-la* Reynolds. We provide also semantic rules for the \Diamond operator. We showed how to handle non-monotonic eventualities using une , \prec_B and min_B . In the end, we proved that the method does not go indefinitely and that it is sound and complete. The loop/prune checkers proposed in this paper are specific to \mathcal{L}_1 since there are no infinitely replicating eventualities (i.e., $\Box\Diamond$ and $\Box\Diamond$), and work well under these restrictions.

The next step is to work on tableaux for far more expressive fragments of the \mathcal{L} language, specially when introducing the \Box to the syntax. An interesting starting point is the sub-fragment of \mathcal{L}^* that is recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha$$

We already established the bounded-model and decidability for state-dependent interpretations \mathcal{I}^{sd} in the case of \mathcal{L}^* sentences (see Theorem 4.30 for a reminder). Since the \mathcal{L}^* contains the aforementioned fragments, the upper-bound for the size of its interpretations is at most equal to the upper-bound for \mathcal{L}^* . The first goal is to define semantic rules for the \Box operator. We need to adapt the notations set for branches B in order to represent state-dependent interpretations. Finally, since we allow the replication of non-monotonic eventualities in the form of $\Box\Diamond$ in this fragment, we need to check if a three step prune similar to the one proposed by Reynolds [Rey16a] would work in this case.

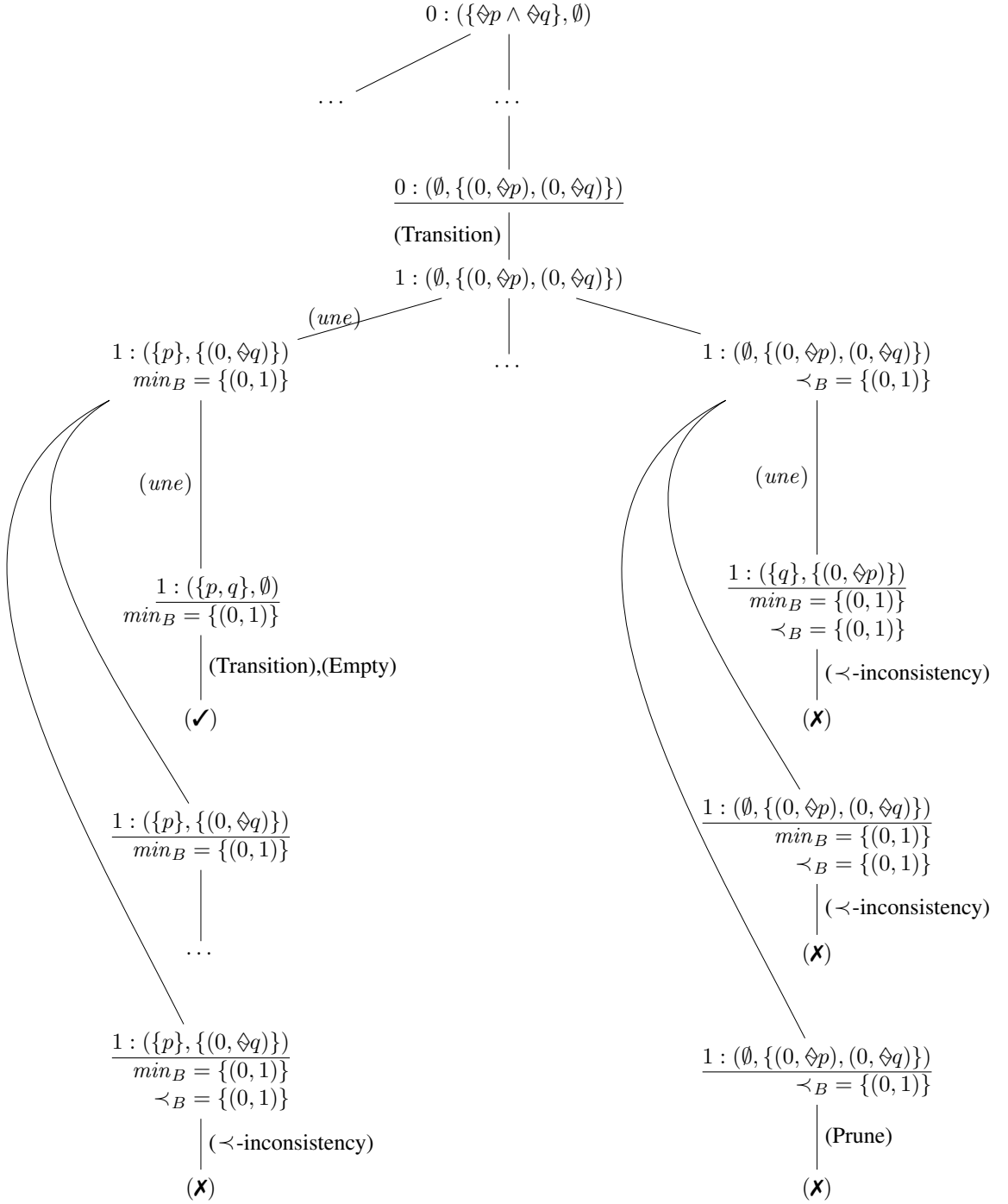


Figure 5.4: Tableau for $\Diamond p \wedge \Diamond q$

Part III

Conclusion

Conclusion and Perspectives

We presented in this memoir a new formalism, named defeasible linear temporal logic. In this logic, the syntax and the semantics of linear temporal logic are enriched by integrating the preferential semantics of the KLM approach. Using LTL^\sim , one can reason about systems with more tolerant behaviour when handling exceptions. Throughout executions of such systems, time points can range from non-pertinent, where exceptions can be ignored, to time points that are important, where it is imperative to check for exceptions to ensure their reliability. The addition of defeasible temporal operators \Box , \Diamond serves a similar purpose to the classical modalities \square , \diamond , all whilst expressing properties that target pertinent time points and tolerating their absence (properties) on others. The advantage of expressing defeasible specifications is to check the overall correctness of statements within the scope of non-monotonic temporal operators, without the need to express the intricacies and nuances within the interpretations. In this regard, the same purpose of the classical temporal modalities is preserved in \Box and \Diamond . When an agent is able to express specifications relating using classical LTL syntax, it is possible to obtain the defeasible version of the specification by simply switching temporal operators to their non-monotonic counterparts. *Defeasible always* \Box is quite interesting by itself. It acts as the operator \square when executions contain no exceptions through out their run. Moreover, it guarantees the overfall safety of properties when the absence of exceptions in the execution cannot be ensured. On the other hand, *preferable eventuality* \Diamond is a stronger version of the eventuality operator \diamond . The statements within the scope of \Diamond are not only true in a future time point, but a preferred one at that.

Three main axes were discussed in this memoir:

- In the first axis, we described non-monotonic operators and preferential temporal interpretations for LTL^\sim . We argued their importance for modelling and expressing specifications of programs that present some time points where exceptions are tolerated. We gave an example that describes a preferential temporal interpretation and highlights the truth values of non-monotonic temporal sentences.
- Second, we discussed the satisfiability problem in LTL^\sim . We showed the decidability of the problem in two fragments, namely \mathcal{L}^* and \mathcal{L}_1 . We defined structures to have compact representation of preferential interpretations. We also introduced an interesting class of interpretations, called state-dependent interpretations \mathcal{I}^{sd} . The bounded model property is shown for both of the fragments. Finally, we provide a procedure for checking whether a guessed interpretation is a model for sentences in the fragments. In the case of \mathcal{L}^* , although the procedure takes a polynomial time with respect to the size of the non-deterministically guessed model, the model can take an exponential space with respect to the input sentence. In addition, the truth values of sub-sentences of the input sentence are checked in each element of the structure. Thus, checking for \mathcal{I}^{sd} -satisfiability (satisfi-

ability in state-dependent interpretations) of the sentences in said model can be take an exponential time. Further analysis on the computational complexity is needed for \mathcal{L}^* fragments, as well as the study of other fragments of the language.

- Finally, we proposed a one-pass tree-shaped tableau for \mathcal{L}_1 sentences. This deviates from classical two-pass graph-shaped tableau such as Wolper's method [Wol85] and follows the newly proposed tableau by Reynolds [Rey16b]. We gave a first intuition on how preferential semantics could work in a tableau method for LTL^\sim , alongside the rules for \Diamond sentences. We showed that the method terminates always. We proved the soundness of the method and showed how an induced interpretation can be generated from a successful branch. We also showed that, thanks to the invariant function, we can always trace any interpretation that satisfies a sentence in the tableau down to a node with a checked mark (✓) (signalling that the tableau is successful). Thereby, the completeness of the method is also obtained.

Perspectives

We discuss in this part some lines of inquiry that can be explored in the future. In regards to LTL^\sim , we still yet to have a definitive class of complexity for the satisfiability problem. We shall work on an implementation of one-pass tree-shaped tableaux and find real case applications for this logic. Another objective is to introduce the non-monotonic inference \sim à la KLM to the formalism. Finally, the work done in LTL^\sim opens the door for integrating preferential semantics to other formalisms, we shall explain the intuition of what it might entails in the case of CTL and CTL^* .

A further complexity analysis is yet to be done in regards to the satisfiability problem in LTL^\sim . As discussed briefly on the summary, checking for \mathcal{I}^{sd} -satisfiability can take an exponential time. We need to find whether this problem is at least an exponential-time hard problem, or the answer to the problem can be optimized further. In addition, we need to investigate the satisfiability problem for extended fragments of LTL^\sim that contain defeasible until \tilde{U} and defeasible versions of past LTL operators.

The one-pass tree-shaped tableau presented in this memoir pertains to the fragment \mathcal{L}_1 which is restrictive in terms of expressivity. Nevertheless, it is a good starting point for exploring this approach for other fragments of the language. The end goal is to have a generic one-pass tableau that works for the entire language. Regarding implementation, the LTL satisfiability-checking solver named Leviathan was developed by Bertello et al. [BGMR19] that is based on Reynolds' tableaux. Leviathan's performance is comparable, both in time and memory, to other LTL satisfiability checkers in the literature such as NuSMV [CCG⁺02], Aalta [LYP⁺14], TRP++ [HK03] and PLTL graph-based tableaux [AGW09]. Our goal is to implement our one-pass tree-shaped tableaux to develop LTL^\sim satisfiability checkers (based on tools such as Leviathan).

We are also investigating defeasible inferences in a temporal formalism. With preferential temporal interpretations $I \in \mathcal{I}$ (which contain preference relation \prec), it is possible to integrate defeasible inference \sim to the LTL^\sim formalism. It can be defined in a language level, along the lines of $I, t \models \alpha \sim \beta$. By adapting the semantics of the KLM approach, the truth value of \sim sentences can be: $I, t \models \alpha \sim \beta$ if either $t \notin \min_{\prec}(\llbracket \alpha \rrbracket)$ or $t \in \llbracket \beta \rrbracket$ (with $\llbracket \alpha \rrbracket$ being the set of time points that satisfy α , same goes for $\llbracket \beta \rrbracket$). The inference can also be used at a meta-language level, such as $\alpha \sim_I \beta$. The formal definition would then be: $\alpha \sim_I \beta$ if $\min_{\prec}(\llbracket \alpha \rrbracket) \subseteq \llbracket \beta \rrbracket$. Each type of inference has its merit, and needs to be expanded further. One can see that both of the definitions are linked. For instance, if a sentence $\alpha \sim \beta$ is true for all $t \in \mathbb{N}$, then we can say that $\alpha \sim_I \beta$. The main reason behind adding the inference operator \sim is to have a defeasible version of specifications of the form $\alpha \rightarrow \beta$. Since the operator \rightarrow behaves monotonically, it means that whenever the premise α is broken (false at some points of time), there is no insurances on the consequence β (it can either be true or false). Thus, systems may behave arbitrarily in

this case. When presented with “tolerable” violations on premises, specifications of the form $\alpha \vdash \beta$ can reliably guarantee that the system can still properly function. Moreover, it ensures that the consequence β is produced on the most preferable time points where α is true. Currently, the non-monotonic relation \vdash in \mathcal{J} -interpretations satisfies the axioms of system **P** (Axioms 1.1 to 1.6). Their proof is analogous to the work of Kraus et al. [KLM90]. With a modular preference relation, the rational monotony (Axiom 1.7) can also be obtained. The remaining work is to prove the representation theorem for non-monotonic inferences in $LT\tilde{L}$ interpretations.

Another perspective is to propose a preferential extension for CTL and CTL^* . In branching-time logics, time is seen as a tree-structure where at the current point of time, each resulting branch is a possible execution. Alongside temporal operators \Box, \Diamond, \bigcirc and \mathcal{U} , these formalisms have quantifiers over branches (often referred to as paths in the literature). The quantifiers **A**, **E** can prefix linear temporal sentences to represent paths where said sentences hold. The quantifier **A** indicates for all paths, and **E** indicates that there exist a path. Same as temporal modalities, these quantifiers also have a monotonic behaviour, in the sense that, a sentence with the quantifier **A** is true only when the assertion within its scope holds for *all* of the paths. When presented with states where exception is tolerated through out the multiple paths, a defeasible version of these quantifiers can express the normality for paths of executions. As a first intuition, the quantifier $\tilde{\mathbf{A}}$ can encode “for all preferable paths” and the quantifier $\tilde{\mathbf{E}}$ can encode “there exists a preferable path”. These defeasible quantifiers can be used together with classical temporal modalities or their defeasible counterparts. Their syntax and meaning are yet to be explored. As for semantics, there are versions for both CTL and CTL^* that are defined using Kripke structure. Therefore, a preferential semantics à la KLM can also be discussed and investigated for these formalisms.

Appendices

Appendix A

The satisfiability problem of LTL^\sim sentences

A.1 Proofs of results in Section 4.2

Proposition 4.10. *Let $I = (V, \prec) \in \mathfrak{J}^{sd}$ and let $i \leq j \leq i' \leq j'$ be time points in $final(I)$ s.t. $V(j) = V(j')$. Then we have $j \in \min_{\prec}(i)$ iff $j' \in \min_{\prec}(i')$.*

Proof. Let $I = (V, \prec) \in \mathfrak{J}^{sd}$, and let us have four time points $i \leq j \leq i' \leq j' \in final(I)$.

- For the only-if part, we suppose that $j \in \min_{\prec}(i)$ and we prove that $j' \in \min_{\prec}(i')$. We have $i \leq i', i' \leq j', V(j) = V(j')$ and $j \in \min_{\prec}(i)$. Thanks to Proposition 4.3, $j' \in \min_{\prec}(i')$.
- For the if part, we suppose that $j' \in \min_{\prec}(i')$ and we prove that $j \in \min_{\prec}(i)$. We use a proof by contradiction. We assume that $j' \in \min_{\prec}(i')$ and we suppose that $j \notin \min_{\prec}(i)$. This implies that there exists $k \in [i, \infty[$ such that $(k, j) \in \prec$.
 - Case 1: $k \in [i', \infty[$. From Definition 4.1, since $V(j) = V(j')$ and $(k, j) \in \prec$, then $(k, j') \in \prec$ and therefore $j' \notin \min_{\prec}(i')$. This conflicts with our assumption that $j' \in \min_{\prec}(i')$.
 - Case 2: $k \in [i, i'[$. From Lemma 4.7, since $k \in final(I)$, there exists $k' \in [i', \infty[$ such that $V(k') = V(k)$. From Definition 4.1, since we have $V(j') = V(j)$, $V(k') = V(k)$ and $(k, j) \in \prec$, we also have $(k', j') \in \prec$, hence $j' \notin \min_{\prec}(i')$. This conflicts with the assumption that $j' \in \min_{\prec}(i')$.

□

Lemma 4.11. *Let $I = (V, \prec) \in \mathfrak{J}^{sd}$ and $i \leq i'$ be time points of $final(I)$ where $V(i) = V(i')$. Then for every $\alpha \in \mathcal{L}^*$, we have $I, i \models \alpha$ iff $I, i' \models \alpha$.*

Proof. Let $I = (V, \prec) \in \mathfrak{J}^{sd}$ and $i \leq i'$ in $final(I)$ be such that $V(i) = V(i')$. We prove that $I, i \models \alpha$ iff $I, i' \models \alpha$ using structural induction on α .

- Base: $\alpha = p$. We know that $I, i \models p$ iff $p \in V(i)$. Since $V(i) = V(i')$, we have $p \in V(i')$. Thus $I, i' \models p$.
- $\alpha = \neg\alpha_1$. For the only-if part, we assume that $I, i \models \neg\alpha_1$ and suppose that $I, i' \not\models \neg\alpha_1$. $I, i' \not\models \neg\alpha_1$ implies that $I, i' \models \alpha_1$. Since the Lemma holds on α_1 and $I, i' \models \alpha_1$, then $I, i \models \alpha_1$. This conflicts with the assumption $I, i \models \neg\alpha_1$. We follow the same reasoning for the if part.

- $\alpha = \alpha_1 \wedge \alpha_2$. $I, i \models \alpha_1 \wedge \alpha_2$ means that $I, i \models \alpha_1$ and $I, i \models \alpha_2$. Since the Lemma holds on both α_1 and α_2 , we have $I, i' \models \alpha_1$ and $I, i' \models \alpha_2$. Thus $I, i' \models \alpha_1 \wedge \alpha_2$.
- $\alpha = \Diamond \alpha_1$. For the only-if part, we assume that $I, i \models \Diamond \alpha_1$. This means that there exists $j \in [i, \infty[$ s.t. $I, j \models \alpha_1$. Thanks to Lemma 4.7, since $j \in final(I)$, then there exists $j' \in [i', \infty[$ where $V(j') = V(j)$. Thanks to the induction hypothesis, if $V(j) = V(j')$ and $I, j \models \alpha_1$, then $I, j' \models \alpha_1$. We conclude that $I, i' \models \Diamond \alpha_1$.

For the if part, we assume that $I, i' \models \Diamond \alpha_1$. This means that there is a $j' \in [i', \infty[$ s.t. $I, j' \models \alpha_1$. We know that $[i', \infty[\subseteq [i, \infty[$, and therefore we conclude that $I, i \models \Diamond \alpha_1$.

- $\alpha = \Diamond \alpha_1$. For the only-if part, we assume that $I, i \models \Diamond \alpha_1$. This means that there is a $j \in [i, \infty[$ s.t. $j \in \min_{\prec}(i)$ and $I, j \models \alpha_1$. Thanks to Lemma 4.7, since $j \in final(I)$, there exists $j' \in [i', \infty[$ such that $V(j') = V(j)$. Thanks to the induction hypothesis, if $V(j) = V(j')$ and $I, j \models \alpha_1$, then (I) $I, j' \models \alpha_1$. Thanks to Proposition 4.3, since $V(j) = V(j')$, $i \leq i'$, $i' \leq j'$ and $j \in \min_{\prec}(i)$, then we have (II) $j' \in \min_{\prec}(i')$. From (I) and (II), we conclude that $I, i' \models \Diamond \alpha_1$.

For the if part, we assume that $I, i' \models \Diamond \alpha_1$. $I, i' \models \Diamond \alpha_1$ means that there is a $j' \in [i', \infty[$ such that $j' \in \min_{\prec}(i')$ and (I) $I, j' \models \alpha_1$. We need to prove that $j' \in \min_{\prec}(i)$. We suppose that $j' \notin \min_{\prec}(i)$. It means that there exists $k \in [i, \infty[$ such that $(k, j') \in \prec$. From Lemma 4.7, since $k \in final(I)$, there is $k' \in [i', \infty[$ such that $V(k) = V(k')$. By Definition 4.1, since $(k, j') \in \prec$ and $V(k') = V(k)$, we have $(k', j') \in \prec$ and therefore $j' \notin \min_{\prec}(i')$, conflicting with the assumption $j' \in \min_{\prec}(i')$. Thus, we have (II) $j' \in \min_{\prec}(i)$. From (I) and (II), we conclude that $I, i \models \Diamond \alpha$.

□

The proof of Lemma 4.13 can be found in Section A.2.

Proposition 4.16. *Let $I = (V, \prec) \in \mathfrak{I}$, N_1, N_2 be two acceptable sequences w.r.t. I . Then $N_1 \cup N_2$ is an acceptable sequence w.r.t. I s.t. $size(I, N_1 \cup N_2) \leq size(I, N_1) + size(I, N_2)$.*

Proof. Let $I = (V, \prec) \in \mathfrak{I}$, N_1, N_2 be two acceptable sequences w.r.t. I and let $I^{N_1} = (V^{N_1}, \prec^{N_1})$, $I^{N_2} = (V^{N_2}, \prec^{N_2})$ be two pseudo-interpretations over N_1 and N_2 respectively. We assume that $N = N_1 \cup N_2$.

We suppose that N is not an acceptable sequence w.r.t. I . It means that there are two time points $t, t' \in final(I)$ s.t. $V(t) = V(t')$ where $t \in N$ and $t' \notin N$. Since $t \in N$, t is either an element of N_1 or N_2 . We assume that $t \in N_1$. By Definition 4.14, since $t \in N_1$ and N_1 is an acceptable sequence w.r.t. I , all time points of $final(I)$ that have the same valuation as t are in N_1 . Since $t' \in final(I)$ and $V(t') = V(t)$, then $t' \in N_1$, and therefore $t' \in N$. This conflicts with the supposition of $t' \notin N$. Same reasoning applies if we take $t \in N_2$. We conclude that for all $t \in N$ s.t. $t \in final(I)$, all $t' \in final(I)$ s.t. $V(t') = V(t)$ are also in N . Thus, N is an acceptable sequence w.r.t. I .

In order to prove that $size(I, N) \leq size(I, N_1) + size(I, N_2)$, we need to prove that $init(I, N) \subseteq init(I, N_1) \cup init(I, N_2)$ and $range(I, N) \subseteq range(I, N_1) \cup range(I, N_2)$. Let $t \in N$ be a time point s.t. $t \in init(I, N)$. By the definition of $init(I, N)$, we know that $t \in init(I)$. Since N is a sequence containing only elements of N_1 or N_2 , the time point t is either in N_1 or N_2 . By definition of $init(I, N_1)$, if $t \in N_1$ and $t \in init(I)$, then $t \in init(I, N_1)$. The same goes in the case of $t \in N_2$. We conclude that if $t \in init(I, N)$, then $t \in init(I, N_1) \cup init(I, N_2)$.

Following the same line of thought, we can prove that $final(I, N) \subseteq final(I, N_1) \cup final(I, N_2)$ and consequently we can prove that $range(I, N) \subseteq range(I, N_1) \cup range(I, N_2)$.

Since $\text{init}(I^N) \subseteq \text{init}(I^{N_1}) \cup \text{init}(I^{N_2})$, we have $\text{length}(\text{init}(I^N)) \leq \text{length}(\text{init}(I^{N_1})) + \text{length}(\text{init}(I^{N_2}))$. Similarly, if $\text{range}(I^N) \subseteq \text{range}(I^{N_1}) \cup \text{range}(I^{N_2})$, then $\text{card}(\text{range}(I^N)) \leq \text{card}(\text{range}(I^{N_1})) + \text{card}(\text{range}(I^{N_2}))$. We conclude that $\text{size}(I^N) \leq \text{size}(I^{N_1}) + \text{size}(I^{N_2})$. \square

Proposition 4.17. *Let $I = (V, \prec) \in \mathfrak{I}$ and N be an acceptable sequence w.r.t. I . If for all distinct $t, t' \in N$, we have $V(t') = V(t)$ only when both $t, t' \in \text{final}(I, N)$, then $\text{size}(I, N) \leq 2^{|\mathcal{P}|}$.*

Proof. Let $I = (V, \prec) \in \mathfrak{I}$ and N be an acceptable sequence w.r.t. I . We assume that for all $t, t' \in N$ s.t. we have $V(t') = V(t)$ only when both $t, t' \in \text{final}(I, N)$. Two cases are possible:

- $\text{init}(I, N)$ is empty. Since $\text{card}(\text{range}(I, N)) \leq 2^{|\mathcal{P}|}$, we have $\text{size}(I, N) \leq 2^{|\mathcal{P}|}$.
- $\text{init}(I, N)$ is not empty. We can see that for all $t \in \text{init}(I, N)$ and $t' \in N$ s.t. $t' \neq t$ we have $V(t') \neq V(t)$. If $\text{init}(I, N)$ has n time points having distinct valuations, then $\text{range}(\text{final}(I, N))$ has at most $2^{|\mathcal{P}|} - n$ valuations. Therefore, we have $\text{size}(I, N) \leq 2^{|\mathcal{P}|}$. \square

A.2 Proofs of results for Lemma 4.13

NB: The results marked (*) are introduced here, while they are omitted in the main text.

Proposition A.1 (*). *Let $I = (V, \prec) \in \mathfrak{I}$ and $i \in \text{final}(I)$. For all $j \in \text{final}(I)$, there exists $j' \geq j$ such that $V(j') = V(i)$.*

Proof. Let $I = (V, \prec) \in \mathfrak{I}$ and $i, j \in \text{final}(I)$. Let E be the set defined by $E = \{i' \in \text{final}(I) : V(i') = V(i)\}$. Since $i \in \text{final}(I)$, we have $E \neq \emptyset$. Suppose now that there does not exist $j' \geq j$ such that $V(j') = V(i)$. We have E is a non empty finite set of integers included in $[0, \dots, j-1]$. Let $k = \max\{k' \in E\}$. From the definitions of E and k , we have $k \in \text{final}(I)$ and there does not exist $k' > k$ such that $V(k') = V(k)$. This contradicts Lemma 4.7. We conclude that there exists $j' \geq j$ such that $V(j') = V(i)$. \square

Proposition A.2 (*). *Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over the same set of atomic propositions \mathcal{P} s.t. $\text{range}(I) = \text{range}(I')$. For all $i \in \text{final}(I)$ and $i' \in \text{final}(I')$ such that $V(i) = V'(i')$, we have :*

- (1) *for all $j \in [i, \infty[$ there exists $j' \in [i', \infty[$ such that $V(j') = V(j)$.*
- (2) *for all $j \in \min_{\prec}(i)$ there exists $j' \in \min_{\prec'}(i')$ such that $V(j) = V'(j')$.*

Proof. Let $I = (V, \prec) \in \mathfrak{I}^{sd}$, $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over \mathcal{P} s.t. $\text{range}(I) = \text{range}(I')$ and let $i, i' \in \text{final}(I)$ be such that $V(i) = V'(i')$.

- (1) Let $j \in [i, \infty[$. Since $i \in \text{final}(I)$, we have $j \in \text{final}(I)$. Moreover, given that $\text{range}(I) = \text{range}(I')$, we can assert that there exists $k \in \text{final}(I')$ such that $V'(k) = V(j)$. Hence, from Proposition A.1, there exists $j' \geq i'$ such that $V'(j') = V'(k) = V(j)$.
- (2) Let $j \in \min_{\prec}(i)$. We have $j \in \text{final}(I)$. From Property (1) above, there exists $j' \geq i'$ such that $V(j') = V(j)$. Suppose that $j' \notin \min_{\prec'}(i')$. Since $j' \geq i'$, there exists $k' \geq i'$ such that $(k', j') \in \prec'$. From Property (1) above, there exists $k \geq i$ such that $V(k) = V'(k')$. Since $V(k) = V'(k')$, $V(j') = V(j)$, $(k', j') \in \prec'$ and, since I and I' are two faithful interpretations, we can assert that $(k, j) \in \prec$. Consequently, since $k \geq i$ and $(k, j) \prec$, we have $j \notin \min_{\prec}(i)$, which leads to a contradiction. We conclude that $j' \in \min_{\prec'}(i')$.

□

Proposition A.3 (*). Let $\alpha \in \mathcal{L}^*$, $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over the same set of atomic propositions \mathcal{P} s.t. $\text{range}(I) = \text{range}(I')$. For every $\alpha \in \mathcal{L}^*$ and every $i \in \text{final}(I)$ and $i' \in \text{final}(I')$ s.t. $V(i) = V'(i')$, we have:

$$I, i \models \alpha \text{ iff } I', i' \models \alpha.$$

Proof. Let $I = (V, \prec)$, $I' = (V', \prec')$ be two faithful interpretations belonging to \mathfrak{I}^{sd} over the same set of atomic propositions \mathcal{P} s.t. $\text{range}(I) = \text{range}(I')$. Let $\alpha \in \mathcal{L}^*$, and let $i \in \text{final}(I)$ and $i' \in \text{final}(I')$ be such that $V(i) = V'(i')$. Without loss of generality, we suppose that α does not contain \vee , \square and \boxminus . This proposition can be proven by induction on the structure of the sentence α .

- Base case : $\alpha = p$, with $p \in \mathcal{P}$. Since $V(i) = V'(i')$, we have $p \in V(i)$ iff $p \in V'(i')$, and thus $I, i \models p$ iff $I', i' \models p$.
- $\alpha = \neg\alpha_1$. By the induction hypothesis, $I, i \models \alpha_1$ iff $I', i' \models \alpha_1$. Hence, we have $I, i \not\models \alpha_1$ iff $I', i' \not\models \alpha_1$. It follows that $I, i \models \neg\alpha_1$ iff $I', i' \models \neg\alpha_1$.
- $\alpha = \alpha_1 \wedge \alpha_2$. We know that $I, i \models \alpha_1 \wedge \alpha_2$ holds iff $I, i \models \alpha_1$ and $I, i \models \alpha_2$. By the induction hypothesis, since $V(i) = V'(i')$, we have $I, i \models \alpha_1$ and $I, i \models \alpha_2$ iff $I', i' \models \alpha_1$ and $I', i' \models \alpha_2$. We conclude that $I, i \models \alpha_1 \wedge \alpha_2$ iff $I', i' \models \alpha_1 \wedge \alpha_2$.
- $\alpha = \Diamond\alpha_1$. First we prove that $I, i \models \Diamond\alpha_1$ implies that $I', i' \models \Diamond\alpha_1$. We assume that $I, i \models \Diamond\alpha_1$. It means that there exists $j \in [i, \infty[$ s.t. $I, j \models \alpha_1$. From Proposition A.2 (1), there exists $j' \in [i', \infty[$ such that $V'(j') = V(j)$. By the induction hypothesis, we have $I', j' \models \alpha_1$. Hence, we conclude that $I', i' \models \Diamond\alpha_1$. The if part can be proved with a similar reasoning.
- $\alpha = \Diamond\alpha_1$. First we prove that $I, i \models \Diamond\alpha_1$ implies that $I', i' \models \Diamond\alpha_1$. We assume that $I, i \models \Diamond\alpha_1$. Hence, there exists $j \in [i, \infty[$ s.t. $j \in \min_{\prec}(i)$ and $I, j \models \alpha_1$. From Proposition A.2 (2), there exists $j' \in \min_{\prec'}(i')$ such that $V'(j') = V(j)$. By the induction hypothesis, we have $I', j' \models \alpha_1$. We conclude that $I', i' \models \Diamond\alpha_1$. The if part can be proved with a similar reasoning.

□

Corollary A.4 (*). Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over the same set of atomic propositions \mathcal{P} s.t. $\text{range}(I) = \text{range}(I')$. For every $i \in \text{final}(I)$ and every $\alpha \in \mathcal{L}^*$, we have: if $I, i \models \alpha$, then there exists $i' \in \text{final}(I')$ s.t. $I', i' \models \alpha$.

Proposition A.5 (*). Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over \mathcal{P} s.t. $\text{init}(I) \doteq \text{init}(I')$ and $\text{range}(I) = \text{range}(I')$. Then we have:

$$\text{For all } t, t' \in \text{init}(I), t' \in \min_{\prec}(t) \text{ iff } t' \in \min_{\prec'}(t).$$

Proof. Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over \mathcal{P} such that $\text{init}(I) \doteq \text{init}(I')$ and $\text{range}(I) = \text{range}(I')$. Let $t, t' \in \text{init}(I)$ be such that $t' \in \min_{\prec}(t)$. Suppose that $t' \notin \min_{\prec'}(t)$. Since $t' \geq t$, there exists $t'' \geq t$ such that $(t'', t') \in \prec'$. There are two possible cases:

- $t'' \in \text{init}(I')$. Since $\text{init}(I) \doteq \text{init}(I')$, we have $V'(t'') = V(t'')$. Moreover, since I and I' are two faithful interpretations and $V'(t') = V(t')$, we have $(t'', t') \in \prec$. Since $t'' \geq t$, it follows that $t' \notin \min_{\prec}(t)$. This leads to a contradiction. We conclude that $t' \in \min_{\prec'}(t)$.

- $t'' \in \text{final}(I')$. Since $\text{range}(I) = \text{range}(I')$, there exists $t''' \in \text{final}(I)$ such that $V'(t'') = V(t''')$. Moreover, since I and I' are two faithful interpretations and $V'(t') = V(t')$, we have $(t''', t') \in \prec$. Since $t''' \geq t$, it follows that $t' \notin \min_{\prec}(t)$. This leads to a contradiction. We conclude that $t' \in \min_{\prec'}(t)$.

Same reasoning can be applied to prove the if part. \square

Proposition A.6 (*). Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over \mathcal{P} such that $\text{init}(I) \doteq \text{init}(I')$ and $\text{range}(I) = \text{range}(I')$. For all $t \in \text{init}(I)$ and $t' \in \text{final}(I)$ such that $t' \in \min_{\prec}(t)$, we have $\{t'' \in \text{final}(I') \mid V'(t'') = V(t')\} \subseteq \min_{\prec'}(t)$.

Proof. Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over \mathcal{P} such that $\text{init}(I) \doteq \text{init}(I')$ and $\text{range}(I) = \text{range}(I')$. Let $t \in \text{init}(I)$, $t' \in \text{final}(I)$, $t'' \in \text{final}(I')$ be such that $t' \in \min_{\prec}(t)$ and $V'(t'') = V(t')$. We show that $t'' \in \min_{\prec'}(t)$.

Suppose that $t'' \notin \min_{\prec'}(t)$. Since $t'' \geq t$, there exists $t''' \geq t$ such that $(t''', t'') \in \prec'$. There are two possible cases.

- $t''' \in \text{init}(I')$. Since $\text{init}(I) \doteq \text{init}(I')$, we have $V'(t''') = V(t''')$. Moreover, since I and I' are two faithful interpretations and $V'(t'') = V(t'')$, we have $(t''', t'') \in \prec$. Since $t''' \geq t$, it follows that $t' \notin \min_{\prec}(t)$. This leads to a contradiction. We conclude that $t'' \in \min_{\prec'}(t)$.
- $t''' \in \text{final}(I')$. Since $\text{range}(I) = \text{range}(I')$, there exists $u \in \text{final}(I)$ such that $V'(t''') = V(u)$. Moreover, since I and I' are two faithful interpretations and $V'(t'') = V(t'')$, we have $(u, t'') \in \prec$. Since $u \geq t$, it follows that $t' \notin \min_{\prec}(t)$. There is a contradiction. We conclude that $t'' \in \min_{\prec'}(t)$.

\square

Lemma A.7 (*). Let $I = (V, \prec) \in \mathfrak{I}^{sd}$ and $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over \mathcal{P} such that $V'(0) = V(0)$, $\text{init}(I) \doteq \text{init}(I')$, and $\text{range}(I) = \text{range}(I')$. Then for all $\alpha \in \mathcal{L}^*$, we have :

$$\text{For all } t \in \text{init}(I) \cup \{0\}, I, t \models \alpha \text{ iff } I', t \models \alpha.$$

The singleton $\{0\}$ is there in case of an empty $\text{init}(I)$.

Proof. Let $I = (V, \prec) \in \mathfrak{I}^{sd}$, $I' = (V', \prec') \in \mathfrak{I}^{sd}$ be two faithful interpretations over \mathcal{P} such that $V'(0) = V(0)$, $\text{init}(I) \doteq \text{init}(I')$, and $\text{range}(I) = \text{range}(I')$. Let $\alpha \in \mathcal{L}^*$ and $t \in \text{init}(I) \cup \{0\}$. Without loss of generality, we suppose that α does not contain \vee , \square and \boxminus .

First, notice that in the case where $\text{init}(I)$ and $\text{init}(I')$ are empty intervals, we necessarily have $t = 0$. Moreover, since $t \in \text{final}(I)$ and $t \in \text{final}(I')$ and $V(0) = V'(0)$, from Proposition A.3, we can assert that $I, t \models \alpha$ iff $I', t \models \alpha$. Consequently, the property to be proved is true. Now, we suppose that $\text{init}(I)$ and $\text{init}(I')$ are non empty intervals. Hence, we have $t \in \text{init}(I)$ and $t \in \text{init}(I')$. We prove that $I, t \models \alpha$ iff $I', t \models \alpha$ by structural induction on α .

- Base case : $\alpha = p$. Since $t \in \text{init}(I)$, we have $V(t) = V'(t)$. Hence, $p \in V(t)$ iff $p \in V'(t)$. Thus $I, t \models p$ iff $I', t \models p$.
- $\alpha = \neg\alpha_1$. By the induction hypothesis, we have $I, t \models \alpha_1$ iff $I', t \models \alpha_1$. Hence, it is not the case that $I, t \models \alpha_1$ iff it is not the case that $I', t \models \alpha_1$. We conclude that, $I, t \models \neg\alpha_1$ iff $I', t \models \neg\alpha_1$.
- $\alpha = \alpha_1 \wedge \alpha_2$. We have $I, t \models \alpha_1 \wedge \alpha_2$ iff $I, t \models \alpha_1$ and $I, t \models \alpha_2$. Using the induction hypothesis, it follows that $I, t \models \alpha_1$ and $I, t \models \alpha_2$ iff $I', t \models \alpha_1$ and $I', t \models \alpha_2$. We conclude that $I, t \models \alpha_1 \wedge \alpha_2$ iff $I', t \models \alpha_1 \wedge \alpha_2$.

- $\alpha = \Diamond\alpha_1$. Suppose that $I, t \models \Diamond\alpha_1$. There exists a $t' \in [t, \infty[$ s.t. $I, t' \models \alpha_1$. Two cases are possible w.r.t. t' .
 - $t' \in \text{init}(I)$. By the induction hypothesis, we have $I', t' \models \alpha_1$. Hence, we conclude that $I', t \models \Diamond\alpha_1$.
 - $t' \in \text{final}(I)$. Since $\text{range}(I) = \text{range}(I')$, there exists $t'' \in \text{final}(I')$ such that $V'(t'') = V(t')$. From Proposition A.3, we have $I', t'' \models \alpha_1$. Since $t'' > t$, we have $I', t \models \Diamond\alpha_1$.

Same reasoning can be applied to prove the if part.

- $\alpha = \Diamond\alpha_1$. Suppose that $I, t \models \Diamond\alpha_1$. There exists $t' \in \min_{\prec}(t)$ s.t. $I, t' \models \alpha_1$. Two cases are possible w.r.t. t' .
 - $t' \in \text{init}(I)$. By the induction hypothesis, we have $I', t' \models \alpha_1$. Moreover, from Proposition A.5, we have $t' \in \min_{\prec'}(t)$. Hence, we conclude that $I', t \models \Diamond\alpha_1$.
 - $t' \in \text{final}(I)$. Since $\text{range}(I) = \text{range}(I')$, there exists $t'' \in \text{final}(I')$ such that $V'(t'') = V(t')$. From Proposition A.3, we have $I', t'' \models \alpha_1$. From Proposition A.6, we have $t'' \in \min_{\prec'}(t)$. Hence, we conclude that $I', t \models \Diamond\alpha_1$.

Same reasoning can be applied to prove the if part.

□

Lemma 4.13 is a direct result of result of Lemma A.7.

A.3 Proofs of results in Section 4.4

Lemma 4.27. *Let $\alpha \in \mathcal{L}_1$ be an \mathfrak{I} -satisfiable sentence and $I = (V, \prec) \in \mathfrak{I}$ be an interpretation such that $I, 0 \models \alpha$. Let I^N be the pseudo-interpretation of I over the finite sequence N such that $I^N, 0 \models_{\mathcal{P}} \alpha$, and $I' = (V', \prec')$ be the induced interpretation from I^N . Let $S = (n, V_S, \prec_S)$ be the finite preferential structure where $n = |N|$, $V_S(t) = V'(t)$ for each $t \in [0, |N| - 1]$ and $\prec_S = \prec'$. Let $\mathfrak{l}(S) = (V'', \prec'')$ be the induced interpretation from S . We have the following:*

- $\prec'' = \prec'$ and $V''(t) = V'(t)$ for each $t \in \mathbb{N}$;
- for every $\alpha_1 \in \text{Sf}(\alpha)$, we have $\alpha_1 \in \text{lab}_\alpha^S(t)$ iff $\mathfrak{l}(S), t \models \alpha_1$.

Proof. Let $\alpha \in \mathcal{L}_1$ be an \mathfrak{I} -satisfiable sentence and $I = (V, \prec) \in \mathfrak{I}$ be an interpretation such that $I, 0 \models \alpha$. Let $I^N = (V^N, \prec^N)$ be the pseudo-interpretation of I over the finite sequence N such that $I^N, 0 \models_{\mathcal{P}} \alpha$, and $I' = (V', \prec')$ be the induced interpretation from I^N . Let $S = (n, V_S, \prec_S)$ be the finite preferential structure where $n = |N|$, $V_S(t) = V'(t)$ for each $t \in [0, |N| - 1]$ and $\prec_S = \prec'$. Let $\mathfrak{l}(S) = (V'', \prec'')$ be the induced interpretation from S .

By Definition 4.24, we have $\prec'' = \prec_S$, $V''(t) = V_S(t)$ for each $t \in [0, n-1]$ and $V''(t) = V''(n-1)$ for each $t \in [n, \infty[$. Since $\prec' = \prec_S$, $V'(t) = V_S(t)$ for each, $t \in [0, |N| - 1]$ $V'(t) = V'(|N| - 1)$ for each $t \in [|N|, \infty[$ (see Definition 4.21) and $n = |N|$, then $\prec'' = \prec'$ and $V''(t) = V'(t)$ for each $t \in \mathbb{N}$. As such, the first item of the lemma holds. Therefore, the interpretations I' and $\mathfrak{l}(S)$ are interchangeable.

Going back to the pseudo-interpretation I^N , we know that N is a finite sequence. By the definition of truth values of sentences in pseudo-interpretations (after Definition 4.6), we have the following:

- for any $\bigcirc\alpha_1$ such that $I^N, t_i \models_{\mathcal{P}} \bigcirc\alpha_1$, we have $t_i + 1 \in N$ and $I^N, t_i + 1 \models_{\mathcal{P}} \alpha_1$;

- for any $\Diamond\alpha_1$ such that $I^N, t_i \models_{\mathcal{P}} \Diamond\alpha_1$, there exists $t_j \in N$ such that $t_j \geq t_i$ and $I^N, t_j \models_{\mathcal{P}} \alpha_1$;
- for any $\Diamond\alpha_1$ such that $I^N, t_i \models_{\mathcal{P}} \Diamond\alpha_1$, there exists $t_j \in N$ such that $t_j \in \min_{\prec_N}(t_i)$ and $I^N, t_j \models_{\mathcal{P}} \alpha_1$.

Since I' is the induced from I^N , the aforementioned properties hold for I' as well. Given $i, j \in [0, |N| - 1]$, we have the following:

- (I) for any $\bigcirc\alpha_1$ such that $I', i \models \bigcirc\alpha_1$, we have $i + 1 \in [0, |N| - 1]$ and $I', i + 1 \models \alpha_1$;
- (II) for any $\Diamond\alpha_1$ such that $I', i \models \Diamond\alpha_1$, there exists $j \in [0, |N| - 1]$ such that $j \geq i$ and $I', j \models \alpha_1$;
- (III) for any $\Diamond\alpha_1$ such that $I', i \models \Diamond\alpha_1$, there exists $j \in [0, |N| - 1]$ such that $j \in \min_{\prec'}(i)$ and $I', j \models \alpha_1$.

It is important to note that these properties hold for $I(S)$ as well. In other words, any sub-sentence of the form $\bigcirc\alpha_1$, $\Diamond\alpha_1$ or $\Diamond\alpha_1$ that holds at $t \in [0, n - 1]$ is satisfied within the finite part of $I(S)$.

Moving on to the second item of the lemma, let $\alpha_1 \in Sf(\alpha)$ and $t \in [0, n - 1]$. We use structural induction on sub-sentences of α_1 .

- $\alpha_1 = p$. By Definition 4.24, since $I(S)$ is the S -induced interpretation and $t \in [0, n - 1]$, then $V''(t) = V_S(t)$. Thus, we have $p \in V_S(t)$ iff $p \in V''(t)$. Therefore, we have $p \in lab_{\alpha}^S(t)$ iff $I(S), t \models p$.
- $\alpha_1 = \neg p$. Following the same reasoning as in the case of p , we have $V(t) = V_S(t)$. Thus, we have $p \notin V_S(t)$ iff $p \notin V''(t)$. Therefore, we have $\neg p \in lab_{\alpha}^S(t)$ iff $I(S), t \models \neg p$.
- $\alpha_1 = \alpha_2 \wedge \alpha_3$. Assume that $\alpha_2 \wedge \alpha_3 \in lab_{\alpha}^S(t)$, we have $\alpha_2, \alpha_3 \in lab_{\alpha}^S(t)$. By the induction hypothesis, $\alpha_2, \alpha_3 \in lab_{\alpha}^S(t)$ iff $I(S), t \models \alpha_2$ and $I(S), t \models \alpha_3$. Therefore, we have $\alpha_2 \wedge \alpha_3 \in lab_{\alpha}^S(t)$ iff $I(S), t \models \alpha_2 \wedge \alpha_3$.
- $\alpha_1 = \alpha_2 \vee \alpha_3$. Assume that $\alpha_2 \vee \alpha_3 \in lab_{\alpha}^S(t)$, we either have $\alpha_2 \in lab_{\alpha}^S(t)$ or $\alpha_3 \in lab_{\alpha}^S(t)$. Assume that $\alpha_2 \in lab_{\alpha}^S(t)$, by the induction hypothesis, $\alpha_2 \in lab_{\alpha}^S(t)$ iff $I(S), t \models \alpha_2$. Thus, we have $\alpha_2 \vee \alpha_3 \in lab_{\alpha}^S(t)$ iff $I(S), t \models \alpha_2 \vee \alpha_3$. Same reasoning can be applied in the case of $\alpha_3 \in lab_{\alpha}^S(t)$.
- $\alpha_1 = \Diamond\alpha_2$.
 - For the only-if part, we assume that $\Diamond\alpha_2 \in lab_{\alpha}^S(t)$, we have $\alpha_2 \in lab_{\alpha}^S(t')$ where $t' \in [t, n - 1]$. By the induction hypothesis, since $\alpha_2 \in lab_{\alpha}^S(t')$ and $t' \in [0, n - 1]$, then we have $\alpha_2 \in lab_{\alpha}^S(t')$ iff $I(S), t' \models \alpha_2$. Therefore, we have $I(S), t \models \Diamond\alpha_2$.
 - For the if part, we assume that $I(S), t \models \Diamond\alpha_2$. Knowing $I(S)$ is the same as I' , since $t \in [0, n - 1]$ and thanks to item (II), then there is $t' \in [t, n - 1]$ where $I(S), t' \models \alpha_2$. By the induction hypothesis, since $I(S), t' \models \alpha_2$ and $t' \in [t, n - 1]$, then we have $\alpha_2 \in lab_{\alpha}^S(t')$. Therefore, we have $\Diamond\alpha_2 \in lab_{\alpha}^S(t)$.
- $\alpha_1 = \bigcirc\alpha_2$. Assume that $\bigcirc\alpha_2 \in lab_{\alpha}^S(t)$, we have $\alpha_2 \in lab_{\alpha}^S(t + 1)$ where $t + 1 \leq n - 1$ (thanks to item (I), there is no need to check the case of $t = n - 1$). By the induction hypothesis, since $\alpha_2 \in lab_{\alpha}^S(t + 1)$ and $t + 1 \in [0, n - 1]$, we have $\alpha_2 \in lab_{\alpha}^S(t + 1)$ iff $I(S), t + 1 \models \alpha_2$. Therefore, we have $\bigcirc\alpha_2 \in lab_{\alpha}^S(t)$ iff $I(S), t \models \bigcirc\alpha_2$.

- $\alpha_1 = \Box\alpha_{bool}$. Assume that $\Box\alpha_{bool} \in lab_\alpha^S(t)$, we have $\alpha_{bool} \in lab_\alpha^S(t')$ for all $t' \in [t, n-1]$. By the induction hypothesis, since we have $\alpha_{bool} \in lab_\alpha^S(t')$ for each $t' \in [t, n-1]$, then for all $t' \in [t, n-1]$, we have (i) $\alpha_{bool} \in lab_\alpha^S(t')$ iff $I(S), t' \models \alpha_{bool}$. Moreover, we have $V''(t') = V_S(n-1) = V(n-1)$ for all $t' \in [n, \infty[$. Since α_{bool} is a Boolean sentence, $I(S), n-1 \models \alpha_{bool}$ and $V''(t') = V''(n-1)$ for all $t' \in [i, \infty[$; then we deduce that (ii) $I(S), n-1 \models \alpha_{bool}$ iff $I(S), t' \models \alpha_{bool}$ for all $t' \in [n, \infty[$. From (i) and (ii), we conclude that $\Box\alpha_{bool} \in lab_\alpha^S(t)$ iff $I(S), t \models \Box\alpha_{bool}$.
- $\alpha_1 = \Diamond\alpha_2$.
 - For the only-if part, we assume that $\Diamond\alpha_2 \in lab_\alpha^S(t)$. We have $\alpha_2 \in lab_\alpha^S(t')$ where $t' \in \min_{\prec_S}(t)$. By the induction hypothesis, since $\alpha_2 \in lab_\alpha^S(t')$ and $t' \in [0, n-1]$, then we have (i) $I(S), t' \models \alpha_2$. Moreover, since $t' \in \min_{\prec_S}(t)$, we have (ii) $t' \in \min_{\prec}(t)$. From (i) and (ii), we conclude that $I(S), t \models \Diamond\alpha_2$.
 - For the if part, we assume that $I(S), t \models \Diamond\alpha_2$. Knowing $I(S)$ is the same as I' , since $t \in [0, n-1]$ and thanks to item (III), then there is $t' \in [t, n-1]$ such that $t' \in \min_{\prec}(t)$ and $I(S), t' \models \alpha_2$. By the induction hypothesis, since $t' \in [t, n-1]$ and $I(S), t' \models \alpha_2$, then we have (i) $\alpha_2 \in lab_\alpha^S(t')$. Moreover, since $t' \in \min_{\prec}(t)$, we have (ii) $t' \in \min_{\prec_S}(t)$. From (I) and (II), we conclude that $\Diamond\alpha_2 \in lab_\alpha^S(t)$.

□

A.4 Proofs of results in Section 4.5

Lemma 4.39. *Let $\alpha_1 \in \mathcal{L}^*$ be a sentence, $I = (V, \prec) \in \mathcal{I}^{sd}$ and let T be a non-empty acceptable sequence w.r.t. I where for all $t_i \in T$ we have $I, t_i \models \Diamond\alpha_1$. Then for all $t, t' \in \text{Anchors}(I, T, \Diamond\alpha_1)$ s.t. $V(t) = V(t')$ and $t \neq t'$, we have $t, t' \in \text{final}(I, \text{Anchors}(I, T, \Diamond\alpha_1))$.*

Proof. Let $\alpha_1 \in \mathcal{L}^*$, let T be a non-empty acceptable sequence w.r.t. $I \in \mathcal{I}^{sd}$ where for all $t_i \in T$ we have $I, t_i \models \Diamond\alpha_1$. Just as a reminder, we have $\text{Anchors}(I, T, \Diamond\alpha_1) = \bigcup_{t_i \in T} ST(I, AS(I, \min_{\prec}(t_i)), \alpha_1)$. Thus, there exists $t_i, t'_i \in T$ such that $t \in ST(I, AS(I, \min_{\prec}(t_i)), \alpha_1)$ and $t' \in ST(I, AS(I, \min_{\prec}(t'_i)), \alpha_1)$. Suppose that the lemma is false. Then there are two time points $t, t' \in \text{Anchors}(I, T, \Diamond\alpha_1)$ with $t \neq t'$ such that t is in $\text{init}(I, \text{Anchors}(I, T, \Diamond\alpha_1))$ at least and $V(t) = V(t')$. Note that $t \in \text{init}(I)$, since we have $t \in \text{init}(I, \text{Anchors}(I, T, \Diamond\alpha_1))$. Without loss of generality, we assume that $t < t'$. From Definition 4.35, we have $t \in AS(I, (\mathbf{t}_{\alpha_1}^{I, AS(I, \min_{\prec}(t_i))}))$ where $\mathbf{t}_{\alpha_1}^{I, AS(I, \min_{\prec}(t_i))}$ is the chosen occurrence that satisfies α_1 in $AS(I, \min_{\prec}(t_i))$. Thanks to Definitions 4.31 and 4.33, since $t \in \text{init}(I)$, we can see that: $t = \mathbf{t}_{\alpha_1}^{I, AS(I, \min_{\prec}(t_i))}$. Moreover, (1) there is no $t'' \in \text{final}(I, AS(I, \min_{\prec}(t_i)))$ s.t. $I, t'' \models \alpha_1$ and (2) $t = \mathbf{t}_{\alpha_1}^{I, AS(I, \min_{\prec}(t_i))} = \max_{\prec}\{t'' \in \text{init}(I, AS(I, \min_{\prec}(t_i))) \mid I, t'' \models \alpha_1\}$ (see Definition 4.33). On the other hand, thanks to Proposition 4.3, since $t < t'$, $V(t) = V(t')$ and $t \in \min_{\prec}(t_i)$, we have $t' \in \min_{\prec}(t_i)$. Hence, we have $t' \in AS(I, \min_{\prec}(t_i))$. Since $t' \in ST(I, AS(I, \min_{\prec}(t'_i)), \alpha_1)$, we also have $I, t' \models \alpha_1$. From this and the property (1), we can assert that t' does not belong to $\text{final}(I, AS(I, \min_{\prec}(t_i)))$. It follows that $t' \in \text{init}(I, AS(I, \min_{\prec}(t_i)))$. From the property (2) we can assert that $t \geq t'$, which leads to a contradiction since $t < t'$. Therefore, for all $t, t' \in \text{Anchors}(I, T, \Diamond\alpha_1)$ s.t. $V(t) = V(t')$, we must have $t, t' \in \text{final}(\text{Anchors}(I, T, \Diamond\alpha_1))$. □

Proposition 4.40. *Let $\alpha \in \mathcal{L}^*$ be of the form $\mathcal{O}\alpha_1$, where $\mathcal{O} \in \{\Diamond, \Box, \Diamond\Box, \Box\Diamond\}$ and $\alpha_1 \in \mathcal{L}^*$. Let $I = (V, \prec) \in \mathcal{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I where for all $t \in T$ we have $I, t \models \alpha$. Then, we have:*

$$\text{size}(I, \text{Anchors}(I, T, \alpha)) \leq 2^{|\mathcal{P}|}.$$

Proof. Let $I = (V, \prec) \in \mathcal{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$. We show that is the case for the temporal operators $\Box, \Diamond, \Box, \Diamond$:

- Since $\text{size}(I, \text{Anchors}(I, T, \Box\alpha_1)) = \text{size}(I, \emptyset) = 0$, the result follows immediately.
- Since $\text{size}(I, \text{Anchors}(I, T, \Diamond\alpha_1)) = \text{size}(I, ST(I, \mathbb{N}, \alpha_1)) = 1$, the result follows immediately.
- T is an acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \Diamond\alpha_1$. From Proposition 4.39, for all $t'_i, t'_j \in \text{Anchors}(I, T, \Diamond\alpha_1)$ s.t. $V(t'_i) = V(t'_j)$ and $t'_i \neq t'_j$, we have $t'_i, t'_j \in \text{final}(I, \text{Anchors}(I, T, \Diamond\alpha_1))$. Thanks to Proposition 4.17, we conclude that $\text{size}(\text{Anchors}(I, T, \Diamond\alpha_1)) \leq 2^{|\mathcal{P}|}$.
- Going back to Definition 4.38, we have $\text{Anchors}(I, T, \Box\alpha_1) = DR(I, \bigcup_{t_i \in T} AS(I, \min_{\prec}(t_i)))$. We denote the acceptable sequence $\bigcup_{t_i \in T} AS(I, \min_{\prec}(t_i))$ by N . From Definition 4.37 we have $\text{Anchors}(I, T, \Box\alpha_1) = DR(I, N) = \bigcup_{v \in \text{val}(I, N)} ST(I, N, \alpha_v)$. Moreover, we know that $\text{size}(ST(I, N, \alpha_v)) = 1$ for all $v \in \text{val}(I, N)$. Consequently, thanks to Proposition 4.16, we have $\text{size}(\bigcup_{v \in \text{val}(I, N)} ST(I, N, \alpha_v)) \leq \text{card}(\text{val}(I, N))$. We can see that $\text{card}(\text{val}(I, N)) \leq 2^{|\mathcal{P}|}$, and therefore $\text{size}(\text{Anchors}(I, T, \Box\alpha_1)) = \text{size}(\bigcup_{v \in \text{val}(I, N)} ST(I, N, \alpha_v)) \leq 2^{|\mathcal{P}|}$.

□

Proposition 4.41. Let $\alpha_1 \in \mathcal{L}^*$, $I = (V, \prec) \in \mathcal{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \Box\alpha_1$, with $\alpha_1 \in \mathcal{L}^*$. For all acceptable sequences N w.r.t. I s.t. $\text{Anchors}(I, T, \Box\alpha_1) \subseteq N$ and for all $t_i \in N \cap T$, let $I^N = (V^N, \prec^N)$ be the pseudo-interpretation over N and $t' \in N$. We have the following:

$$\text{If } t' \notin \min_{\prec}(t_i), \text{ then } t' \notin \min_{\prec^N}(t_i).$$

Proof. Let $I = (V, \prec) \in \mathcal{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \Box\alpha_1$, with $\alpha_1 \in \mathcal{L}^*$. Let N be an acceptable sequence w.r.t. I s.t. $\text{Anchors}(I, T, \Box\alpha_1) \subseteq N$. Let $t_i \in N \cap T$. Let $t' \in N$ be a time point s.t. $t' \notin \min_{\prec}(t_i)$. There are two possible cases:

- $t' \notin [t_i, \infty[$: Since $t' \notin [t_i, \infty[$, then $t' \notin [t_i, \infty[\cap N$. Therefore, we conclude that $t' \notin \min_{\prec^N}(t_i)$.
- $t' \in [t_i, \infty[$: Since \prec satisfies the well-foundedness condition (that is why T must not be empty), $t' \notin \min_{\prec}(t_i)$ implies that there exists a time point $t'' \in \min_{\prec}(t_i)$ s.t. $(t'', t') \in \prec$. Let $\alpha_{t''}$ be the representative sentence of $V(t'')$ (recall that $\alpha_{t''} = \bigwedge \{p \mid p \in V(t'')\} \wedge \bigwedge \{\neg p \mid p \notin V(t'')\}$). For the sake of readability, we shall denote the sequence $\bigcup_{t \in T} AS(I, \min_{\prec}(t))$ with M . Notice that there exists $V \in \text{val}(I, M)$ such that $V = V(t'')$ since $t_i \in T$ and $t'' \in \min_{\prec}(t_i)$. Thanks to Definition 4.37, since $DR(I, M) = \bigcup_{v \in \text{val}(I, M)} ST(I, M, \alpha_v)$ and $V(t'') \in \text{val}(I, M)$, we can find $t''' \in ST(I, M, \alpha_{t''})$ where $t''' \in DR(I, M) \subseteq N$, $V(t''') = V$ and $t''' \geq t''$. Since $(t'', t') \in \prec$, $I \in \mathcal{I}^{sd}$ and $V(t''') = V(t'')$, we have $(t''', t') \in \prec$. Moreover, we have $t''', t' \in N$, and therefore $(t''', t') \in \prec^N$. Since $t''' \in [t_i, \infty[\cap N$ and $(t''', t') \in \prec^N$, we conclude that $t' \notin \min_{\prec^N}(t_i)$.

□

Proposition 4.43. Let $\alpha \in \mathcal{L}^*$ be in NNF, $I = (V, \prec) \in \mathcal{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$. Then, we have $\text{size}(I, \text{Keep}(I, T, \alpha)) \leq \mu(\alpha) \times 2^{|\mathcal{P}|}$.

Proof. Let $I = (V, \prec) \in \mathfrak{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$ where $\alpha \in \mathcal{L}^*$.

We use structural induction on T and α in order to prove this property.

- Base: $\alpha = p$ or $\alpha = \neg p$. We have $Keep(I, T, \alpha) = \emptyset$. Since $size(I, \emptyset) = 0 \leq \mu(\alpha) \times 2^{|\mathcal{P}|} = 0$, then the property holds on atomic propositions.
- $\alpha = \alpha_1 \wedge \alpha_2$. Since $I, t \models \alpha_1 \wedge \alpha_2$ for all $t \in T$, we can assert that $I, t \models \alpha_1$ and $I, t \models \alpha_2$. By applying the induction hypothesis on T, α_1 and α_2 , we have $size(I, Keep(I, T, \alpha_1)) \leq \mu(\alpha_1) \times 2^{|\mathcal{P}|}$ and $size(I, Keep(I, T, \alpha_2)) \leq \mu(\alpha_2) \times 2^{|\mathcal{P}|}$. Thanks to Proposition 4.16, $size(Keep(I, T, \alpha_1 \wedge \alpha_2)) \leq (\mu(\alpha_1) + \mu(\alpha_2)) \times 2^{|\mathcal{P}|}$. We conclude that $size(I, Keep(I, T, \alpha_1 \wedge \alpha_2)) \leq (\mu(\alpha_1 \wedge \alpha_2)) \times 2^{|\mathcal{P}|}$.
- $\alpha = \alpha_1 \vee \alpha_2$. Since $I, t \models \alpha_1 \vee \alpha_2$ for all $t \in T$, we have $I, t \models \alpha_1$ or $I, t \models \alpha_2$. Let T_1 be the sequence (resp. T_2) containing all $t_1 \in T$ (resp. $t_2 \in T$) s.t. $I, t_1 \models \alpha_1$ (resp. $I, t_2 \models \alpha_2$). Using induction hypothesis on T_1, T_2, α_1 and α_2 , we have $size(I, Keep(I, T_1, \alpha_1)) \leq \mu(\alpha_1) \times 2^{|\mathcal{P}|}$ and $size(I, Keep(I, T_2, \alpha_2)) \leq \mu(\alpha_2) \times 2^{|\mathcal{P}|}$. We conclude that $size(I, Keep(I, T, \alpha_1 \vee \alpha_2)) \leq (\mu(\alpha_1 \vee \alpha_2)) \times 2^{|\mathcal{P}|}$ in the same way as $\alpha_1 \wedge \alpha_2$.
- $\alpha = \Diamond \alpha_1$. First of all, we proved in Proposition 4.40 that (I) $size(I, Anchors(I, T, \Diamond \alpha_1)) \leq 2^{|\mathcal{P}|}$. On the other hand, thanks to Definition 4.38 it is easy to see that $size(I, Anchors(I, T, \Diamond \alpha_1))$ is a non-empty acceptable sequence w.r.t. I s.t. for all $t' \in Anchors(I, T, \Diamond \alpha_1)$ we have $I, t' \models \alpha_1$. By the induction hypothesis on $Anchors(I, T, \Diamond \alpha_1)$ and α_1 , we have (II) $size(I, Keep(I, Anchors(I, T, \Diamond \alpha_1), \alpha_1)) \leq \mu(\alpha_1) \times 2^{|\mathcal{P}|}$. Thanks to Proposition 4.16, from (I) and (II) we conclude that $size(I, Keep(I, T, \Diamond \alpha_1)) \leq (1 + \mu(\alpha_1)) \times 2^{|\mathcal{P}|} = \mu(\Diamond \alpha_1) \times 2^{|\mathcal{P}|}$.
- $\alpha = \Box \alpha_1$. As a result of semantics of the \Box operator, we can see that for all $t \in T$ we have $I, t \models \alpha_1$. By the induction hypothesis on T and α_1 , we have $size(I, Keep(I, T, \alpha_1)) \leq \mu(\alpha_1) \times 2^{|\mathcal{P}|}$. Since $Keep(I, T, \alpha_1) = Keep(I, T, \Box \alpha_1)$ then $size(I, Keep(I, T, \Box \alpha_1)) \leq \mu(\alpha_1) \times 2^{|\mathcal{P}|}$. We conclude that $size(I, Keep(I, T, \Box \alpha_1)) \leq \mu(\Box \alpha_1) \times 2^{|\mathcal{P}|}$.
- $\alpha = \Diamond \alpha_1$. First of all, we proved in Proposition 4.40 that (I) $size(I, Anchors(I, T, \Diamond \alpha_1)) \leq 2^{|\mathcal{P}|}$. On the other hand, thanks to Definition 4.38, it is easy to see that $Anchors(I, T, \Diamond \alpha_1)$ is a non-empty acceptable sequence w.r.t. I s.t. for all $t' \in Anchors(I, T, \Diamond \alpha_1)$ we have $I, t' \models \alpha_1$. By the induction hypothesis on $Anchors(I, T, \Diamond \alpha_1)$ and α_1 , we have (II) $size(I, Keep(I, Anchors(I, T, \Diamond \alpha_1), \alpha_1)) \leq \mu(\alpha_1) \times 2^{|\mathcal{P}|}$. Thanks to Proposition 4.16, from (I) and (II), we conclude that $size(I, Keep(I, T, \Diamond \alpha_1)) \leq (1 + \mu(\alpha_1)) \times 2^{|\mathcal{P}|} = \mu(\Diamond \alpha_1) \times 2^{|\mathcal{P}|}$.
- $\alpha = \Box \alpha_1$. First of all, we proved in Proposition 4.40 that (I) $size(I, Anchors(I, T, \Box \alpha_1)) \leq 2^{|\mathcal{P}|}$. On the other hand, from Definition 4.42, we have $T' = \bigcup_{t_i \in T} AS(I, \min_{\prec}(t_i))$. It is easy to see that for all $t' \in T'$ we have $I, t' \models \alpha_1$ and that T' is a non-empty acceptable sequence w.r.t. I . By the induction hypothesis on T' and α_1 , we have (II) $size(I, Keep(I, T', \alpha_1)) \leq \mu(\alpha_1) \times 2^{|\mathcal{P}|}$. Thanks to Proposition 4.16, from (I) and (II) we conclude that $size(I, Keep(I, T, \Box \alpha_1)) \leq (1 + \mu(\alpha_1)) \times 2^{|\mathcal{P}|} = \mu(\Box \alpha_1) \times 2^{|\mathcal{P}|}$.

□

Lemma 4.44. Let $\alpha \in \mathcal{L}^*$ be in NNF, $I = (V, \prec) \in \mathfrak{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$. For all acceptable sequences N w.r.t. I , if $Keep(I, T, \alpha) \subseteq N$, then for every $t \in N \cap T$, we have $I^N, t \models_p \alpha$.

Proof. Let $\alpha \in \mathcal{L}^*$ be in NNF, $I = (V, \prec) \in \mathfrak{I}^{sd}$, and let T be a non-empty acceptable sequence w.r.t. I s.t. for all $t \in T$ we have $I, t \models \alpha$. Let N be an acceptable sequence w.r.t. I s.t. $\text{Keep}(I, T, \alpha) \subseteq N$ and $t \in N \cap T$ (we assume that N contains at least one $t \in T$). Let $I^N = (N, V^N, \prec^N)$ be the pseudo-interpretation over N .

We use structural induction on T and α in order to prove this property.

- $\alpha = p$ or $\alpha = \neg p$. Since $I, t \models p$ (resp. $\neg p$), it means that $p \in V(t)$ (resp. $p \notin V(t)$). We know that $V^N(t) = V(t)$. We conclude that $I^N, t \models_{\mathcal{P}} p$ (resp. $\neg p$).
- $\alpha = \alpha_1 \wedge \alpha_2$. Since $I, t \models \alpha_1 \wedge \alpha_2$ for all $t \in T$, we can assert that $I, t \models \alpha_1$ and $I, t \models \alpha_2$. By applying the induction hypothesis on T , α_1 and α_2 , since $\text{Keep}(I, T, \alpha_1) \subseteq N$ and $\text{Keep}(I, T, \alpha_2) \subseteq N$, therefore we have $I^N, t \models_{\mathcal{P}} \alpha_1$ and $I^N, t \models_{\mathcal{P}} \alpha_2$. Thus, we have $I^N, t \models_{\mathcal{P}} \alpha_1 \wedge \alpha_2$.
- $\alpha = \alpha_1 \vee \alpha_2$. Suppose that $I, t \models \alpha_1$ (the case $I, t \models \alpha_2$ can be treated in a similar way) and let T_1 be the sequence containing all $t_1 \in T$ s.t. $I, t_1 \models \alpha_1$. Here, since $t \in T_1$, therefore T_1 is non-empty and $t \in T_1 \cap N$. We know that $\text{Keep}(I, T_1, \alpha_1) \cup \text{Keep}(I, T_2, \alpha_2) \subseteq N$. Consequently $\text{Keep}(I, T_1, \alpha_1) \subseteq N$. From the induction hypothesis, we have $I^N, t \models_{\mathcal{P}} \alpha_1$. Therefore, we have $I^N, t \models_{\mathcal{P}} \alpha_1 \vee \alpha_2$.
- $\alpha = \Diamond \alpha_1$. We have $I, t \models \Diamond \alpha_1$ and we need to prove that $I^N, t \models_{\mathcal{P}} \Diamond \alpha_1$. $I, t \models \Diamond \alpha_1$ means that there exists $t' \in [t, \infty[$ such that $I, t' \models \alpha_1$. Therefore $\text{Anchors}(I, T, \Diamond \alpha_1)$ is non-empty (see Definition 4.38). We know that $\text{Anchors}(I, T, \Diamond \alpha_1) \subseteq \text{Keep}(I, T, \Diamond \alpha_1) \subseteq N$, consequently $\text{Anchors}(I, T, \Diamond \alpha_1) \cap N$ is non-empty. Thanks to Definition 4.38 it is easy to see that for all $t_1 \in \text{Anchors}(I, T, \Diamond \alpha_1)$ we have $I, t_1 \models \alpha_1$. By the induction hypothesis on $\text{Anchors}(I, T, \Diamond \alpha_1)$ and α_1 , since $\text{Keep}(I, \text{Anchors}(I, T, \Diamond \alpha_1), \alpha_1) \subseteq N$, $t' \in \text{Anchors}(I, T, \Diamond \alpha_1)$ (a non-empty acceptable sequence w.r.t I) and $I, t' \models \alpha_1$, thus $I^N, t' \models_{\mathcal{P}} \alpha_1$. Therefore, we have $I^N, t \models_{\mathcal{P}} \Diamond \alpha_1$.
- $\alpha = \Box \alpha_1$. We have $I, t \models \Box \alpha_1$ and we need to prove that $I^N, t \models_{\mathcal{P}} \Box \alpha_1$. We know that for all $t' \geq t$ we have $I, t' \models \alpha_1$. We can assert that for all $t' \in N \cap T$ such that $t' \geq t$, we have $I^N, t' \models_{\mathcal{P}} \alpha_1$. By the induction hypothesis on T and α_1 , $\text{Keep}(I, T, \alpha_1) = \text{Keep}(I, T, \Box \alpha_1)$. Consequently $\text{Keep}(I, T, \alpha_1) \subseteq N$ since for all $t' \in N \cap T$, we have $I^N, t' \models_{\mathcal{P}} \alpha_1$. We conclude that $I^N, t \models_{\mathcal{P}} \Box \alpha_1$.
- $\alpha = \Diamond \alpha_1$. We have $I, t \models \Diamond \alpha_1$ and we need to prove that $I^N, t \models_{\mathcal{P}} \Diamond \alpha_1$. $I, t \models \Diamond \alpha_1$ means that there exists $t' \in \min_{\prec}(t)$ such that $I, t' \models \alpha_1$, and therefore $\text{Anchors}(I, T, \Diamond \alpha_1)$ is non-empty (see Definition 4.38). We know that $\text{Anchors}(I, T, \Diamond \alpha_1) \subseteq \text{Keep}(I, T, \Diamond \alpha_1) \subseteq N$, consequently $\text{Anchors}(I, T, \Diamond \alpha_1) \cap N$ is non-empty. Thanks to Definition 4.38 it is easy to see that for all $t_1 \in \text{Anchors}(I, T, \Diamond \alpha_1)$ we have $I, t_1 \models \alpha_1$. By the induction hypothesis on $\text{Anchors}(I, T, \Diamond \alpha_1)$ and α_1 , since $\text{Keep}(I, T_1, \alpha_1) \subseteq N$ with $T_1 = \text{Anchors}(I, T, \Diamond \alpha_1)$, and T_1 is an acceptable sequence where $I, t' \models \alpha_1$ for all $t' \in T_1$, we conclude (I) $I^N, t' \models_{\mathcal{P}} \alpha_1$. Thanks to the construction of the pseudo-interpretation I^N , since $t' \in \min_{\prec_N}(t)$, we have (II) $t' \in \min_{\prec}(t)$. From (I) and (II), we conclude that $I^N, t \models_{\mathcal{P}} \Diamond \alpha_1$.
- $\alpha = \Box \alpha_1$. We have $I, t \models \Box \alpha_1$ and we need to prove that $I^N, t \models_{\mathcal{P}} \Box \alpha_1$. $I, t \models \Box \alpha_1$ means that for all $t' \in \min_{\prec}(t)$ we have $I, t' \models \alpha_1$, therefore for all $t' \in T' = \bigcup_{t_i \in T} \text{AS}(I, \min_{\prec}(t_i))$ we have $I, t' \models \alpha_1$. In addition, thanks to the well-founded condition on \prec , T' is non-empty. We know that $\text{Anchors}(I, T, \Box \alpha_1) \subseteq \text{Keep}(I, T, \Box \alpha_1) \subseteq N$ and that $\text{Anchors}(I, T, \Box \alpha_1) = \text{DR}(I, T')$, consequently $T' \cap N$ is non-empty. We use proof by contradiction. Suppose that $I^N, t \not\models_{\mathcal{P}} \Box \alpha_1$, which means there exists $t' \in \min_{\prec_N}(t)$ s.t. $I^N, t' \not\models_{\mathcal{P}} \alpha_1$. Thanks to Proposition

4.41, if $t' \in \min_{\prec_N}(t_i)$, then $t' \in \min_{\prec}(t_i)$. Just a reminder, we have $T' = \bigcup_{t_i \in T} AS(I, \min_{\prec}(t_i))$ where for all $t'' \in T'$ we have $I, t'' \models \alpha_1$. Note that T' is a non-empty acceptable sequence w.r.t. I . By the induction hypothesis on T' and α_1 , since $Keep(I, T', \alpha_1) \subseteq N$, and $t' \in AS(I, \min_{\prec}(t_i)) \subseteq T'$, therefore $I^N, t' \models_{\mathcal{P}} \alpha_1$. This conflicts with our supposition. We conclude that there is no $t' \in \min_{\prec_N}(t)$ s.t. $I^N, t' \not\models_{\mathcal{P}} \alpha_1$, and therefore $I^N, t \models_{\mathcal{P}} \Box \alpha_1$.

□

A.5 Proof of results in Section 4.6

NB: The results marked (*) are introduced here, while they are omitted in the main text.

Proposition 4.51. *Let $S = (i, \pi, V_S, \prec_S)$ be a periodical preferential structure, $\mathbf{l}(S) = (V, \prec)$ be its corresponding interpretation and $t, t', t_S, t'_S \in \mathbb{N}$ s.t.:*

$$t_S = \begin{cases} t, & \text{if } t < i; \\ i + (t - i) \bmod \pi, & \text{otherwise.} \end{cases} \quad t'_S = \begin{cases} t', & \text{if } t' < i; \\ i + (t' - i) \bmod \pi, & \text{otherwise.} \end{cases}$$

We have the following: $t' \in \min_{\prec}(t)$ iff $t'_S \in \min_{\prec_S}(t_S)$.

Proof. Let $S = (i, \pi, V_S, \prec_S)$ be a periodical preferential structure, $\mathbf{l}(S) = (V, \prec)$ and $t, t' \in \mathbb{N}$.

- For the only-if part, we use proof by contradiction. We assume that $t' \in \min_{\prec}(t)$ and suppose that $t'_S \notin \min_{\prec_S}(t_S)$. Following the assumption, $t' \in \min_{\prec}(t)$ means that there is no $t'' \in [t, \infty[$ s.t. $(t'', t') \in \prec$. On the other hand, $t'_S \notin \min_{\prec_S}(t_S)$ means that there exists $t''_S \in [\min_{\prec}\{t_S, i\}, i + \pi[$ with $(V_S(t''_S), V_S(t'_S)) \in \prec_S$ (Note that t'_S is also in $[\min_{\prec}\{t_S, i\}, i + \pi[$). Note that following Definition 4.49, we have $V_S(t_S) = V(t_S)$, $V_S(t'_S) = V(t'_S)$ and $V_S(t''_S) = V(t''_S)$. Knowing that $t'_S, t''_S \in [0, i + \pi[$ and $(V_S(t''_S), V_S(t'_S)) \in \prec_S$, then we have $(t''_S, t'_S) \in \prec$. We discuss two cases: $t \in [0, i[$ and $t \in [i, \infty[$.
 - If $t \in [0, i[$, then we have $t = t_S$ and $t''_S \in [t, i + \pi[$ ($t = t_S = [\min_{\prec}\{t_S, i\}, i + \pi[$). Thanks to Definition 4.49, since $t'_S = t'$ in the case of $t' \in [0, i[$ and $t'_S = i + (t' - i) \bmod \pi$ in the case of $t' \in [i, \infty[$, then we have $V(t') = V(t'_S)$. Moreover, since $\mathbf{l}(S) \in \mathcal{I}^{sd}$, $V(t') = V(t'_S)$ and $(t''_S, t'_S) \in \prec$, then we have $(t''_S, t') \in \prec$. This conflicts with the assumption of $t' \in \min_{\prec}(t)$.
 - If $t \in [i, \infty[$, then $t_S, t'_S, t''_S \in [i, i + \pi[$ ($t_S \geq i$ and therefore $i = \min_{\prec}\{t_S, i\}, i + \pi[$ for both t'_S, t''_S). This entails that $t_S, t'_S, t''_S \in \text{final}(\mathbf{l}(S))$. On the hand we have $V(t) = V_S(t_S)$ and $V(t') = V_S(t'_S)$ thanks to Definition 4.49. Thanks to Proposition 4.7, since t''_S and t are in $\text{final}(\mathbf{l}(S))$, then there exists $t''' \in [t, \infty[$ where $V(t''') = V(t''_S)$. Since $\mathbf{l}(S)$, $V(t''') = V(t''_S)$ and $V(t') = V_S(t'_S)$. This conflicts with the assumption of $t' \in \min_{\prec}(t)$.
- For the if part, we also use proof by contradiction. We assume that $t'_S \in \min_{\prec_S}(t_S)$ and suppose that $t' \notin \min_{\prec}(t)$. Following the assumption, $t'_S \in \min_{\prec_S}(t_S)$ entails that there is no $t''_S \in [\min_{\prec}\{t_S, i\}, i + \pi[$ with $(V_S(t''_S), V_S(t'_S)) \in \prec_S$. On the other hand, $t' \notin \min_{\prec}(t)$ means that there exists $t''' \in [t, \infty[$ where $(t''', t') \in \prec$. Let t'''_S be its corresponding points on the periodical preferential structure S . t'''_S is defined as follows:

$$t'''_S = \begin{cases} t''', & \text{if } t''' < i; \\ i + (t''' - i) \bmod \pi, & \text{otherwise.} \end{cases}$$

The definition of t_S, t'_S, t'''_S in this proposition has two results. The first of which is that t_S, t'_S, t'''_S are all in $[0, i + \pi[$. The second is that, thanks to Definition 4.49, we have $V_S(t_S) = V(t)$, $V_S(t'_S) = V(t')$ and $V_S(t'''_S) = V(t''')$. Moreover, since $(t''', t') \in \prec$, then $(V(t'''), V(t')) \in \prec_S$ and therefore $(V_S(t'''_S), V_S(t'_S)) \in \prec_S$. Next we show that $t'''_S \in [\min_{<} \{t_S, i\}, i + \pi[$. We discuss two cases.

- If $t'''_S \in [0, i[$, then we have $t'''_S = t'''$. Moreover, since $t''' \geq t$, we also have $t \in [0, i[$, therefore $t_S = t$ and $t'''_S \in [t_S, i + \pi[$. Thus, we have $t'''_S \in [\min_{<} \{t_S, i\}, i + \pi[$.
- The other case is when $t'''_S \in [i, i + \pi[$. It follows that $t'''_S \in [\min_{<} \{t_S, i\}, i + \pi[$.

Since $(V_S(t'''_S), V_S(t'_S)) \in \prec_S$ and $t'''_S \in [\min_{<} \{t_S, i\}, i + \pi[$, then $t'_S \notin \min_{<} (t_S)$. This conflicts with the assumption of $t'_S \in \min_{<} (t_S)$.

□

Definition A.8 (*). Given a UPI $I = (V, \prec)$, we define the periodical preferential structure $S(I) = (i, \pi, V_S, \prec_S)$ by:

- $i = \text{length}(\text{init}(I))$, $\pi = \text{card}(\text{range}(I))$;
- $V_S(t) = V(t)$ for all $t \in [0, i + \pi[$;
- for all $t, t' \in [0, i + \pi[$, $(V(t), V(t')) \in \prec_S$ iff $(t, t') \in \prec$.

It is worth to note that Definition A.8 is possible because I is an UPI. In particular, UPIs are state-dependent interpretations, i.e., in \mathcal{I}^{sd} . Therefore, for each t, t', t'', t''' where $V(t) = V(t')$ and $V(t'') = V(t''')$, then $t \prec t''$ iff $t'' \prec t'''$. Thus, it is possible to have a compact representation of the preference relation of UPIs.

Next, we shall show that given an UPI I , the induced interpretation from the periodical preferential structure $\mathbf{l}(S(I))$ and I are the same. Let $I = (V, \prec)$ be an UPI and $S(I) = (i, \pi, V_S, \prec_S)$ be its periodical preferential structure. Let $\mathbf{l}(S(I)) = (V', \prec')$ be the induced interpretation of $S(I)$. Since $V_S(t) = V(t)$ for all $t \in [0, i + \pi[$ and $V'(t) = V_S(t)$ for all $t \in [0, i + \pi[$, then $V'(t) = V(t)$ for all $t \in [0, i + \pi[$. Given a $t \in [i + \pi, \infty[$, since $V(t) = V(i + (t - i) \bmod \pi)$ (see Definition 4.46, note that $t_I = \text{length}(\text{init}(I)) = i$ and $\pi = \text{card}(\text{range}(I))$) and $i + (t - i) \bmod \pi \in [i, i + \pi[$, then $V_S(i + (t - i) \bmod \pi) = V(i + (t - i) \bmod \pi)$. On the other hand, we have $V'(t) = V_S(i + (t - i) \bmod \pi)$ for $t \in [i + \pi, \infty[$ (see Definition 4.49), then we have $V'(t) = V(t)$. Therefore, for any $t \in \mathbb{N}$, we have $V'(t) = V(t)$. Moreover, given any $(t, t') \in \prec$, we have $((V(t), V(t')) \in \prec_S)$. Since $(V(t), V(t')) \in \prec_S$, $V(t) = V'(t)$ and $V(t') = V'(t')$, then we have $(t, t') \in \prec'$. The if part follows the same reasoning. Therefore for any $t, t' \in \mathbb{N}$, we have $(t, t') \in \prec$ iff $(t, t') \in \prec'$.

Proposition 4.55. Let $\alpha \in \mathcal{L}^*$. We have that α is \mathcal{I}^{sd} -satisfiable iff there exists a periodical preferential structure S such that $\mathbf{l}(S), 0 \models \alpha$ and $\text{size}(\mathbf{l}(S)) \leq |\alpha| \times 2^{|\mathcal{P}|}$.

Proof. Let $\alpha \in \mathcal{L}^*$.

- For the only if part, let α be \mathcal{I}^{sd} -satisfiable. Thanks to Theorem 4.30 and Proposition 4.47, there exists a UPI $I = (V, \prec) \in \mathcal{I}^{sd}$ s.t. $I, 0 \models \alpha$ and $\text{size}(I) \leq |\alpha| \times 2^{|\mathcal{P}|}$. We define the periodical preferential structure $S(I)$ from I . Since I and $\mathbf{l}(S(I))$ are the same interpretation, then from \mathcal{I}^{sd} -satisfiable sentence α , we can find a periodical preferential structure S such that $\mathbf{l}(S), 0 \models \alpha$ and $\text{size}(\mathbf{l}(S)) \leq |\alpha| \times 2^{|\mathcal{P}|}$.

- For the if part, let $S = (i, \pi, V_S, \prec_S)$ be a periodical preferential structure s.t. $I(S), 0 \models \alpha$. Since $I(S) \in \mathfrak{J}^{sd}$, therefore α is \mathfrak{J}^{sd} -satisfiable.

□

Lemma 4.54 is a particular case of the following Lemma.

Lemma A.9 (*). *Let $S = (i, \pi, V_S, \prec_S)$, $\alpha \in \mathcal{L}^*$ be a periodical preferential structure, $\alpha_1 \in Sf(\alpha)$ and $t, t' \in \mathbb{N}$ such that:*

$$t' = \begin{cases} t, & \text{if } t < i; \\ i + (t - i) \bmod \pi, & \text{otherwise.} \end{cases}$$

We have $I(S), t \models \alpha_1$ iff $\alpha_1 \in lab_\alpha^S(t')$.

Proof. Let $S = (i, \pi, V_S, \prec_S)$ be a periodical preferential structure, $\alpha \in \mathcal{L}^*$, $\alpha_1 \in Sf(\alpha)$, $t \in \mathbb{N}$ and $I(S) = (V, \prec)$. We use structural induction on α_1 to prove the Lemma. Let t' be a time point s.t. $t' = t$, if $t \in [0, i]$, and $t' = i + (t - i) \bmod \pi$, if $t \in [i, \infty[$.

- $\alpha_1 = p$. If $t \in [0, i]$, then we have $V_S(t') = V(t)$. Thus $p \in V_S(t)$ iff $p \in V(t)$, and therefore $I(S), t \models p$ iff $p \in lab_\alpha^S(t)$. If $t \in [i, \infty[$, we have $V_S(t') = V(t)$. Following the same reasoning as the previous case, $I(S), t \models p$ iff $p \in lab_\alpha^S(t')$.
- $\alpha_1 = \neg\alpha_2$. By the induction hypothesis, we have $I(S), t \models \alpha_2$ iff $\alpha_2 \in lab_\alpha^S(t')$, and therefore $I(S), t \not\models \alpha_2$ iff $\alpha_2 \notin lab_\alpha^S(t')$. We conclude that $I(S), t \models \neg\alpha_2$ iff $\neg\alpha_2 \in lab_\alpha^S(t')$.
- $\alpha_1 = \alpha_2 \wedge \alpha_3$. By the induction hypothesis, we have $I(S), t \models \alpha_2$ iff $\alpha_2 \in lab_\alpha^S(t')$ and $I(S), t \models \alpha_3$ iff $\alpha_3 \in lab_\alpha^S(t')$, and therefore $I(S), t \models \alpha_2 \wedge \alpha_3$ iff $\alpha_2 \wedge \alpha_3 \in lab_\alpha^S(t')$.
- $\alpha_1 = \Diamond\alpha_2$.
 - For the only-if part, let $I(S), t \models \Diamond\alpha_2$. We have $t_2 \in [t, \infty[$ s.t. $I(S), t_2 \models \alpha_2$. Depending on where t_2 is, there exists a t'_2 s.t. $t'_2 = t_2$ if $t_2 \in [0, i]$ and $t'_2 = i + (t_2 - i) \bmod \pi$ if $t_2 \in [i, \infty[$. By the induction hypothesis on α_2 , we have $\alpha_2 \in lab_\alpha^S(t'_2)$. Note that t'_2 is $[0, i + \pi[$. Next, we show that $t'_2 \in [\min_{<}\{t', i\}, i + \pi[$. When $t'_2 \in [0, i]$, then we have $t' = t$ and $t'_2 = t_2$. Since $t \leq t_2$, then $t'_2 \in [t', i]$ and without a loss of generality $t'_2 \in [\min_{<}\{t', i\}, i + \pi[$. When $t'_2 \in [i, i + \pi[$, it follows that $t'_2 \in [\min_{<}\{t', i\}, i + \pi[$. In both cases, since $\alpha_2 \in lab_\alpha^S(t'_2)$ and $t'_2 \in [\min_{<}\{t', i\}, i + \pi[$, we conclude that $\Diamond\alpha_2 \in lab_\alpha^S(t')$ (see Definition 4.53).
 - For the if part, let $I(S), t \not\models \Diamond\alpha_2$. $I(S), t \models \neg\Diamond\alpha_2$ means that for all $t_2 \geq t$ we have $I(S), t_2 \not\models \alpha_2$. By the induction hypothesis, for all $t_2 \geq t$, we have $\alpha_2 \notin lab_\alpha^S(t'_2)$ where $t'_2 = t_2$ if $t_2 \in [0, i]$ and $t'_2 = i + (t_2 - i) \bmod \pi$ if $t_2 \in [i, \infty[$. Following the same reasoning as the only-part proof, we can check that for all $t_2 \geq t$, we have $t'_2 \in [\min_{<}\{t', i\}, i + \pi[$. Therefore $\alpha_2 \notin lab_\alpha^S(t'_2)$ for all $t'_2 \in [\min_{<}\{t', i\}, i + \pi[$. Going back to Definition 4.53, we conclude that $\Diamond\alpha_2 \notin lab_\alpha^S(t')$.
- $\alpha_1 = \Diamond_S\alpha_2$.
 - For the only-if part, let $I(S), t \models \Diamond_S\alpha_2$. We have $t_2 \in \min_{<}(t)$ s.t. $I(S), t_2 \models \alpha_2$. Depending on where t_2 is, there exists a t'_2 s.t. $t'_2 = t_2$ if $t_2 \in [0, i]$ and $t'_2 = i + (t_2 - i) \bmod \pi$ if $t_2 \in [i, \infty[$. By the induction hypothesis on α_2 , we have $\alpha_2 \in lab_\alpha^S(t'_2)$. From Proposition 4.51, we can see that $t_2 \in \min_{<}(t)$ iff $t'_2 \in \min_{<_S}(t')$. Going back to Definition 4.53, since there is $t'_2 \in \min_{<_S}(t')$ where $\alpha_2 \in lab_\alpha^S(t'_2)$, we conclude that $\Diamond_S\alpha_2 \in lab_\alpha^S(t')$.

- For the if part, let $I(S), t \not\models \Diamond \alpha_2$. $I(S), t \models \neg \Diamond \alpha_2$ means that for all $t_2 \in \min_{\prec}(t)$ we have $I(S), t_2 \not\models \alpha_2$. By the induction hypothesis on α_2 , for all $t_2 \in \min_{\prec}(t)$, we have (I) $\alpha_1 \notin \text{lab}_{\alpha}^S(t'_2)$ where $t'_2 = t_2$ if $t_2 \in [0, i[$ and $t'_2 = i + (t_2 - i) \bmod \pi$ if $t_2 \in [i, \infty[$. From Proposition 4.51, we can see that (II) $t_2 \in \min_{\prec}(t)$ iff $t'_2 \in \min_{\prec_S}(t')$ for all $t_2 \geq t$. Going back to Definition 4.53, since $\alpha_2 \in \text{lab}_{\alpha}^S(t'_2)$ for all $t'_2 \in \min_{\prec_S}(t')$, we conclude that $\Diamond \alpha_1 \notin \text{lab}_{\alpha}^S(t')$.

□

Bibliography

- [AA00] Ofer ARIELI and Arnon AVRON. « General patterns for nonmonotonic reasoning: from basic entailments to plausible relations ». *Logic Journal of the IGPL*, 8(2):119–148, March 2000.
- [ABHL20] Alex ABUIN, Alexander BOLOTOV, Montserrat HERMO, and Paqui LUCIO. « One-Pass Context-Based Tableaux Systems for CTL and ECTL ». In *27th International Symposium on Temporal Representation and Reasoning (TIME 2020)*, volume 178 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:20, Dagstuhl, Germany, 2020. Schloss DagstuhlLeibniz-Zentrum für Informatik.
- [Ada65] Ernest ADAMS. « The Logic of Conditionals ». *Inquiry: An Interdisciplinary Journal of Philosophy*, 8(1-4):166–197, 1965.
- [AGW09] Pietro ABATE, Rajeev GORÉ, and Florian WIDMANN. « An On-the-fly Tableau-based Decision Procedure for PDL-Satisfiability ». *Electronic Notes in Theoretical Computer Science*, 231:191–209, March 2009.
- [ASB⁺94] Adnan AZIZ, Vigyan SINGHAL, Felice BALARIN, Robert BRAYTON, and Alberto SANGIOVANNI-VINCENTELLI. « Equivalences for fair Kripke structures ». volume 820, 1994.
- [BA12a] Mordechai BEN-ARI. Temporal Logic: A Deductive System. In *Mathematical Logic for Computer Science*, pages 263–272. Springer, London, 2012.
- [BA12b] Mordechai BEN-ARI. Temporal Logic: Formulas, Models, Tableaux. In *Mathematical Logic for Computer Science*, pages 231–262. Springer, London, 2012.
- [BAMP81] Mordechai BEN-ARI, Zohar MANNA, and Amir PNUELI. « The temporal logic of branching time ». In *Proceedings of the 8th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '81, pages 164–176, New York, NY, USA, January 1981. Association for Computing Machinery.
- [BBCR08] Nicolas BEDON, Alexis BÈS, Olivier CARTON, and Chloé RISPAL. « Logic and Rational Languages of Words Indexed by Linear Orderings ». In *Computer Science Theory and Applications*, Lecture Notes in Computer Science, pages 76–85, Berlin, Heidelberg, 2008. Springer.
- [BCD⁺93] Salem BENFERHAT, Claudette CAYROL, Didier DUBOIS, Jérôme LANG, and Henri PRADE. « Inconsistency Management and Prioritized Syntax-Based Entailment ». In *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, pages 640–647. Morgan Kaufmann, 1993.

-
- [BCM⁺92] Jerry Robert BURCH, Edmund Melson CLARKE, Kenneth Lauchlin MCMILLAN, David L DILL, and L J HWANG. « Symbolic model checking: 10^{20} States and beyond ». *Information and Computation*, 98(2):142–170, 1992.
- [BCM⁺07] Franz BAADER, Diego CALVANESE, Deborah MCGUINNESS, Daniele NARDI, and Peter PATEL-SCHNEIDER. *The Description Logic Handbook: Theory, Implementation, and Applications*. January 2007.
- [BCM⁺20] Katarina BRITZ, Giovanni CASINI, Thomas MEYER, Kody MOODLEY, Uli SATTler, and Ivan VARZINCZAK. « Principles of KLM-style Defeasible Description Logics ». *ACM Transactions on Computational Logic*, 22:1–46, 2020.
- [BDP97] Salem BENFERHAT, Didier DUBOIS, and Henri PRADE. « Nonmonotonic reasoning, conditional objects and possibility theory ». *Artificial Intelligence*, 92(1):259–276, 1997.
- [Ben94] Salem BENFERHAT. « *Raisonnement non-monotone et traitement de l'inconsistance en logique possibiliste* ». These de doctorat, Toulouse 3, January 1994.
- [Bes89] Philippe BESNARD. *An introduction to default logic*. Symbolic computation. Springer, 1989.
- [BG85] John P. BURGESS and Yuri GUREVICH. « The decision problem for linear temporal logic. ». *Notre Dame Journal of Formal Logic*, 26(2):115–128, April 1985.
- [BGM⁺19] Matteo BERTELLO, Nicola GIGANTE, Angelo MONTANARI, and Mark REYNOLDS. « Leviathan: A New LTL Satisfiability Checking Tool Based on a One-Pass Tree-Shaped Tableau ». July 2019.
- [BK08] Christel BAIER and Joost-Pieter KATOEN. *Principles of Model Checking*. MIT Press, Cambridge, MA, USA, April 2008.
- [BLS11] Andreas BAUER, Martin LEUCKER, and Christian SCHALLHART. « Runtime Verification for LTL and TLTL ». *ACM Transactions on Software Engineering and Methodology*, 20(4):14:1–14:64, September 2011.
- [BMV11a] Katarina BRITZ, Thomas MEYER, and Ivan VARZINCZAK. « Preferential Reasoning for Modal Logics ». *Electronic Notes in Theoretical Computer Science*, 278:55–69, November 2011.
- [BMV11b] Katarina BRITZ, Thomas MEYER, and Ivan VARZINCZAK. « Semantic Foundation for Preferential Description Logics ». In *AI 2011: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 491–500, Berlin, Heidelberg, 2011. Springer.
- [BMV12] Richard BOOTH, Thomas MEYER, and Ivan VARZINCZAK. « PTL: A Propositional Typicality Logic ». In *Logics in Artificial Intelligence*, Lecture Notes in Computer Science, pages 107–119, Berlin, Heidelberg, 2012. Springer.
- [Bou94] Craig BOUTILIER. « Conditional logics of normality: A modal approach ». *Artificial Intelligence*, 68(1):87–154, July 1994.
- [BS85] Genevieve BOSSU and Pierre SIEGEL. « Saturation, nonmonotonic reasoning and the closed-world assumption ». *Artificial Intelligence*, 25(1):13–63, January 1985.

- [BV18] Katarina BRITZ and Ivan VARZINCZAK. « From KLM-style conditionals to defeasible modalities, and back ». *Journal of Applied Non-Classical Logics*, 28(1):92–121, January 2018.
- [BV19] Katarina BRITZ and Ivan VARZINCZAK. « Preferential Tableaux for Contextual Defeasible \mathcal{ALC} ». In *Automated Reasoning with Analytic Tableaux and Related Methods*, Lecture Notes in Computer Science, pages 39–57, Cham, 2019. Springer International Publishing.
- [BvB07] Patrick BLACKBURN and Johan van BENTHEM. 1 Modal logic: a semantic perspective. In *Studies in Logic and Practical Reasoning*, volume 3 of *Handbook of Modal Logic*, pages 1–84. Elsevier, January 2007.
- [CCACV20] Anasse CHAFIK, Fahima CHEIKH-ALILI, Jean-François CONDOTTA, and Ivan VARZINCZAK. « On the Decidability of a Fragment of preferential LTL ». In *27th International Symposium on Temporal Representation and Reasoning (TIME 2020)*, volume 178 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:19, Dagstuhl, Germany, 2020. Schloss DagstuhlLeibniz-Zentrum für Informatik.
- [CCACV21] Anasse CHAFIK, Fahima CHEIKH-ALILI, Jean-François CONDOTTA, and Ivan VARZINCZAK. « A One-Pass Tree-Shaped Tableau for Defeasible LTL ». In *28th International Symposium on Temporal Representation and Reasoning (TIME 2021)*, volume 206 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl Leibniz-Zentrum für Informatik.
- [CCG⁺02] Alessandro CIMATTI, Edmund CLARKE, Enrico GIUNCHIGLIA, Fausto GIUNCHIGLIA, Marco PISTORE, Marco ROVERI, Roberto SEBASTIANI, and Armando TACCHIELLA. « NuSMV 2: An OpenSource Tool for Symbolic Model Checking ». In *Computer Aided Verification*, Lecture Notes in Computer Science, pages 359–364, Berlin, Heidelberg, 2002. Springer.
- [CE82] Edmund Melson CLARKE and Ernest Allen EMERSON. « Design and synthesis of synchronization skeletons using branching time temporal logic ». In *Logics of Programs*, Lecture Notes in Computer Science, pages 52–71, Berlin, Heidelberg, 1982. Springer.
- [CES86] Edmund CLARKE, Ernest EMERSON, and Aravinda SISTLA. « Automatic Verification of Finite State Concurrent Systems Using Temporal Logic Specifications ». *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8:244–263, 1986.
- [CGH94] Edmund CLARKE, Orna GRUMBERG, and Kiyoharu HAMAGUCHI. « Another Look at LTL Model Checking ». In *Formal Methods in System Design*, pages 415–427. Springer-Verlag, 1994.
- [Che80] Brian CHELLAS. *Modal Logic: An Introduction*. Cambridge University Press, Cambridge, 1980.
- [CHV02] Marcos A. CASTILHO, Andreas HERZIG, and Ivan José VARZINCZAK. « It depends on the context! A decidable logic of actions and plans based on a ternary dependence relation ». In *9th International Workshop on Non-Monotonic Reasoning (NMR 2002)*, April 19-21, Toulouse, France, *Proceedings*, pages 343–348, 2002.
- [CHV18] Edmund M. CLARKE, Thomas A. HENZINGER, and Helmut VEITH. Introduction to Model Checking. In *Handbook of Model Checking*, pages 1–26. Springer International Publishing, Cham, 2018.

-
- [DR10] Stephane DEMRI and Alexander RABINOVICH. « The complexity of linear-time temporal logic over the class of ordinals ». *Logical Methods in Computer Science*, 6(4):9, December 2010. arXiv: 1009.5206.
- [DS02] Stéphane DEMRI and Philippe SCHNOEBELEN. « The Complexity of Propositional Linear Temporal Logics in Simple Cases ». *Information and Computation*, 174(1):84–103, April 2002.
- [EC80] Ernest Allen EMERSON and Edmund Melson CLARKE. « Characterizing Correctness Properties of Parallel Programs Using Fixpoints ». In *ICALP*, 1980.
- [EH83] Ernest Allen EMERSON and Joseph Yehuda HALPERN. « "Sometimes" and "not never" revisited: on branching versus linear time (preliminary report) ». In *Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, POPL '83, pages 127–140, New York, NY, USA, January 1983. Association for Computing Machinery.
- [EH85] Ernest Allen EMERSON and Joseph Yehuda HALPERN. « Decision procedures and expressiveness in the temporal logic of branching time ». *Journal of Computer and System Sciences*, 30(1):1–24, 1985.
- [FG16] Valeria FIONDA and Gianluigi GRECO. « The Complexity of LTL on Finite Traces: Hard and Easy Fragments ». *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), February 2016.
- [Fre91] Michael FREUND. « A semantic characterization of disjunctive relations ». In *Fundamentals of Artificial Intelligence Research*, Lecture Notes in Computer Science, pages 71–83, Berlin, Heidelberg, 1991. Springer.
- [Fre93] Michael FREUND. « Injective Models and Disjunctive Relations ». *Journal of Logic and Computation*, 3(3):231–247, 1993.
- [Gab85] Dov GABBAY. « Theoretical Foundations for Non-Monotonic Reasoning in Expert Systems ». In *Logics and Models of Concurrent Systems*, NATO ASI Series, pages 439–457, Berlin, Heidelberg, 1985. Springer.
- [Gab89] Dov GABBAY. « The declarative past and imperative future ». In *Temporal Logic in Specification*, Lecture Notes in Computer Science, pages 409–448, Berlin, Heidelberg, 1989. Springer.
- [Gef89] Hector GEFNER. « Default Reasoning, Minimality and Coherence ». In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR'89). Toronto, Canada, May 15-18 1989*, pages 137–148. Morgan Kaufmann, 1989.
- [GG21] Laura GIORDANO and Valentina GLIOZZI. « A reconstruction of multipreference closure ». *Artificial Intelligence*, 290:103398, January 2021.
- [GGOP07] Laura GIORDANO, Valentina GLIOZZI, Nicola OLIVETTI, and Gian Luca POZZATO. « Preferential Description Logics ». In *Logic for Programming, Artificial Intelligence, and Reasoning*, Lecture Notes in Computer Science, pages 257–272, Berlin, Heidelberg, 2007. Springer.

- [GGOP09] Laura GIORDANO, Valentina GLIOZZI, Nicola OLIVETTI, and Gian Luca POZZATO. « Analytic tableaux calculi for KLM logics of nonmonotonic reasoning ». *ACM Transactions on Computational Logic*, 10(3):18:1–18:47, April 2009.
- [GGOP13] Laura GIORDANO, Valentina GLIOZZI, Nicola OLIVETTI, and Gian Luca POZZATO. « A non-monotonic Description Logic for reasoning about typicality ». *Artificial Intelligence*, 195:165–202, February 2013.
- [GGOP15] Laura GIORDANO, Valentina GLIOZZI, Nicola OLIVETTI, and Gian Luca POZZATO. « Semantic characterization of rational closure: From propositional logic to description logics ». *Artificial Intelligence*, 226:1–33, September 2015.
- [GMM14] Giuseppe De GIACOMO, Riccardo De MASELLIS, and Marco MONTALI. « Reasoning on LTL on Finite Traces: Insensitivity to Infiniteness ». In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1027–1033. AAAI Press, 2014.
- [GMR17] Nicola GIGANTE, Angelo MONTANARI, and Mark REYNOLDS. « A One-Pass Tree-Shaped Tableau for LTL+Past ». In *LPAR*, 2017.
- [GOGP09] Laura GIORDANO, Nicola OLIVETTI, Valentina GLIOZZIC, and Gian Luca POZZATO. « ALC + T: a Preferential Extension of Description Logics ». *Fundamenta Informaticae*, 96(3):341–372, January 2009.
- [Gor99] Rajeev GORÉ. *Tableau Methods for Modal and Temporal Logics*. In *Handbook of Tableau Methods*, pages 297–396. Springer Netherlands, Dordrecht, 1999.
- [Gou89] Graham D. GOUGH. « Decision procedures for temporal logic ». Master’s thesis, University of Manchester, 1989.
- [GPVW95] Rob GERTH, Doron A. PELED, Moshe Y. VARDI, and Pierre WOLPER. « Simple on-the-fly automatic verification of linear temporal logic ». In *Protocol Specification, Testing and Verification XV, Proceedings of the Fifteenth IFIP WG6.1 International Symposium on Protocol Specification, Testing and Verification, Warsaw, Poland, June 1995*, volume 38 of *IFIP Conference Proceedings*, pages 3–18. Chapman & Hall, 1995.
- [GV13] Giuseppe De GIACOMO and Moshe Y. VARDI. « Linear Temporal Logic and Linear Dynamic Logic on Finite Traces ». In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 854–860. IJCAI/AAAI, 2013.
- [HFS⁺14] Lothar HOTZ, Alexander FELFERNIG, Markus STUMPTNER, Anna RYABOKON, Claire BAGLEY, and Katharina WOLTER. Chapter 6 - Configuration Knowledge Representation and Reasoning. In *Knowledge-Based Configuration*, pages 41–72. Morgan Kaufmann, Boston, January 2014.
- [HK03] Ullrich HUSTADT and Boris KONEV. « TRP++ 2.0: A Temporal Resolution Prover ». In *Automated Deduction CADE-19, Lecture Notes in Computer Science*, pages 274–278, Berlin, Heidelberg, 2003. Springer.
- [Kam68] Hans KAMP. « *Tense Logic and the Theory of Linear Order* ». PhD Thesis, UCLA, 1968.

-
- [KLM90] Sarit KRAUS, Daniel LEHMANN, and Menachem MAGIDOR. « Nonmonotonic reasoning, preferential models and cumulative logics ». *Artificial Intelligence*, 44(1):167–207, July 1990.
- [KMMP93] Yonit KESTEN, Zohar MANNA, Hugh MCGUIRE, and Amir PNUELI. « A decision algorithm for full propositional temporal logic ». In *Computer Aided Verification*, Lecture Notes in Computer Science, pages 97–109, Berlin, Heidelberg, 1993. Springer.
- [Kri59] Saul A. KRIPKE. « A completeness theorem in modal logic1 ». *The Journal of Symbolic Logic*, 24(1):1–14, March 1959.
- [Kri80] Saul A. KRIPKE. *Naming and necessity*. 1980. OCLC: 5726909.
- [Lew73] David LEWIS. « Counterfactuals and comparative possibility ». *Journal of Philosophical Logic*, 2(4):418–446, October 1973.
- [LL05] Noël LAVERNY and Jérôme LANG. « From Knowledge-based Programs to Graded Belief-based Programs, Part I: On-line Reasoning* ». *Synthese*, 147(2):277–321, November 2005.
- [LM89] Daniel LEHMANN and Menachem MAGIDOR. « What does a Conditional Knowledge Base Entail? ». In *Artif. Intell.*, 1989.
- [LPZ85] Orna LICHTENSTEIN, Amir PNUELI, and Lenore ZUCK. « The glory of the past ». In *Logics of Programs*, pages 196–218. Springer, Berlin, Heidelberg, June 1985.
- [LW21] Yongming LI and Jieli WEI. « Possibilistic Fuzzy Linear Temporal Logic and Its Model Checking ». *IEEE Transactions on Fuzzy Systems*, 29(7):1899–1913, July 2021.
- [LYP⁺14] Jianwen LI, Yinbo YAO, Geguang PU, Lijun ZHANG, and Jifeng HE. « Aalta: an LTL satisfiability checker over Infinite/Finite traces ». In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2014, pages 731–734, New York, NY, USA, November 2014. Association for Computing Machinery.
- [Mak87] David MAKINSON. « On the status of the postulate of recovery in the logic of theory change ». *Journal of Philosophical Logic*, 16(4):383–394, November 1987.
- [Mak89] David MAKINSON. « General theory of cumulative inference ». In *Non-Monotonic Reasoning*, Lecture Notes in Computer Science, pages 1–18, Berlin, Heidelberg, 1989. Springer.
- [Mak05] David MAKINSON. How to Go Nonmonotonic. In *Handbook of Philosophical Logic*, Handbook of Philosophical Logic, pages 175–278. Springer Netherlands, Dordrecht, 2005.
- [Mar03] Nicolas MARKEY. « Temporal logic with past is exponentially more succinct, Concurrency Column ». *Bull. EATCS*, 79:122–128, 2003.
- [McC80] John MCCARTHY. « CircumscriptionA form of non-monotonic reasoning ». *Artificial Intelligence*, 13(1):27–39, 1980.
- [MH69] John MCCARTHY and Patrick HAYES. Some Philosophical Problems From the Standpoint of Artificial Intelligence. In *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

- [Moi88] Yves MOINARD. « *Raisonnement non-monotone : contribution a l'etude de la circonscription* ». These de doctorat, Rennes 1, January 1988.
- [Moo93] Robert C. MOORE. « Autoepistemic logic revisited ». *Artificial Intelligence*, 59(1):27–30, 1993.
- [MP95] Zohar MANNA and Amir PNUELI. *Temporal verification of reactive systems - safety*. Springer, 1995.
- [Muk97] Madhavan MUKUND. « Linear-time temporal logic and Büchi automata ». *Tutorial talk, Winter School on Logic and Computer Science, Indian Statistical Institute, Calcutta*, Page 8, 1997.
- [Pea89] Judea PEARL. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.
- [Pea90] Judea PEARL. « System Z: A Natural Ordering of Defaults with Tractable Applications to Nonmonotonic Reasoning ». In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge, Pacific Grove, CA, USA, March 1990*, pages 121–135. Morgan Kaufmann, 1990.
- [Pnu77] Amir PNUELI. « The temporal logic of programs ». In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57, October 1977. ISSN: 0272-5428.
- [Pri62] Arthur Norman PRIOR. « Tense-Logic and the Continuity of Time ». *Studia Logica: An International Journal for Symbolic Logic*, 13:133–151, 1962.
- [Rab12] Alexander RABINOVICH. « Temporal logics over linear time domains are in PSPACE ». *Information and Computation*, 210:40–67, January 2012.
- [RC81] Raymond REITER and Giovanni CRISCUOLO. « On Interacting Defaults ». In *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI '81, Vancouver, BC, Canada, August 24-28, 1981*, pages 270–276. William Kaufmann, 1981.
- [Rei80] Raymond REITER. « A logic for default reasoning ». *Artificial Intelligence*, 13(1):81–132, April 1980.
- [Rey00] Mark REYNOLDS. « More past glories [temporal logic] ». In *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No.99CB36332)*, pages 229–240, June 2000. ISSN: 1043-6871.
- [Rey03] Mark REYNOLDS. « The complexity of the temporal logic with until over general linear time ». *Journal of Computer and System Sciences*, 66(2):393–426, March 2003.
- [Rey16a] Mark REYNOLDS. « A New Rule for LTL Tableaux ». *Electronic Proceedings in Theoretical Computer Science*, 226:287–301, September 2016. arXiv: 1609.04102.
- [Rey16b] Mark REYNOLDS. « A traditional tree-style tableau for LTL ». *arXiv:1604.03962 [cs]*, April 2016. arXiv: 1604.03962.
- [RV07] Kristin Y. ROZIER and Moshe Y. VARDI. « LTL Satisfiability Checking ». In *Model Checking Software*, Lecture Notes in Computer Science, pages 149–167, Berlin, Heidelberg, 2007. Springer.

-
- [RV11] Kristin Y. ROZIER and Moshe Y. VARDI. « A Multi-encoding Approach for LTL Symbolic Satisfiability Checking ». In *FM 2011: Formal Methods*, pages 417–431, Berlin, Heidelberg, 2011. Springer.
- [SA19] Christian STRASSER and Aldo G. ANTONELLI. Non-monotonic Logic. In *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2019 edition, 2019.
- [SC85] Aravinda Prasad SISTLA and Edmund Melson CLARKE. « The complexity of propositional linear temporal logics ». *Journal of the ACM*, 32(3):733–749, July 1985.
- [Sch98] Stefan SCHWENDIMANN. « A New One-Pass Tableau Calculus for PLTL ». In *Automated Reasoning with Analytic Tableaux and Related Methods*, Lecture Notes in Computer Science, pages 277–291, Berlin, Heidelberg, 1998. Springer.
- [Sch02] Philippe SCHNOEBELEN. « The Complexity of Temporal Logic Model Checking ». *Advances in Modal Logic*, 4, November 2002.
- [SGL97] Peter H. SCHMITT and Jean GOUBAULT-LARRECQ. « A tableau system for linear-TIME temporal logic ». In *Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science, pages 130–144, Berlin, Heidelberg, 1997. Springer.
- [Sho87] Yoav SHOHAM. « A Semantical Approach to Nonmonotonic Logics ». In *Proceedings of the Symposium on Logic in Computer Science (LICS '87), Ithaca, New York, USA, June 22-25, 1987*, pages 275–279. IEEE Computer Society, 1987.
- [Sho88] Yoav SHOHAM. *Reasoning about change: time and causation from the standpoint of artificial intelligence*. MIT Press, Cambridge, MA, USA, 1988.
- [Smu68] Raymond M. SMULLYAN. Analytic Tableaux. In *First-Order Logic*, Ergebnisse der Mathematik und ihrer Grenzgebiete, pages 15–30. Springer, Berlin, Heidelberg, 1968.
- [Sta81] Robert C. STALNAKER. A Theory of Conditionals. In *IFS: Conditionals, Belief, Decision, Chance and Time*, The University of Western Ontario Series in Philosophy of Science, pages 41–55. Springer Netherlands, Dordrecht, 1981.
- [SW12] Martin SUDA and Christoph WEIDENBACH. « A PLTL-Prover Based on Labelled Superposition with Partial Model Guidance ». volume 7364, pages 537–543, June 2012.
- [TN16] Paulo TABUADA and Daniel NEIDER. « Robust Linear Temporal Logic ». In *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:21, Dagstuhl, Germany, 2016. Schloss DagstuhlLeibniz-Zentrum fuer Informatik.
- [Var96] Moshe Y. VARDI. An automata-theoretic approach to linear temporal logic. In *Logics for Concurrency: Structure versus Automata*, Lecture Notes in Computer Science, pages 238–266. Springer, Berlin, Heidelberg, 1996.
- [VW94] Moshe Y. VARDI and Pierre WOLPER. « Reasoning about Infinite Computations ». *Information and Computation*, 115(1):1–37, November 1994.
- [Wol83] Pierre WOLPER. « Temporal logic can be more expressive ». *Information and Control*, 56(1):72–99, January 1983.

- [Wol85] Pierre WOLPER. « The Tableau method for Temporal Logic: an overview ». *Logique et Analyse*, 28(110/111):119–136, 1985.

Résumé

Les logiques temporelles sont des outils formels de spécification et de vérification des systèmes informatiques. Le succès de ces logiques est principalement dû à leur syntaxe élégante et à leur sémantique intuitive permettant une représentation riche et une analyse fine des propriétés des systèmes évoluant au cours du temps. Différents outils ont été développés à partir de ces formalismes, notamment ceux basés sur des approches par automates et des méthodes des tableaux. Cependant, ces logiques restent limitées quand il s'agit de la modélisation et du raisonnement à propos de certains aspects des systèmes informatiques. En effet, les systèmes informatiques ne sont pas garantis totalement sûrs, et les propriétés que l'on souhaite vérifier peuvent avoir des exceptions triviales et tolérables, ou au contraire, des exceptions devant être gérées avec soin afin de garantir la fiabilité générale du système. De même, le comportement attendu d'un système peut ne pas être correct pour toutes ses exécutions possibles. Cependant, il doit l'être pour celles qui sont *les plus normales où les plus plausibles*.

La logique non-monotone est un domaine de recherche qui modélise l'aspect révisable du raisonnement du sens commun au delà du raisonnement déductif de la logique classique. De plus, il permet de raisonner avec des exceptions. L'objectif principal de ce travail est d'intégrer des approches non monotones aux logiques temporelles pour mieux représenter le comportement de systèmes tolérants aux exceptions.

Le formalisme présenté dans ce mémoire, appelé logique temporelle linéaire révisable, combine la syntaxe et la sémantique de LTL classique avec l'approche préférentielle KLM sur des inférences conditionnelles. Sa syntaxe contient une version révisable des opérateurs temporels permettant d'exprimer d'une façon plus souple des spécifications similaires à leurs contreparties classiques. Plus précisément, ces opérateurs permettent de considérer des points de temps admettant des exceptions lors des exécutions du système. LTL révisable étend les interprétations temporelles avec une relation de préférence afin de nuancer la prise en compte des différents points temporels.

Nous avons tout d'abord étudié la décidabilité du problème de la satisfiabilité des formules de cette nouvelle logique. Plus précisément, nous nous sommes intéressés à deux sous-langages de LTL révisable, nommés \mathcal{L}_1 et \mathcal{L}^* . Nous avons prouvé que le problème de satisfiabilité pour les formules de \mathcal{L}_1 est NP-complet. Concernant le fragment \mathcal{L}^* , nous avons pu démontrer la décidabilité de ce problème pour un ensemble d'interprétations nommées *state-dependent interpretations*. Nous avons montré que la propriété des modèles bornés est vérifiée pour les deux sous-langages. Grâce à ces propriétés, nous avons défini des structures et des méthodes pour résoudre le problème de satisfiabilité de ces deux fragments.

Dans un deuxième temps, nous avons défini une méthode des tableaux pour \mathcal{L}_1 en s'appuyant sur l'approche *one-pass tree-shaped tableaux* récemment proposée pour LTL classique. Nous avons adapté cette méthode en y intégrant la sémantique préférentielle KLM. La méthode introduite est basée sur un ensemble de règles relatives aux différents opérateurs (dites statiques), ainsi qu'un ensemble de règles dynamiques permettant de vérifier la cohérence des branches au fur et à mesure de l'exécution de la méthode. Nous avons prouvé que la méthode proposée est correcte et complète.

Nous terminons notre étude en proposant des perspectives concernant LTL révisable. Nous envisageons également une adaptation de la sémantique préférentielle sur d'autres formalismes temporels, notamment CTL et CTL*.

Mots-clés: Représentation des connaissances, logiques temporelles et modales, raisonnement non monotone.

Abstract

Temporal logics are formal tools for specifications and verification of computer systems. The success of these logics is mainly due to their elegant syntax and intuitive semantics that simplify the representation of system's properties that change over time. From automata based solvers to tableaux methods, several approaches are developed based on these formalisms. However, these logics remain limited when modeling and reasoning about certain aspects of computer systems. Indeed, computer systems are not guaranteed to be totally safe, and the properties one wishes to verify may have trivial and tolerable exceptions, or on the contrary, exceptions that must be carefully handled to guarantee the general reliability of the system. Similarly, the expected behaviour of a system may not be correct for every possible execution, but rather for its most "normal" or plausible executions.

Non-monotonic logic is a field of research that captures defeasible modes of reasoning which accurately represent common sense reasoning more than the deductive reasoning of classical logic. Moreover, it allows for reasoning with exceptions. The main objective of this work is to integrate non-monotonic approaches to temporal logics to better represent the behavior of exception-tolerant systems.

The formalism presented in this memoir is called defeasible linear temporal logic. This logic combines the syntax and semantics of LTL with the preferential KLM approach to conditional statements. The syntax contains a defeasible version of temporal operators, which express specifications similar to their classical counterparts, but are more lenient when time points may have exceptions during executions. Defeasible LTL extends temporal interpretations with a preference relation that nuances the degree of importance between time points.

We have studied the decidability of the satisfiability problem for defeasible LTL sentences. In this setting, we have considered two fragments of the language, which are \mathcal{L}_1 and \mathcal{L}^* . We proved that the satisfiability of \mathcal{L}_1 sentences is an NP-complete problem. As for the fragment \mathcal{L}^* , we showed the decidability of the problem for a class of interpretations called *state-dependent interpretations*. We have proved that the bounded-model property holds for both of these fragments. Thanks to these properties, we have introduced structures and methods for solving the satisfiability problem.

We have developed also a tableau method for \mathcal{L}_1 by adapting the recently proposed one-pass tree-shaped tableaux for classical LTL. The novelty of our approach is to show how KLM's preferential semantics work in a tableau for defeasible LTL. We have defined a set of static rules for different operators, as well as a set of dynamic rules for checking the correctness of branches at the same time of their expansion. We have proved that the method is sound and complete.

Finally, we investigate future work relating to defeasible LTL. We also plan to integrate preferential semantics to other temporal formalisms, namely CTL and CTL*.

Keywords: Knowledge representation, modal and temporal logics, non-monotonic reasoning.

