université
PARIS-SACLAY

Expressivity of first-order logic, star-free propositional dynamic logic and communicating automata

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Unité de recherche: Université Paris-Saclay, ENS Paris-Saclay, CNRS, LSV, 91190, Gif-sur-Yvette, France
Référent: ENS Paris-Saclay

Thèse présentée et soutenue en visioconférence totale, le 27 novembre 2020, par

Marie FORTIN

Composition du jury:

Jean-Marc TALBOT Professeur, Aix-Marseille Université	Président
Volker DIEKERT Professeur, Université de Stuttgart	Rapporteur
Blaise GENEST Directeur de recherche, CNRS, Université de Rennes	Rapporteur
Sylvain CONCHON Professeur, Université Paris-Saclay	Examinateur
Cristina SIRANGELO Professeure, Université de Paris	Examinatrice
Paul GASTIN Professeur, ENS Paris-Saclay	Directeur de thèse
Benedikt BOLLIG Directeur de recherche, CNRS, ENS Paris-Saclay	Co-encadrant de thèse

Thèse de doctorat

NNT: 2020UPASG046

Résumé

Cette thèse porte sur l'expressivité de la logique du premier ordre et d'autres formalismes sur différentes classes de structures ordonnées, parmi lesquelles les MSC (Message Sequence Charts), un modèle standard pour les exécutions de systèmes concurrents avec échange de messages. Cette étude est motivée par deux questions classiques : celle de l'équivalence, pour certaines classes de structures, entre la logique du premier ordre et son fragment avec k variables, et celle de la comparaison entre automates et logique, dans l'esprit du théorème de Büchi-Elgot-Trakhtenbrot.

Notre approche repose, pour les deux problèmes, sur la logique dynamique propositionnelle sans étoile (PDL sans étoile). Il s'agit d'une variante de PDL basée sur les opérations de concaténation, inverse, union et complément de relations binaires. Cette logique se rapproche de plusieurs formalismes classiques, et est équivalente à la logique du premier ordre avec 3 variables.

On étudie d'abord l'expressivité de PDL sans étoile, et donc de la logique du premier ordre avec 3 variables, sur des structures linéairement ordonnées avec des prédicats unaires et binaires. On montre que sous certaines conditions de monotonie, PDL sans étoile devient aussi expressive que la logique du premier ordre. Cela implique que toute formule de la logique du premier ordre peut alors être réécrite en une formule équivalente qui utilise au plus 3 variables. Ce résultat s'applique, directement ou indirectement, à un certain nombre de classes naturelles, généralisant des résultats connus (comme le cas des ordres linéaires, des réels avec la relation $+1$, ou des traces de Mazurkiewicz), et répondant à des questions ouvertes (réels avec des fonctions polynomiales, ordres linéaires avec des fonctions monotones).

On se concentre ensuite sur les MSC, auxquels ce premier résultat s'applique également. On montre que, sur les MSC, la syntaxe de PDL sans étoile peut être simplifiée pour obtenir une logique plus proche de la logique temporelle linéaire (LTL), mais toujours équivalente à la logique du premier ordre. Cela nous permet d'aborder un autre problème important: celui de la synthèse d'automates communicants à partir de spécifications écrites en logique du premier ordre. Les automates communicants sont un modèle de systèmes concurrents dans lequel un nombre fixé d'automates finis échangent des messages via des canaux FIFO. Ils définissent des langages de MSC. Bien que des caractérisations de l'expressivité des automates communicants en termes de fragments de la logique monadique du second ordre aient déjà été établies pour certaines restrictions (borne sur la taille des canaux de communications, ou omission de la relation "arrivé-avant" au niveau de la logique), la question suivante restait ouverte dans le cas général : toute formule du premier ordre sur les MSC peut-elle être traduite en un automate communicant équivalent ? On montre que c'est le cas, en utilisant PDL sans étoile comme langage intermédiaire.

Abstract

This thesis is concerned with the expressive power of first-order logic and other formalisms over different classes of ordered structures, among which MSCs (Message Sequence Charts), a standard model for executions of message-passing systems. This study is motivated by two classic problems: the k-variable property, that is, the equivalence of first-order logic and its k-variable fragment over certain classes of structures, and the study of logic-automata connections, in the spirit of Büchi-Elgot-Trakhtenbrot theorem.

Our approach to both problems relies on star-free propositional dynamic logic (star-free PDL), a variant of PDL based on the operations of concatenation, converse, union and complement of binary relations. It combines features from several classic formalisms, and has the same expressive power as the 3-variable fragment of first-order logic.

We start by studying the expressive power of star-free PDL, and thus of the 3-variable fragment of first-order logic, over linearly ordered structures with unary and binary predicates. We show that under certain monotonicity conditions, star-free PDL becomes as expressive as first-order logic. This implies that any first-order formula can then be rewritten into an equivalent formula with at most 3 variables. This result applies, directly or indirectly, to various natural classes of structures, generalizing several known results (such as the case of linear orders, the real line with the $+1$ relation, or Mazurkiewicz traces), and answering some open questions (reals with polynomial functions, linear orders with monotone functions).

We then focus on MSCs, to which this first result also applies. We show that, over MSCs, star-free PDL syntax can be simplified into a logic that is closer to linear temporal logic (LTL), but still equivalent to first-order logic. This gives us the means to address another important problem: the synthesis of communicating finite-state machines from first-order specifications. Communicating finite-state machines (CFMs) are a model of concurrent systems in which a fixed number of finite-state automata communicate through unbounded FIFO channels. They accept languages of MSCs. While characterizations of the expressive power of CFMs in terms of fragment of monadic second-order logic have been established under different restrictions (bounding the size of the communication channels, or removing the "happened-before" relation from the logic), the following question had remained open in the general case: can every first-order formula over MSCs be translated into an equivalent CFM? We prove that this is the case, using star-free PDL as an intermediate language.

Acknowledgments

I am deeply grateful to my advisors, Benedikt Bollig and Paul Gastin. You have introduced me to a wonderful research topic, and I am proud that part of my thesis is in continuity with Benedikt's own. Thank you for always being available, for your kindness, your invaluable advice, for helping me build the confidence to share and explore my own ideas, and also for being honest when something wasn't working out. Our discussions were incredibly fruitful, and I hope we will have many more in the future.

I would also like to thank Volker Diekert and Blaise Genest for reviewing my thesis and for their kind feedback. Many thanks to all the other members of the jury as well, Sylvain Conchon, Cristina Sirangelo, and Jean-Marc Talbot.

I was lucky to work in a great environment at LSV. I was surrounded by too many amazing people to list them all, but to all of you: thank you! My only regret is defending in the middle of a pandemic, and thus being unable to actually come to the lab for my defense or celebrate with you afterwards.

I would probably never have started on this path if not for the wonderful teachers and advisors I had as an undergraduate, who shared with me their love of mathematics, computer science, and research in general, and then logic and automata in particular. They had a huge influence on me at a time where I was still trying to decide what I wanted to do, so many thanks in particular to Guillaume Pagot, Hubert Comon, Paul Gastin, Marc Zeitoun, Thomas Place, and to my father. There were of course many others later on who reinforced my interests.

Finally, I would like to thank my friends and family, for their constant support and confidence in my abilities. And, of course, Jérémy, without whom nothing would be the same.

Contents

Chapter 1

Introduction

One of the most fundamental characteristics of a logic or a computational system is its expressive power. It can be studied for a number of reasons: the goal might simply be to better understand the limits of a logic used to write program specifications or database queries, or, for instance, to relate two distinct formalisms through equivalence results of the form "X has the same expressive power as Y". This may allow to transfer techniques from one to the other, and gain new insights on a problem; a good example would be descriptive complexity, or the application of automata theory to prove decidability results in logic.

Expressivity is a particularly important question in formal verification. Broadly speaking, the aim of verification is to ensure that hardware and software systems behave as expected (essentially, to prevent bugs), by providing mathematical guarantees of correctness. Here we focus on the automata-theoretic approach. In this setting, systems are modeled as variants of automata, and their specifications are formalized e.g. as logical formulas. There are two central questions: *model-checking* and *synthesis*. Model-checking algorithms take as input a system and its specification, and test whether the system satisfies the specification. Synthesis consists in constructing automatically a correct system from a given specification.

Take as a basic example the case of sequential finite-state systems modeled as finite automata, and of specifications written in linear temporal logic (LTL). Possible executions are represented as words over a finite alphabet. An automaton $\mathcal{A}$ or a formula φ both define a language of words: $L(\mathcal{A})$ is the set of all possible behaviors of $\mathcal{A}$, while $L(\varphi)$ is the set of all models of φ. The model-checking problem asks, given an automaton $\mathcal{A}$ and a formula φ, if all executions of $\mathcal{A}$ are models of φ, that is, if $L(\mathcal{A}) \subseteq L(\varphi)$. On the other hand, the synthesis problem asks, given a formula φ, to construct an automaton $\mathcal{A}$ such that $L(\mathcal{A}) = L(\varphi)$.[1] Both problems can be solved

[1] Note that our definition differs from the classical problem of reactive synthesis, in which the system has to respond to external actions from the environment [17, 69]. Here we will consider only closed systems.

with a polynomial space complexity [78, 91]. In fact, the translation from LTL to finite automata answering the synthesis problem is also central to model-checking. Indeed, the model-checking problem can be reformulated as follows: given $\mathcal{A}$ and φ, is it the case that $L(\mathcal{A}) \cap L(\neg\varphi) = \emptyset$? After constructing an automaton equivalent to $\neg\varphi$ and taking its product with $\mathcal{A}$, this comes down to a simple emptiness problem for finite automata. The core of the decision procedure for synthesis and model-checking is thus the fact that LTL is (effectively) no more expressive than finite automata: every finite automaton can be translated into an equivalent LTL formula.

Formal verification has been applied to more and more complex systems, leading to a wide variety of system models. They may involve various forms of concurrency, recursion, unbounded data, time, probabilities, etc. Accordingly, their executions are often represented by richer structures as well, as opposed to the previous example of words. Specification languages are also extremely varied; classic examples include first-order and monadic second-order logic, temporal logics such as LTL (linear temporal logic) [68] and CTL (computation tree logic) [18], PDL (propositional dynamic logic) [28], or the modal μ-calculus [55]. The complexity and even decidability of the model-checking and synthesis problems depend heavily on the system models and specification languages used. Usually, and especially for the system models, a balance has to be found between expressivity and complexity or decidability. On the other hand, two specification languages may have the same expressive power but widely different complexities. For instance, over finite or infinite words, LTL is as expressive as first-order logic [54], but its satisfiability problem is only in PSPACE [78], while it is non-elementary for first-order logic [80]. While the expressive power of classic specification languages is well understood over words, the situation is not always so clear for other classes of structures, associated with more complex systems.

1.1 Motivations

The aim of the present thesis is to study the expressive power of first-order logic and other formalisms over different classes of ordered structures, with a special focus on message-passing systems. Our motivations stem from the following classical problems.

Connections between logic and automata. The relation between logic and automata has been studied since the early 60's, starting with the work of Büchi [15], Elgot [27], and Trakhtenbrot [90], who established the equivalence in expressive power of finite automata and monadic second-order logic (MSO) over finite words: every MSO formula can be translated into an automaton which accepts precisely the models of the formula, and conversely, every automaton can be translated into

an MSO formula defining the same language. This was then extended to infinite words [16] and finite and infinite trees [84, 25, 71].

These early results were used to prove the decidability of MSO over words and trees: thanks to the translation from MSO to automata, the satisfiability of an MSO formula reduces to the non-emptiness of the corresponding automaton. The emptiness problem is decidable both for word and tree automata, making satisfiability (and validity) decidable for MSO. The equivalence of MSO and tree automata later found many other applications in logic.

As mentioned before, translations from logic to automata also play a central role in automated verification and synthesis. While the translation from MSO to finite-state automata is non-elementary [80], procedures with more reasonable complexities have been developed for logics such as LTL [91]. Even for MSO, the MONA tool has proven that despite the high worst-case complexity, implementations can be efficient in many practical examples [43].

Following the Büchi-Elgot-Trakhtenbrot theorem, logical characterizations of the expressive power of automata models have been established for various classes of structures beyond words and trees. For instance, automata are still expressively equivalent to MSO over nested words [1] or Mazurkiewicz traces [96, 87]. As in the case of words, the proof of these results relies on an inductive translation from MSO to automata: the operations of disjunction, conjunction, negation, and existential quantification at the level of logical formulas correspond respectively to the operations of union, intersection, complementation, and projection at the automata level. In these inductive translations, the main difficulty is often to prove the effective closure under complementation of the automata model. There are other classes of automata for which this property fails, and MSO may be too expressive. In that case, a logical characterization can sometimes still be obtained in terms of fragments of MSO. For instance, Thomas's graph acceptors, a generic automata model for bounded-degree graphs, are equivalent to existential monadic second-order logic (EMSO) [88, 89]. As a second example, data automata, accepting languages of data words, are expressively equivalent to the existential fragment of MSO with two first-order variables (EMSO^2) [6].

In the previous examples, the expressive power of automata is defined in terms of the language of *behaviors* they accept. Besides numerous other results of this kind, models of automata for distributed systems have also been studied as acceptors of (graph) architectures, and several logical characterizations in terms of MSO and modal logics have been obtained in that setting [41, 57, 73, 74].

Expressive completeness of temporal logics. Choosing the right specification language is a key step in verifying or synthesizing a system model. Formalizing intuitive requirements in a given logic is an error-prone process, so choosing one in which writing and understanding specifications is as simple as possible is paramount.

In addition, the complexities of synthesis or model-checking algorithms are largely dependent on the logic in which the specifications are written. Usually, decision problems for logics such as monadic second-order logic (MSO) or first-order logic (FO) are non-elementary (or even undecidable, depending on the class of systems and behaviors considered), while temporal or modal logics may result in more efficient algorithms. On the other hand, one should make sure that the chosen specification language is expressive enough for the kind of properties they want to consider. One way to argue that this is the case is through comparisons to the expressive power of other classic formalisms.

Kamp started a long trend of measuring the expressive power of temporal logics in terms of first-order logic, by proving that, over Dedekind-complete linear orders, LTL is expressively equivalent to first-order logic. Similar results have been established for other classes of structures, such as arbitrary linear orders [32], trees [77, 64], Mazurkiewicz traces [86, 23], or the real line equipped with predicates $+q$ for all rationals q [49]. In fact, there are at least two questions to consider:

(a) the expressive completeness of a particular temporal logic $\mathcal{L}$, that is, can every formula $\varphi(x)$ in FO be translated into an equivalent formula of $\mathcal{L}$?

(b) whether there *exists* an expressively complete temporal logic, over a given class of structures $\mathcal{C}$ (see below).

While question (a) may be concerned with any "natural" temporal logic, the second problem should be made more precise. A temporal logic consists of a set of connectives, their arities, and their semantics. Formulas are simply terms formed from atomic propositions and these connectives. For instance, the connectives of LTL are the boolean connectives, and the temporal connectives Until and Since (from which all other modalities can be defined). Following Gabbay [30], the problem of the existence of an expressively complete temporal logic is usually concerned with logics given by a *finite* set of *FO-definable* connectives (meaning that the semantics of each modality should be given as an FO formula). Even then, there are several variants of this problem, depending on whether only one-dimensional connectives are allowed, or possibly multi-dimensional ones. Intuitively, a one-dimensional temporal logic such as LTL considers a single reference point, while k-dimensional ones involve tuples of k reference points [30, 31].

It should be noted that, besides the positive examples given above, there are also various instances where the answer to these two questions is negative [30, 46, 45, 48].

The k-variable property. Logics with a bounded number of variables have been extensively studied. As observed by Gabbay [30], many temporal logics can be embedded into a fragment FO^k of FO, consisting of all first-order formulas involving at most k variables. Bounded variable logics also play an important role in descriptive complexity (see e.g. [38, 51]); for instance, Immerman's early characterizations of

the classes PSPACE and PTIME relate the space and time required by a Turing machine to decide a property to the number of variables and size of formulas in a uniform sequence of first-order formulas defining the same property [50].

One question has gathered a lot of interest [30, 70, 52, 20, 75, 3]: over a fixed class $\mathcal{C}$ of structures, is there $k \in \mathbb{N}$ such that all first-order properties can be expressed with at most k variables? If this is the case, $\mathcal{C}$ is said to have the k-variable property.[2]

Such a k does not always exist: if $\mathcal{C}$ is the class of all possible structures (over a given signature), then it does not have the k variable property for any k. For instance, the property "there are at least k distinct elements" can be expressed in FO^k, but not in FO^{k-1}. A classic example of a class which does have the k-variable property is the class of all linear orders (for $k = 3$) [52]. In the presence of a linear order, the previous property ("there are at least k distinct elements") can easily be expressed with 2 variables, for instance, for $k = 4$:

$$\exists x.\exists y.\big(x < y \wedge \exists x.(y < x \wedge \exists y.x < y)\big)\,.$$

Note that both x and y are quantified twice in the formula, denoting different elements each time. This ability to reuse variables and nest quantifications means that FO^k can express many properties whose "natural" definition in FO would use more variables.

Gabbay established tight connections between the k-variable property and the expressive completeness of temporal logics [30]. More precisely, a class $\mathcal{C}$ of time flows admits an expressively complete temporal logic, with a finite set of FO-definable multi-dimensional temporal connectives, if, and only if, there exists k such that every FO sentence is equivalent to one in FO^k. Gabbay left open whether this was still true for *one*-dimensional temporal logics, but Hodkinson later proved that this is not the case [46]. The equivalence also fails if one asks that every formula *with up to k free variables* (rather than sentences only) be equivalent to one in FO^k [47].

Another classic approach to proving or disproving that a class of structures has the k-variable property is through Ehrenfeucht-Fraïssé games with a bounded number of pebbles. Positive results were established in this way for linear orders and bounded-degree trees [52], and for $(\mathbb{R}, <, +1)$ [3]. Ehrenfeucht-Fraïssé games are used to prove that a class does *not* have the k-variable property for any k in [93].

1.2 Ordered structures and message-passing systems

The problems described in Section 1.1 can be instantiated to various classes of structures. In this thesis, we focus on classes of ordered structures with only

[2] It should be noted that there are, in fact, several non-equivalent definitions of this notion in the literature (see [47] for an overview). It is also related to Henkin dimension.

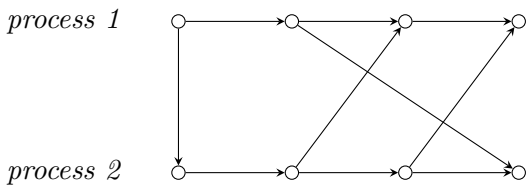


Figure 1.1: A simple MSC

unary and binary predicates, satisfying some monotonicity conditions. Chapters 4-6 are concerned with one particular class, message sequence charts (MSCs), which correspond to executions of message-passing systems. We first discuss the latter.

Communicating finite-state machines and message sequence charts. Communicating finite-state machines (CFMs) are a classic model of message-passing systems [14]. A CFM consists of a finite set of processes, described by finite-state automata, that communicate by exchanging messages through unbounded point-to-point FIFO channels. Behaviors of CFMs can be represented in a visual manner by message sequence charts (MSCs), as in Figure 1.1. Each process is associated with a sequence of *events* in the MSC, corresponding to internal actions, sending and receiving of messages. The causal dependencies between events are indicated by two binary relations connecting (i) the emission of a message with its reception, and (ii) successive events executed by one and the same process. In addition, events are partially ordered by Lamport's happened-before relation: an event e happens before an event f if, and only if, there is a "message flow" path from e to f [58].

CFMs are not as well-behaved as finite-state (word) automata. Their emptiness problem is undecidable, as message queues can be used to simulate a Turing machine. In addition, the class of MSC languages accepted by CFMs is not closed under complementation, making them strictly less expressive than MSO [13]. This led to the investigation of decidable restrictions of CFMs, such as assuming a bound on the channel capacity. This amounts to restricting the set of possible behaviors to *universally bounded* or *existentially bounded* MSCs (in which all, respectively at least one, scheduling of events ensures that the number of pending messages in each channel remains bounded at all times).

The expressive power of CFMs over *bounded* MSCs is well-understood: they are equivalent to MSO, both in the case of existentially and universally bounded MSCs, and for finite as well as infinite MSCs [44, 56, 34, 35]. These results relied on the discovery of strong connections between bounded MSCs (and CFMs) and Mazurkiewicz traces (and asynchronous automata). This allowed a transfer of certain techniques and results from trace theory to bounded MSCs.

The expressive power of *unbounded* CFMs was investigated in [13, 11]. When

the happened-before relation is dropped from the signature (leaving only the direct process successor and message relations), CFMs are equivalent to EMSO, the existential fragment of MSO. The core of the proof is a translation from FO to CFMs, which relies on the locality of FO over structures of bounded degree (the techniques used would therefore not go through with the happened-before relation, of unbounded degree). The result can then be lifted from FO to EMSO using the closure under projection of CFMs. The converse translation, from CFMs to EMSO, is similar to the case of words.

While this gives a nice characterization of the expressive power of CFMs, it is not always very practical to write specifications using only the direct process successor and message relations. For instance, the mutual exclusion property is much easier to define in terms of the happened-before relation $\leq$. Two events e and f are *concurrent* when they are not ordered by the happened-before relation: $e \not\leq f$ and $f \not\leq e$. The mutual exclusion property asks that there are no two concurrent events which are both in the critical section. Assuming that a unary predicate CS denotes "being in the critical section", the mutual exclusion property can be defined in FO as:

$$\neg\Big(\exists x.\exists y.\, CS(x) \wedge CS(y) \wedge \neg(x \leq y) \wedge \neg(y \leq x)\Big)\,.$$

While we will see that our results imply that there is an equivalent EMSO formula which does not use $\leq$, coming up with such a formula is much harder.

This raises the question of whether the synthesis of CFMs from FO specifications is still possible when we get rid of both restrictions (the bounded channels of [44, 56, 34, 35], and the restricted set of predicates of [13, 11]):

Question 1

Can every FO formula over MSCs be translated into an equivalent CFM, even when the happened-before relation is included in the signature and the channels are unbounded?

We give a positive answer to Question 1 in Chapter 6. As in the case of words, the translation from FO to MSCs is non-elementary [80], leaving open the question of efficient translations for other logics. While several temporal logics over MSCs have been considered [67, 65, 12, 66], there is no canonical one, and their precise expressive power relative to FO and MSO is not very well understood.

Question 2

Is there any expressively complete temporal logic for MSCs?

While the question remains open for one-dimensional temporal logics, we define a logic which is expressively equivalent to FO, and which can be considered as a two-dimensional temporal logic (more details in Section 1.3).

As discussed in Section 1.1, questions related to the expressive completeness of temporal logics are closely related to the k-variable property, but not always quite equivalent, depending on the exact definitions. Thus, it makes sense to ask separately:

Question 3

Do MSCs have the k-variable property for some k?

We prove that MSCs indeed have the 3-variable property, in a strong sense: every FO formula $\varphi(x_1, \ldots, x_n)$ is equivalent, over MSCs, to a boolean combination of FO^3 formulas. Note that this applies to formulas with arbitrarily many free variables $x_1, \ldots, x_n$. Each FO^3 component in the boolean combination may have a different subset of at most three free variables from $\{x_1, \ldots, x_n\}$, and at most three (bound or free) variables in total.

Interval-preserving structures. Our answer to Question 3 is in fact a special case of a more general result.

Broadly speaking, we are concerned with classes of ordered structures, that is, structures in which a special predicate $\leq$ is interpreted as a partial order. Assuming an underlying order in the structure is a natural assumption when considering models of executions of computer systems. In the case of sequential systems, events in the execution (points in the structure) are linearly ordered according to the progression of the computation. The underlying order may for instance be any finite or infinite interval of $\mathbb{Z}$ (for discrete systems), or of $\mathbb{Q}$ or $\mathbb{R}$ (for real-time systems). In the case of distributed systems, where events may occur in parallel, executions are naturally partially ordered, as in the example of MSCs, rather than linearly ordered.

Still, we focus on classes of *linearly* ordered structures. This is not as restrictive as it may first appear. Even in partially ordered executions of distributed systems, the set of events associated with a single process is still linearly ordered by the *process order*. From a purely technical point of view, in the case of MSCs (and some other structures), the process order essentially plays the same role as would a linear order, even though it is rather a *union* of linear orders.[3] In addition, over MSCs, FO specifications written using the happened-before relation and the

[3] This comes from the fact that MSCs only involves a fixed, finite number of processes, which we can assume to be ordered.

message relation can easily be rewritten with only the process order and the message relation, and vice versa.

From a more practical point of view, linearly ordered structures are a natural starting point to study the k-variable property. One of the most basic examples of classes with the k-variable property is the class of all linear orders (with monadic predicates, but no other binary relation besides the linear order), which has the 3-variable property. However, the situation becomes more complicated as soon as other binary predicates are added to the signature. Venema proved that the class of all dense linear orders, with a second binary relation, does not have the k-variable property for any k [93]. For a long time, the question remained open for *finite* linear orders (see [20] for a survey), before Rossman proved that finite ordered graphs do not have the k-variable property for any k [75]. These examples from Venema and Rossman show that, while the class of all linear orders has the 3-variable property, adding a single binary relation suffices to obtain classes of structures which do not have the k-variable property for any k. On the positive side, Antonopoulos et al. proved that the real line equipped with a binary relation $+1$ and arbitrary monadic predicates has the 3-variable property [3]. They also showed that this is still true when the relation $+1$ is replaced with any linear function $f : x \mapsto ax + b$. The conclusion of [3] raised the question of whether this also holds for arbitrary polynomial functions, or for other linear orders and classes of monotone functions, besides $(\mathbb{R}, <)$. More generally, one may ask:

Question 4

What are some sufficient conditions for a class of linearly ordered structures (with unary and binary relations) to have the k-variable property?

It turns out that *monotonicity* is a possible answer. We prove that any class of ordered structures equipped with non-increasing or non-decreasing functions (seen as binary predicates), and arbitrary unary predicates, has the k-variable property. This applies to $(\mathbb{R}, <, +1)$, generalizing the result from [3], but also to MSCs (seen as linearly ordered by the process order). Indeed, the assumption that the channels are FIFO implies that the function mapping send events to matching receive events, for any given channel, is monotone. Other examples of "indirect" applications of this result are Mazurkiewicz traces (for which the equivalence of FO and FO^3 was already known [23]), and $(\mathbb{R}, <)$ with polynomial functions.

In fact, we define a slightly more general monotonicity condition, for arbitrary binary relations rather than only functional ones, which we call *interval-preserving*. We prove that any class of linearly ordered structures with unary and binary relations in which every binary relation is interval-preserving has the 3-variable property.

1.3 Star-free propositional dynamic logic

While several of our contributions can be stated purely in terms of first-order logic and automata, another formalism plays a central role in all the results of the thesis: star-free propositional dynamic logic (star-free PDL, or $\mathrm{PDL}_{\mathsf{sf}}$). It is both a convenient tool, serving as an intermediate language in translations from one formalism to another, and a logic of independent interest. It combines features from several classic formalisms. One of them is, of course, propositional dynamic logic (PDL), but star-free PDL also presents similarities with star-free regular expressions, or with relation algebras.

Propositional dynamic logic was introduced by Fischer and Ladner [28] to reason about program schemes, and has now found a large range of applications in artificial intelligence and verification [40, 21, 60, 59, 37]. It combines event formulas which are evaluated at a "current element" of the model, and path formulas defining binary relations between points, which are used to navigate in the structure. Event formulas are constructed using boolean and modal operators, while path formulas are based on the operations of concatenation, union and Kleene star. Several extensions have been studied, including PDL with converse [81], intersection [19], or negation of atomic programs [63].

Star-free PDL is a variant of PDL with converse in which the Kleene star of path formulas is removed, and replaced with a complement operation (analogously to star-free regular expressions). Star-free PDL thus uses a classic set of operations on binary relations: union, intersection, complement, concatenation and converse. These are in particular the operations of relation algebras [83]. Connections between relation algebras, modal logics and interval temporal logics are discussed in [92].

As was already known for similar logics (see e.g. [83, 94]), in general, star-free PDL is expressively equivalent to FO^3 (and less that FO). We investigate more precisely the expressive power of star-free PDL over the classes of structures discussed in the previous section: interval-preserving structures (that is, linearly ordered structures in which all binary relations are interval-preserving), and MSCs.

Sufficient conditions for the 3-variable property. One of the main result of the thesis is that, over interval-preserving structures, star-free PDL is as expressive as full first-order logic. Since star-free PDL is equivalent to FO^3, this proves that interval-preserving structures have the 3-variable property, giving an answer to Question 4. The translations from FO to $\mathrm{PDL}_{\mathsf{sf}}$ and from $\mathrm{PDL}_{\mathsf{sf}}$ to FO^3 are effective.

Specification and synthesis of message-passing systems. This first result also implies that, over MSCs, the logics $\mathrm{PDL}_{\mathsf{sf}}$, FO^3 and FO have the same expressive power. Furthermore, we define an expressively complete fragment $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ of $\mathrm{PDL}_{\mathsf{sf}}$ where the complement operator is replaced with path operations similar to the *until*

and *since* modalities of LTL, evaluated on the current process. It could in fact also be defined as a fragment of regular PDL with converse and loop (or intersection).

The logic $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ constitutes a nice specification language for message-passing systems. It can be seen as a simple extension of LTL to MSCs, except for the presence of a Loop construct which allows to come back to the starting evaluation point. Intuitively, in LTL, when evaluating a formula such as $\varphi_1 \mathsf{U} (\varphi_2 \mathsf{S} \varphi_3)$ ("φ_1 holds until a position from which φ_2 has been true since φ_3"), we forget about the initial position as soon as we start evaluating the second half of the formula. More precisely, $\varphi_1 \mathsf{U} (\varphi_2 \mathsf{S} \varphi_3)$ is true at a position e if there exists $f > e$ and $g < f$ such that φ_1 holds between e and f, φ_2 holds between f and g, and φ_3 holds at g; there is no constraint on the position of g relative to e. In $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$, the Loop construct would allow us to require that $e = g$. In fact, $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ can be seen as a two-dimensional temporal logic, giving a partial answer to Question 2 (it is still open whether there exists an equivalent one-dimensional temporal logic). It is also sufficiently succinct and intuitive to express other classic temporal connectives over partial orders with formulas of reasonable size. More importantly, while as expressive as FO and $\mathrm{PDL}_{\mathsf{sf}}$, the logic $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ is an easier starting point – conceptually as well as complexity-wise – for the problem of translating logical specifications into CFMs. This comes from the fact that it does not use the complement operation on path formulas. Intuitively, the main difficulty when translating first-order formulas into automata is negation: over words, each negation results in an exponential blow-up from taking the complement of the automaton; over MSCs, the situation is even more complicated because CFMs are not closed under complement. The complement of path formulas in star-free PDL (or, over words, the complement in star-free regular expressions) raises similar issues. On the other hand, by removing complements of path formulas, the logic $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ becomes closer to LTL, in that negation is only allowed at the level of unary predicates (with, however, the remaining difficulty of Loop formulas).

One of our main results is that every $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ formula can be translated into an equivalent CFM of exponential size. This is orthogonal to what was previously known regarding translations from variants of PDL to CFMs. It was shown in [12] that any one-way PDL formula, where the Kleene star is allowed but we can move only in one direction (future or past), can be translated into an equivalent CFM. However, $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ and PDL have incomparable expressive powers. On the other hand, $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ can be defined in the extension of PDL with converse and intersection, but the latter contains formulas which are not equivalent to any CFM [12].

Combining the translations from FO to $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$, and from $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ to CFMs, we obtain a positive answer to Question 1. This also has interesting consequences for the case of existentially bounded MSCs: using the translation from FO to CFMs over unrestricted MSCs, we give a new proof of the equivalence of the full MSO logic and CFMs over existentially bounded MSCs, which was shown in [34] for

finite MSCs. We also extend the result to infinite MSCs.

1.4 Outline

Chapter 2 consists of general preliminaries on logic. We introduce first-order logic and monadic second-order logic, and give a more precise presentation of the k-variable property. We also define propositional dynamic logic (PDL) and its variants with intersection and converse.

Chapter 3 introduces star-free PDL and our results concerning interval-preserving structures. The main result of the chapter is the equivalence, over interval-preserving structures, of star-free PDL, FO, and FO^3.

We first define star-free PDL, and give a proof of its equivalence to FO^3 over arbitrary structures with unary and binary predicates. We then introduce interval-preserving relations and structures, and a fragment $\mathrm{PDL}_{\mathsf{sf}}^{\mathsf{int}}$ of star-free PDL in which all path formulas define interval-preserving relations, provided all atomic binary predicates are themselves interpreted as interval-preserving relations. We then prove that over interval-preserving structures, $\mathrm{PDL}_{\mathsf{sf}}^{\mathsf{int}}$, $\mathrm{PDL}_{\mathsf{sf}}$, FO^3 and FO are all expressively equivalent. We also discuss some examples of structures where this result applies, either directly or indirectly. Finally, we prove that over complete linear orders, the complement operator can be replaced by path operators similar to the *until* and *since* modalities of LTL.

Chapter 4 provides preliminaries on message sequence charts (MSCs) and communicating finite-state machines (CFMs).

We first define MSCs (as well as existentially and universally-bounded MSCs), and discuss possible signatures for FO and MSO over MSCs. We then introduce communicating finite-state machines, and give a brief account of known results concerning logical characterizations of CFMs, over bounded or unbounded MSCs.

Chapter 5 discusses variants of star-free PDL and temporal logics for MSCs, and proves a critical result towards the synthesis of CFMs: the removal of Loop in $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ formulas.

We first explain how results from Chapter 3 can be applied to MSCs. We then define $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ and show that it is as expressive as full $\mathrm{PDL}_{\mathsf{sf}}$ or FO over MSCs. We then prove that every $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ sentence is equivalent, up to projection, to one which does not use Loop. Finally, we show how to define various classic temporal modalities over MSCs, based on the happened-before relation rather than the process order, in $\mathrm{PDL}_{\mathsf{sf}}$.

Chapter 6 considers the problem of synthesizing CFMs from logical specifications. The main result of the chapter is the equivalence of CFMs and EMSO with the happened-before relation.

We first give a translation from $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ formulas constructed without Loop into CFMs, very similar to usual translations from LTL to Büchi automata. Combining this with the results from Chapter 5, we obtain translations from $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ and FO to CFMs. As a corollary, CFMs are exactly as expressive as EMSO. Our translation produces an automaton of exponential size in the case of $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$, and non-elementary for FO. We also apply this to the temporal logic defined in Chapter 5, and obtain in that case a CFM of size exponential in the size of the formula, and doubly-exponential in the number of processes. Finally, we show that our results can be used to give a new proof of the fact that, over existentially bounded MSCs, CFMs have the same expressive power as MSO, and as regular specifications over (bounded) linearizations [34]. We also extend this result from finite to infinite MSCs.

Most of the results presented in the thesis appear in [9, 29, 10].

Chapter 2

Logical background

This chapter introduces the two main logics studied in the thesis: Monadic Second-Order Logic (MSO), and Propositional Dynamic Logic (PDL). In later chapters, we will consider several variants or restrictions of these logics, first in a general setting and then in the special case of message-passing systems. For now, we simply introduce basic notations and definitions. We also present one of the main questions studied in the thesis: the *k-variable property* (Section 2.3).

A summary of logics introduced throughout the thesis is given on pages 153-154.

2.1 Models

All the logics we consider are interpreted over relational structures with only unary and binary relations. These structures may represent, e.g., behaviors of message-passing systems or real-time systems.

Signature

Accordingly, we let $\Sigma = (\mathsf{Prop}, \mathsf{Rel})$ denote a *signature* consisting of a set of unary relation symbols Prop, and a set of binary relation symbols Rel. We make no assumption on the cardinality of Prop and Rel. We use letters $P, Q, \ldots$ to denote elements of Prop, and $\alpha, \beta, \ldots$ to denote elements of Rel.

Σ-structure

A Σ-*structure* is a tuple $M = (E^M, (P^M)_{P \in \mathsf{Prop}}, (\alpha^M)_{\alpha \in \mathsf{Rel}})$, where E^M is the domain of M, $P^M \subseteq E^M$ is the interpretation of P, and $\alpha^M \subseteq E^M \times E^M$ is the interpretation of α. We refer to elements of E^M as *events*.

Example 2.1. The simplest class of structures we will consider are *words*. A word over a finite alphabet A is a particular Σ-structure over the signature $\Sigma = (A, \{<\})$. The domain of the Σ-structure consists of positions in the word, linearly ordered according to the interpretation of $<$. For each $a \in A$, the interpretation of the unary predicate a indicates which positions are occurrences of the letter a. For instance, for $A = \{a, b\}$, the word *abbaab* denotes the Σ-structure

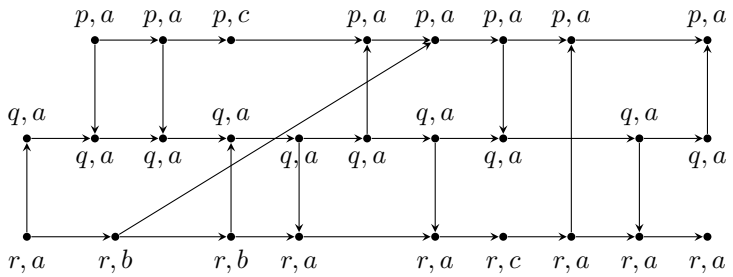


Figure 2.1: An MSC

$$w = (E^w = \{0, \ldots, 5\}, a^w = \{0, 3, 4\}, b^w = \{1, 2, 5\}, <^w = \{(i, j) \mid 0 \leq i < j \leq 5\}) .$$

Of course, not all $(A, \{<\})$-structures are words. The set of all words over A can be defined as Σ-structures w such that (i) for all $e \in E^w$, there exists a unique $a \in A$ such that $e \in a^w$, and (ii) $E^w = \mathbb{N}$ or $E^w = \{0, \ldots, n-1\}$ for some $n \in \mathbb{N}$, and $<^w$ is the usual ordering of the natural numbers.

Alternatively, a word over A can be seen as a Σ'-structure over the signature $\Sigma' = (A, \{\rightarrow\})$, where $\rightarrow$ is interpreted as the direct successor relation.

Example 2.2. Figure 2.1 shows an example Σ-structure M over the signature $\Sigma = (\{a, b, c, p, q, r\}, \{\rightarrow, \lhd\})$. Nodes correspond to elements of the domain E^M, and labels indicate the interpretation of the unary predicates. Horizontal edges are in the interpretation of $\rightarrow$, and vertical or oblique edges in the interpretation of $\lhd$.

This Σ-structure is a *message sequence chart* (MSC). Intuitively, MSCs represent executions of message-passing systems; they will be defined precisely in Chapter 4. The relation $\rightarrow$ corresponds to a process successor relation, i.e., connects events which are executed successively by a same process, while the relation $\lhd$ connects matching sends and receives. Here, there are three processes (thus three "horizontal lines"), p, q and r. The interpretation of the corresponding unary predicates identifies which events are performed by each process: notice that each event has exactly one label among $\{p, q, r\}$. The other unary predicates (here, $\{a, b, c\}$) may denote e.g. an action performed by the process, or a property of its current state.

Notations for binary relations. Let $\mathcal{R} \subseteq E \times E$ be a binary relation over a set E. We sometimes write $e \mathcal{R} f$ if $(e, f) \in \mathcal{R}$. Given $e \in E$ and $F \subseteq E$, we let $\mathcal{R}(e) = \{f \in E \mid e \mathcal{R} f\}$ and $\mathcal{R}(F) = \bigcup_{e \in F} \mathcal{R}(e)$. We define the *converse* of a relation $\mathcal{R}$ as $\mathcal{R}^{-1} = \{(f, e) \in E \times E \mid (e, f) \in \mathcal{R}\}$, and the *composition* of two binary relations $\mathcal{R}_1 \subseteq E \times E$ and $\mathcal{R}_2 \subseteq E \times E$ as

$$\mathcal{R}_1 \cdot \mathcal{R}_2 = \{(e, g) \in E \times E \mid \exists f \in E.\ (e, f) \in \mathcal{R}_1 \wedge (f, g) \in \mathcal{R}_2\} .$$

The *n-th iterate* $\mathcal{R}^n$ of a relation $\mathcal{R}$ is defined inductively by

$$\mathcal{R}^0 = \{(e,e) \mid e \in E\} \qquad \text{and} \qquad \mathcal{R}^{n+1} = \mathcal{R} \cdot \mathcal{R}^n \,,$$

and the *reflexive transitive closure* of $\mathcal{R}$ by

$$\mathcal{R}^* = \bigcup_{n \in \mathbb{N}} \mathcal{R}^n \,.$$

Finally, we write $\mathcal{R}^{\mathsf{c}} = (E \times E) \setminus \mathcal{R}$ for the *complement* of $\mathcal{R}$.

2.2 Monadic Second-Order Logic

We first recall the syntax and semantics of *monadic second-order logic* (MSO).

We assume an infinite supply of *first-order variables* $x, y, z, \ldots$ ranging over events in a structure, and *second-order variables* $X, Y, Z, \ldots$ ranging over sets of events. Given a signature $\Sigma = (\mathsf{Prop}, \mathsf{Rel})$, the set MSO[$\Sigma$] of MSO formulas over Σ is defined as follows:

Monadic second-order logic (MSO)

$$\Phi ::= P(x) \mid \alpha(x,y) \mid x = y \mid x \in X \mid \Phi \vee \Phi \mid \neg\Phi \mid \exists x.\Phi \mid \exists X.\Phi \,,$$

where $P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel}$, x and y are first-order variables, and X is a second-order variable. We use the usual abbreviations to also include implication $\implies$, conjunction $\wedge$, and universal quantification $\forall$.

We denote by Free(Φ) the set of free (first and second-order) variables of a formula Φ. We sometimes write $\Phi(x_1, \ldots, x_m, X_1, \ldots, X_n)$ to indicate that the free variables of Φ are among $x_1, \ldots, x_m, X_1, \ldots, X_n$ (but do not necessarily include all of them). A *sentence* is a formula without any free variable.

Let $\Phi \in$ MSO[Σ], M a Σ-structure, and $V \supseteq$ Free(Φ) a set of variables containing at least all free variables of Φ. An *interpretation* of V in M is a function $\nu : V \to E^M \cup 2^{E^M}$ which maps each first-order variable $x \in V$ to an element $\nu(x) \in E^M$, and each second-order variable $X \in V$ to a subset $\nu(X) \subseteq E^M$. We sometimes use the notation, e.g., $[x \mapsto e, y \mapsto f, X \mapsto F]$ to denote the interpretation of $\{x, y, X\}$ which maps x to e, y to f, and X to F. Given an interpretation ν of V, a first-order variable x (which may or may not be in V), and $e \in E^M$, we also write $\nu[x \mapsto e]$ for the interpretation of $V \cup \{x\}$ which maps x to e and all variables $y \in V \setminus \{x\}$ to $\nu(y)$. We define similarly $\nu[X \mapsto F]$.

We write $M, \nu \models \Phi$ if M satisfies Φ when the free variables of Φ are interpreted

according to the interpretation ν:

$$
\begin{array}{lll}
M, \nu \models P(x) & \text{if} & \nu(x) \in P^M \\
M, \nu \models \alpha(x, y) & \text{if} & (\nu(x), \nu(y)) \in \alpha^M \\
M, \nu \models x = y & \text{if} & \nu(x) = \nu(y) \\
M, \nu \models x \in X & \text{if} & \nu(x) \in \nu(X) \\
M, \nu \models \Phi \vee \Psi & \text{if} & M, \nu \models \Phi \text{ or } M, \nu \models \Psi \\
M, \nu \models \neg\Phi & \text{if} & M, \nu \not\models \Phi \\
M, \nu \models \exists x.\Phi & \text{if} & \text{there exists } e \in E^M \text{ such that } M, \nu[x \mapsto e] \models \Phi \\
M, \nu \models \exists X.\Phi & \text{if} & \text{there exists } F \subseteq E^M \text{ such that } M, \nu[X \mapsto F] \models \Phi\,.
\end{array}
$$

Logical equivalence

We say that two formulas $\Phi, \Psi \in \mathrm{MSO}[\Sigma]$ are *equivalent* over a class of Σ-structures $\mathcal{C}$, written $\Phi \equiv_{\mathcal{C}} \Psi$, if for all $M \in \mathcal{C}$, for all sets of variables $V \supseteq \mathrm{Free}(\Phi) \cup \mathrm{Free}(\Psi)$ and interpretation $\nu : V \to E^M \sqcup 2^{E^M}$, we have $M, \nu \models \Phi$ if and only if $M, \nu \models \Psi$. We say that Φ and Ψ are *equivalent*, written $\Phi \equiv \Psi$, if they are equivalent over the class of all Σ-structures.

FO, monadic vs. non-monadic

The set of *first-order* (or *monadic first-order*) formulas over a signature Σ, denoted $\mathrm{FO}[\Sigma]$, is the set of formulas without second-order quantification $\exists X$. However, they may contain formulas $x \in X$. We say that a first-order formula is *non-monadic* if it contains no subformula $x \in X$.

EMSO

Existential monadic second-order logic over Σ ($\mathrm{EMSO}[\Sigma]$) is the fragment of $\mathrm{MSO}[\Sigma]$ consisting of formulas of the form $\exists X_1 \ldots \exists X_n.\Phi$, where $\Phi \in \mathrm{FO}[\Sigma]$.

2.3 Bounded-variable fragments

k-variable fragment ($\mathrm{FO}^k, \mathrm{EMSO}^k$)

Given $k \in \mathbb{N}$, we denote by $\mathrm{FO}^k[\Sigma]$ the set of $\mathrm{FO}[\Sigma]$ formulas which use at most k distinct first-order variables (and arbitrarily many second-order variables), and $\mathrm{EMSO}^k[\Sigma]$ the set of formulas of the form $\exists X_1 \ldots \exists X_n.\Phi$, where $\Phi \in \mathrm{FO}^k[\Sigma]$. Note that a same variable can be quantified and re-used several times. For instance, the formula $\exists x.\exists y.\exists z.\alpha(x, y) \wedge \alpha(y, z)$ is not syntactically in $\mathrm{FO}^2[\Sigma]$; however, it is equivalent to the $\mathrm{FO}^2[\Sigma]$ formula $\exists x.\exists y.\,(\alpha(x, y) \wedge \exists x.\alpha(y, x))$.

The bounded variable hierarchy is strict in general: for all k, there exists a sentence in $\mathrm{FO}^{k+1}[\Sigma]$ which is not equivalent to any formula in $\mathrm{FO}^k[\Sigma]$. A simple example of such a formula is

$$\exists x_1 \ldots \exists x_{k+1}. \bigwedge_{1 \leq i < j \leq k+1} \neg(x_i = x_j)\,,$$

which states that there are at least $k + 1$ distinct elements in the model.

However, there are certain classes of structures over which this hierarchy collapses. For instance, it is well-known that over finite or infinite words, or more

generally over linear orders, $\mathrm{FO}[\Sigma]$ and $\mathrm{FO}^3[\Sigma]$ have the same expressive power [54, 52]. As an example, the above formula ("at least $k+1$ distinct elements") is equivalent, for $k=3$, to:

$$\exists x.\exists y.\big(x < y \wedge \exists x.(y < x \wedge \exists y.x < y)\big)\,.$$

It is then natural to ask for which classes this is the case:

> *Given a class $\mathcal{C}$ of structures, is there a k such that all* FO*-definable properties over $\mathcal{C}$ can be expressed with at most k variables?*

In fact, there are several non-equivalent versions of this question, which have been studied for various classes [30, 70, 52, 20, 75, 3]. For instance, it is not equivalent to ask that all *sentences* in $\mathrm{FO}[\Sigma]$, or all formulas with up to k free variables, are equivalent to one in $\mathrm{FO}^k[\Sigma]$. There may also be variations depending on whether one considers monadic or non-monadic first-order logic (that is, whether second-order variables are allowed in the syntax). The differences between several of these properties, including connections to expressively complete temporal logics, are discussed in [47].

In this thesis, we will study what Hodkinson and Simon [47] call the *strong k-variable property*: we say that a class $\mathcal{C}$ of structures has the strong k-variable property if and only if every formula in $\mathrm{FO}[\Sigma]$ (with an arbitrary number of free variables) is equivalent to a finite boolean combination of formulas in $\mathrm{FO}^k[\Sigma]$. Hodkinson and Simon showed that for *monadic* first-order logic, this is in fact equivalent to the seemingly weaker *k-variable property*, which requires that all $\mathrm{FO}[\Sigma]$ formulas with at most k free first-order variables are equivalent to one in $\mathrm{FO}^k[\Sigma]$.

(strong) k-variable property

Example 2.3. Let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial function. Fix $\Sigma = (\emptyset, \{<, p\})$, and consider the Σ-structure $M = (\mathbb{R}, <, \{(x, p(x)) \mid x \in \mathbb{R}\})$, where $<$ is the usual ordering of the real numbers. Antonopoulos et al. [3] showed that $\{M\}$ has the 3-variable property if p is an affine function $x \mapsto ax+b$, and left open in the conclusion of [3] whether this is the case for an arbitrary polynomial p. We answer positively this question in Section 3.7.

Let us give some examples of $\mathrm{FO}^3[\Sigma]$ formulas over this structure. We can compare the value of p at two points using the formula

$$p(x) \leq p(y) \;:=\; \exists z.\, p(x,z) \wedge \exists x.\, (p(y,x) \wedge z \leq x)\,,$$

where $z \leq x$ is an abbreviation for $z = x \vee z < x$. We can then define formulas $\min(x) \in \mathrm{FO}^3[\Sigma]$ and $\max(x) \in \mathrm{FO}^3[\Sigma]$ which state that x is a local minimum (resp. maximum) of p, for instance:

$$\begin{aligned}\min(x) := {} & (\exists z.\, z < x \wedge \forall y.\, (z < y < x \implies p(x) \leq p(y))) \wedge \\ & (\exists z.\, x < z \wedge \forall y.\, (x < y < z \implies p(x) \leq p(y)))\,.\end{aligned}$$

Finally, if $m_1 < \cdots < m_n$ are the local extrema of p, we can also define a formula $m_i \leq x \in \mathrm{FO}^3[\Sigma]$ which states that there exist at least i local extrema before x, alternating quantifications over y and x to identify them. For instance, $m_2 \leq x$ is the formula

$$\exists y.\, y \leq x \wedge (\min(y) \vee \max(y)) \wedge \exists x.\, x < y \wedge (\min(x) \vee \max(x))\,.$$

2.4 Propositional Dynamic Logic

While MSO is a very natural and expressive logic, satisfiability of MSO formulas and other verification problems are undecidable in general, and non-elementary over structures such as words where decidability is recovered.

Propositional dynamic logic (PDL) is another classical logic which is quite expressive, though less than MSO, and has better algorithmic properties. Originally, PDL has been used to reason about program schemas and transition systems [28]. Since then, PDL and its extensions with intersection and converse [81] have developed a rich theory with applications in artificial intelligence and verification [40, 21, 60, 59, 37].

Syntax and semantics. PDL consists of two sorts of formulas: *event formulas* which are evaluated at events in a structure, and *path formulas* which are evaluated at pairs of events and allow us to navigate inside the structure. In addition, we can also define *sentences* to reason about global properties of the model. Given a signature $\Sigma = (\mathsf{Prop}, \mathsf{Rel})$, formulas in PDL[$\Sigma$] are defined as follows:

Propositional Dynamic Logic (PDL)

$$\begin{aligned} \xi &::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi && \text{(sentences)} \\ \varphi &::= P \mid \mathit{true} \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi && \text{(event formulas)} \\ \pi &::= \alpha \mid \{\varphi\}? \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^* && \text{(path formulas)} \end{aligned}$$

where $P \in \mathsf{Prop}$ and $\alpha \in \mathsf{Rel}$. We will also consider the extension of PDL with *intersection* and *converse* of path formulas: ICPDL[Σ] is the set of formulas

ICPDL

$$\begin{aligned} \xi &::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi && \text{(sentences)} \\ \varphi &::= P \mid \mathit{true} \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi && \text{(event formulas)} \\ \pi &::= \alpha \mid \{\varphi\}? \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^* \mid \pi \cap \pi \mid \pi^{-1} && \text{(path formulas)} \end{aligned}$$

Let $M = (E^M, (P^M)_{P\in\mathsf{Prop}}, (\alpha^M)_{\alpha\in\mathsf{Rel}})$ be a Σ-structure. The semantics $[\![\varphi]\!]^M$ of an event formula φ is a subset of E^M, and the semantics $[\![\pi]\!]^M$ of a path formula is a binary relation over E^M. Both are defined below. When M is clear from the context, we simply write $[\![\varphi]\!]$ and $[\![\pi]\!]$ instead of $[\![\varphi]\!]^M$ and $[\![\pi]\!]^M$. We also write $M, e \models \varphi$ when $e \in [\![\varphi]\!]^M$, and $M, e, f \models \pi$ when $(e, f) \in [\![\pi]\!]^M$.

For event formulas,

$$\begin{aligned}
[\![P]\!]^M &= P^M \\
[\![\mathit{true}]\!]^M &= E^M \\
[\![\varphi \vee \varphi']\!]^M &= [\![\varphi]\!]^M \cup [\![\varphi']\!]^M \\
[\![\neg\varphi]\!]^M &= E^M \setminus [\![\varphi]\!]^M \\
[\![\langle\pi\rangle\,\varphi]\!]^M &= \{e \in E^M \mid \exists f \in [\![\varphi]\!]^M.\ (e,f) \in [\![\pi]\!]^M\}\,.
\end{aligned}$$

Intuitively, $\langle\pi\rangle\,\varphi$ holds at e if there exists a path matching π which starts at e and ends in some f such that φ holds at f. The semantics of path formulas is defined as follows:

$$\begin{aligned}
[\![\alpha]\!]^M &= \alpha^M \\
[\![\{\varphi\}?]\!]^M &= \{(e,e) \mid e \in [\![\varphi]\!]^M\} \\
[\![\pi \cdot \pi']\!]^M &= [\![\pi]\!]^M \cdot [\![\pi']\!]^M \\
[\![\pi + \pi']\!]^M &= [\![\pi]\!]^M \cup [\![\pi']\!]^M \\
[\![\pi^*]\!]^M &= \big([\![\pi]\!]^M\big)^* \\
[\![\pi^{-1}]\!]^M &= \big([\![\pi]\!]^M\big)^{-1} \\
[\![\pi \cap \pi']\!]^M &= [\![\pi]\!]^M \cap [\![\pi']\!]^M\,.
\end{aligned}$$

Finally, sentences are evaluated at the level of the structure:

$$\begin{array}{lll}
M \models \mathsf{E}\,\varphi & \text{if} & [\![\varphi]\!]^M \neq \emptyset \\
M \models \xi \vee \xi' & \text{if} & M \models \xi \text{ or } M \models \xi' \\
M \models \neg\xi & \text{if} & M \not\models \xi\,.
\end{array}$$

We use the usual abbreviations to define the conjunction $\wedge$ of sentences or event formulas and the event formula *false*. The dual of the $\langle\pi\rangle$ modality is defined as $[\pi]\,\varphi := \neg\langle\pi\rangle\neg\varphi$, and the dual of E quantification as $\mathsf{A}\,\varphi := \neg\,\mathsf{E}\,\neg\varphi$. We also write $\langle\pi\rangle$ for the event formula $\langle\pi\rangle\mathit{true}$. In ICPDL[$\Sigma$], we also define an event formula $\mathsf{Loop}(\pi) := \langle\pi \cap \{\mathit{true}\}?\rangle$, which holds at e if and only if $(e,e) \in [\![\pi]\!]$. Note that this is not a PDL[Σ] formula (and in general, it cannot be defined in PDL[Σ]).

ICPDL macros

Example 2.4. PDL captures various temporal logic modalities. For instance, for $\mathsf{Rel} = \{\rightarrow\}$, the CTL formula $\mathsf{EX}\,\varphi$ can be expressed in PDL as $\langle\rightarrow\rangle\,\varphi$, and $\mathsf{E}\,\varphi\,\mathsf{U}\,\psi$ as $\langle(\{\varphi\}? \cdot \rightarrow)^*\rangle\,\psi$.

Example 2.5. The MSC from Figure 2.1 satisfies the ICPDL[Σ] sentence

$$\mathsf{E}\,\big(p \wedge \neg\langle\rightarrow^{-1}\rangle \wedge \langle(\rightarrow \cdot \rightarrow)^* \cdot \lhd^{-1}\rangle\, r\big)$$

which states that there exists an event on process p which is at an even position (counting from 0 for the leftmost position), and which corresponds to the reception of a message from process r.

Expressivity and complexity. In general, PDL and ICPDL are less expressive than MSO, and incomparable with FO or EMSO. More formally, each event formula $\varphi \in \mathrm{ICPDL}[\Sigma]$ and path formula $\pi \in \mathrm{ICPDL}[\Sigma]$ can be inductively translated (in a standard way, see also Section 3.3) into $\mathrm{MSO}[\Sigma]$ formulas $\widetilde{\varphi}(x)$ and $\widetilde{\pi}(x, y)$ with respectively one and two free variables, such that for all Σ-structures M and events e and f in M, we have $M, e \models \varphi$ if and only if $M, [x \mapsto e] \models \widetilde{\varphi}(x)$, and $M, e, f \models \pi$ if and only if $M, [x \mapsto e, y \mapsto f] \models \widetilde{\pi}(x, y)$. Similarly, each $\mathrm{ICPDL}[\Sigma]$ sentence can be translated into an equivalent $\mathrm{MSO}[\Sigma]$ sentence.

In contrast with MSO and FO, the satisfiability of PDL or ICPDL formulas is decidable (respectively in EXPTIME [28] and 2-EXPTIME [37]). However, this is not always the case for satisfiability over a fixed class of structures $\mathcal{C}$ (that is, the problem of deciding if the input formula has a model in $\mathcal{C}$). For instance, PDL satisfiability over (unbounded) MSCs or over grids is undecidable. Note that neither of these classes is definable in ICPDL.

Chapter 3

Star-free Propositional Dynamic Logic

In this chapter, we introduce *star-free propositional dynamic logic*. Star-free PDL is a variant of PDL which uses the operations from star-free regular expressions $(\cdot, +, \mathsf{c})$, instead of the operations of regular expressions $(\cdot, +, *)$. It also allows the use of the converse operator, as in ICPDL.

In general, star-free PDL is equivalent to FO^3. We show that when restricted to linearly ordered structures where binary relation symbols are interpreted as *interval-preserving relations*, it is also equivalent to full first-order logic.

The equivalence between star-free PDL and first-order logic over linear orders with interval-preserving relations generalizes several known results. In particular, it applies to linear orders and real-time signals, which were respectively shown to have the 3-variable property by Immerman and Kozen [52], and Antonopoulos et al. [3]. The usual approach to proving that a class of structures has the k-variable property is through Ehrenfeucht-Fraïssé games with a bounded number of pebbles [42, 70, 52, 3]. The use of star-free PDL as an intermediate language provides a new effective translation from FO to FO^3. We give other examples of applications in Section 3.7.

We also show in the last section that over *complete* linear orders (with additional interval-preserving relations), the complement operation in star-free PDL can be replaced with path formulas similar to the *until* and *since* modalities of LTL. Besides highlighting interesting connections with temporal logics, this results in a logic with better algorithmic properties.

3.1 Monadic variables

In this chapter, we will only talk about *non*-monadic first-order logic; that is, we consider first-order formulas without any occurrence of predicates $x \in X$. We still denote this logic by $\mathrm{FO}[\Sigma]$.

In other terms, we choose not to distinguish between monadic variables X and relation symbols in Prop. Our results still apply to both monadic and non-monadic first-order logic: we assume Prop and its interpretation to be arbitrary, and we could therefore handle any monadic variable as an extra relation symbol from the signature.

3.2 Syntax and semantics

Star-free propositional dynamic logic ($\mathrm{PDL}_{\mathsf{sf}}$)

We introduced star-free propositional dynamic logic ($\mathrm{PDL}_{\mathsf{sf}}$) in [9] for message sequence charts, and in a more general setting in [29]. It is a variant of ICPDL where the Kleene star operator on path formulas is replaced with a complement operator. The syntax of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ is as follows:

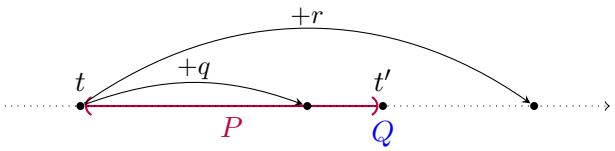
$$\begin{aligned} \xi &::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi && \text{(sentences)} \\ \varphi &::= \mathit{true} \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi && \text{(event formulas)} \\ \pi &::= \alpha \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^{\mathsf{c}} && \text{(path formulas)} \end{aligned}$$

where $P \in \mathsf{Prop}$ and $\alpha \in \mathsf{Rel}$. The semantics is defined as in Section 2.4 for ICPDL, with additionally

$$[\![\pi^{\mathsf{c}}]\!]^M = \left(E^M \times E^M\right) \setminus \left([\![\pi]\!]^M\right).$$

We can define $\pi \cap \pi'$ as $(\pi^{\mathsf{c}} + \pi'^{\mathsf{c}})^{\mathsf{c}}$, and use the same abbreviations as in Section 2.4. We also define $\top := (\{\mathit{true}\}? + \{\mathit{true}\}?^{\mathsf{c}})$. We then have $[\![\top]\!]^M = E^M \times E^M$.

Example 3.1. Suppose that $\mathsf{Rel} = \{<\} \cup \{+q \mid q \in \mathbb{Q}\}$, and that we consider only models with domain $\mathbb{R}$ and with the natural interpretations of $<$ and $+q$: $[\![+q]\!] = \{(r, r+q) \mid r \in \mathbb{R}\}$. Let $q, r \in \mathbb{Q}_{\geq 0}$ and $P, Q \in \mathsf{Prop}$. Let us show that the formula $P\,\mathsf{U}_{(q,r)}\,Q$ of metric temporal logic can be expressed in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$. This formula holds at time $t \in \mathbb{R}$ if there exists $t' \in [\![Q]\!]$ such that $t + q < t' < t + r$ and for all $t < t'' < t'$, $t'' \in [\![P]\!]$, as illustrated below:



It is equivalent to the following $\mathrm{PDL}_{\mathsf{sf}}$ event formula:

$$\langle (+q \cdot <) \cap (+r \cdot >) \cap (< \cdot \{\neg P\}? \cdot <)^{\mathsf{c}} \rangle\, Q\,,$$

where $> := (<)^{-1}$.

Connection with the calculus of relations. Path formulas of star-free PDL use the same operations as terms in the *calculus of relations* [82, 83, 36]: concatenation, converse, union, and complement. The calculus of relations is a logic without quantifiers, which has been studied in proof theory and algebraic logic, as well as foundations of mathematics. Its *terms* are formed from atomic relations (including the identity relation), boolean operations, composition, and converse. *Sentences* in the calculus of relations are equations of the form $t = t'$, where t and t' are terms.

There is therefore a correspondence between path formulas of star-free PDL and terms in the calculus of relation. The identity relation of the calculus of relations corresponds to the formula $\{true\}?$ in star-free PDL. If we include formulas $\{P\}?$ as atomic relations, other test formulas $\{\varphi\}?$ of star-free PDL could be removed by defining $\{\varphi \vee \psi\}?$ as $\{\varphi\}? + \{\psi\}?$, $\{\neg\varphi\}?$ as $\{true\}? \cap \{\varphi\}?^{\mathsf{c}}$, and $\{\langle\pi\rangle\,\varphi\}?$ as $\{true\}? \cap (\pi \cdot \{\varphi\}? \cdot \pi^{-1})$.

The correspondence between sentences of star-free PDL and sentences of the calculus of relations is somewhat less straightforward, but it is always possible to define e.g. a sentence $\pi = \pi'$ in star-free PDL as $\neg\,\mathsf{E}\,\langle \pi \cap \pi'^{\mathsf{c}} + \pi^{\mathsf{c}} \cap \pi' \rangle$.

However, the focus of star-free PDL and the calculus of relations is different. It is easier to reason about properties of events in star-free PDL, and the distinction between boolean operations at the level of event formulas and boolean operations at the level of path formulas will become important in the next sections. On the other hand, most of the work on the calculus of relations is concerned with proof systems and algebraic axiomatisations.

3.3 Equivalence of star-free PDL and FO^3

Over arbitrary structures, $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ is equivalent to $\mathrm{FO}^3[\Sigma]$. This can be deduced from the equivalence of first-order logic and the calculus of relations [83]. Nevertheless, we provide a proof of this result, first for completeness, and also to illustrate the difference with the case of first-order logic with arbitrarily many variables.

Equivalence between FO and $\mathrm{PDL}_{\mathsf{sf}}$ formulas

Let us first formalize this equivalence. We say that a sentence $\xi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ is equivalent to a sentence $\Phi \in \mathrm{FO}[\Sigma]$, written $\xi \equiv \Phi$, when for all Σ-structures M, we have $\xi \models M$ if and only if $\Phi \models M$. Given an event formula $\varphi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, and an FO formula $\Phi(x)$ with a single free variable x, we say that φ and Φ are equivalent, written $\varphi \equiv \Phi(x)$, if for all M and events e in M, we have $M, e \models \varphi$ if and only if $M, [x \mapsto e] \models \Phi(x)$. Similarly, for a path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ and an $\mathrm{FO}[\Sigma]$ formula $\Phi(x, y)$ with exactly two (ordered) free variables x and y, we write $\pi \equiv \Phi(x, y)$ if for all M and events e, f in M, we have $M, e, f \models \pi$ if and only if $M, [x \mapsto e, y \mapsto f] \models \Phi(x, y)$.

Example 3.2. Let us give $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ formulas equivalent to the $\mathrm{FO}^3[\Sigma]$ formulas from Example 2.3. First, the formula $x \leq y := x < y \vee x = y$ is equivalent to the

path formula

$$\leq \;:=\; < + \{true\}? \,.$$

The formula $p(x) \leq p(y)$ is then equivalent to the path formula

$$\pi := p \cdot \leq \cdot p^{-1}$$

Finally, $\min(x)$ is equivalent to the formula

$$\langle > \cap ((\pi^{\mathsf{c}} \cap >) \cdot >)^{\mathsf{c}} \rangle \quad \wedge \quad \langle < \cap ((\pi^{\mathsf{c}} \cap <) \cdot <)^{\mathsf{c}} \rangle$$

where $> = (<)^{-1}$.

An easy induction shows that any formula in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ can be translated into an $\mathrm{FO}[\Sigma]$ formula which uses at most three distinct variables:

Lemma 3.3.

1. *For every sentence $\xi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, there exists a sentence $\widetilde{\xi} \in \mathrm{FO}^3[\Sigma]$ such that $\xi \equiv \widetilde{\xi}$.*

2. *For every event formula $\varphi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, there exists a formula $\widetilde{\varphi}(x) \in \mathrm{FO}^3[\Sigma]$ such that $\varphi \equiv \widetilde{\varphi}(x)$.*

3. *For every path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, there exists a formula $\widetilde{\pi}(x,y) \in \mathrm{FO}^3[\Sigma]$ such that $\pi \equiv \widetilde{\pi}(x,y)$.*

Proof. The formulas $\widetilde{\xi}$, $\widetilde{\varphi}(x)$ and $\widetilde{\pi}(x,y)$ are defined inductively as follows:

$$\begin{aligned}
\widetilde{\mathsf{E}\varphi} &:= \exists x.\widetilde{\varphi}(x) & \widetilde{\alpha}(x,y) &:= \alpha(x,y) \\
\widetilde{\xi \vee \xi'} &:= \widetilde{\xi} \vee \widetilde{\xi'} & \widetilde{\{\varphi\}?}(x,y) &:= \widetilde{\varphi}(x) \wedge x = y \\
\widetilde{\neg\xi} &:= \neg\widetilde{\xi} & \widetilde{\pi^{-1}}(x,y) &:= \widetilde{\pi}(y,x) \\
\widetilde{true}(x) &:= \forall x.(x = x) & \widetilde{\pi \cdot \pi'}(x,y) &:= \exists z.\widetilde{\pi}(x,z) \wedge \widetilde{\pi'}(z,y) \\
\widetilde{P}(x) &:= P(x) & \widetilde{\pi + \pi'}(x,y) &:= \widetilde{\pi}(x,y) \vee \widetilde{\pi'}(x,y) \\
\widetilde{\varphi \vee \varphi'}(x) &:= \widetilde{\varphi}(x) \vee \widetilde{\varphi'}(x) & \widetilde{\pi^{\mathsf{c}}}(x,y) &:= \neg\widetilde{\pi}(x,y) \\
\widetilde{\neg\varphi}(x) &:= \neg\widetilde{\varphi}(x) & & \\
\widetilde{\langle\pi\rangle\,\varphi}(x) &:= \exists y.\widetilde{\pi}(x,y) \wedge \widetilde{\varphi}(y) & &
\end{aligned}$$

□

The converse is also true. For formulas with three free variables, the translation in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ gives a boolean combination of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ path formulas.

Proposition 3.4.

1. *Any* $\mathrm{FO}^3[\Sigma]$ *sentence is equivalent to some* $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ *sentence.*
2. *Any* $\mathrm{FO}^3[\Sigma]$ *formula with a single free variable is equivalent to some* $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ *event formula.*
3. *Any* $\mathrm{FO}^3[\Sigma]$ *formula with two free variables is equivalent to some* $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ *path formula.*
4. *Any* $\mathrm{FO}^3[\Sigma]$ *formula* $\Phi(x,y,z)$ *with three free variables is equivalent to a finite disjunction of formulas of the form* $\widetilde{\pi}(x,y) \wedge \widetilde{\pi}'(x,z) \wedge \widetilde{\pi}''(y,z)$*, where* $\pi, \pi', \pi'' \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$.

Proof. We prove the result by induction on $\mathrm{FO}^3[\Sigma]$ formulas. For atomic formulas, we have:

$$\begin{array}{lll} P(x) \equiv P & \alpha(x,x) \equiv \mathsf{Loop}(\alpha) & (x=x) \equiv \mathit{true} \\ \alpha(x,y) \equiv \alpha & \alpha(y,x) \equiv \alpha^{-1} & (x=y) \equiv (y=x) \equiv \{\mathit{true}\}? \,. \end{array}$$

The case of negation is easy since it is allowed at the level of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ sentences and event formulas, and corresponds to the complement operation for path formulas. For formulas with three free variables, if $\Phi(x,y,z) \equiv \bigvee_i \widetilde{\pi}_i(x,y) \wedge \widetilde{\pi}'_i(x,z) \wedge \widetilde{\pi}''_i(y,z)$ (induction hypothesis), then

$$\neg\Phi(x,y,z) \equiv \bigwedge_i \widetilde{\pi_i^{\mathsf{c}}}(x,y) \vee \widetilde{\pi_i'^{\mathsf{c}}}(x,z) \vee \widetilde{\pi_i''^{\mathsf{c}}}(y,z) \,,$$

and we only need to bring the resulting formula into disjunctive normal form.

For a disjunction $\Phi \vee \Psi$, the only non-trivial cases are when the two subformulas do not have the same free variables. If $\Phi(x) \equiv \varphi$ and $\Psi(x,y) \equiv \pi$, then

$$\Phi(x) \vee \Psi(x,y) \equiv (\{\varphi\}? \cdot \top) + \pi \,.$$

If $\Phi(x) \equiv \varphi$ and $\Psi(x) \equiv \psi$, then

$$\Phi(x) \vee \Psi(y) \equiv (\{\varphi\}? \cdot \top) + (\top \cdot \{\psi\}?) \,.$$

If $\Phi(x,y) \equiv \pi$ and $\Psi(x,y) \equiv \pi'$, then

$$\Phi(x,y) \vee \Psi(x,z) \equiv \Big(\widetilde{\pi}(x,y) \wedge \widetilde{\top}(x,z) \wedge \widetilde{\top}(y,z)\Big) \vee \Big(\widetilde{\top}(x,y) \wedge \widetilde{\pi}'(x,z) \wedge \widetilde{\top}(y,z)\Big) \,.$$

The other cases are similar.

We are left with the case of existential quantification. The interesting case is $\Phi(x,y) = \exists z.\Psi(x,y,z)$. By induction (and distribution of $\exists$ over $\vee$), $\Phi(x,y)$ is equivalent to a finite disjunction of formulas of the form

$$\widetilde{\pi}(x,y) \wedge \exists z. \big(\widetilde{\pi}'(x,z) \wedge \widetilde{\pi}''(y,z)\big) \equiv \pi \cap (\pi' \cdot \pi''^{-1}) \,.$$

The other cases are easy: if $\Phi = \exists x.\Psi(x)$ and $\Psi(x) \equiv \varphi$, then $\Phi \equiv \mathsf{E}\,\varphi$. If $\Phi = \exists x.\Psi$ and $\Psi \equiv \xi$, then $\Phi \equiv \xi \wedge \mathsf{E}\,\mathit{true}$. If $\Phi(x) = \exists y.\Psi(x,y)$ and $\Psi(x,y) \equiv \pi$, then $\Phi(x) \equiv \langle\pi\rangle$. If $\Phi(x) = \exists y.\Psi(x)$ and $\Psi(x) \equiv \varphi$, then $\Phi(x) \equiv \varphi$. If $\Phi(x,y) = \exists z.\Psi(x,y)$, and $\Psi(x,y) \equiv \pi$, then $\Phi(x,y) \equiv \pi$. □

Remark 3.5. The fragment of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ where concatenation of path formulas is disallowed is expressively equivalent to $\mathrm{FO}^2[\Sigma]$ [62].

Going beyond 3 variables. In the remainder of the chapter, we define conditions under which $\mathrm{FO}[\Sigma]$, $\mathrm{FO}^3[\Sigma]$ and $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ have the same expressive power. As in Proposition 3.4, we want to define an inductive translation from $\mathrm{FO}[\Sigma]$ formulas (this time with arbitrarily many variables) into $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ formulas, or boolean combinations of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ formulas if there are more than two free variables. The difficulty is to deal with existential quantification, typically, formulas such as

$$\exists x.\widetilde{\pi}_1(x_1,x) \wedge \widetilde{\pi}_2(x_2,x) \wedge \cdots \wedge \widetilde{\pi}_n(x_n,x)\,.$$

Informally, this formula holds if and only if the intersection of all $[\![\pi_i]\!](x_i)$ is non-empty. For $n = 2$, as in Proposition 3.4, this is easy to check: $[\![\pi_1]\!](x_1) \cap [\![\pi_2]\!](x_2)$ is non-empty if and only if $(x_1,x_2) \in [\![\pi_1 \cdot \pi_2^{-1}]\!]$. In the next two sections, we define restrictions on the structures and on the formulas which ensure that each $[\![\pi_i]\!](x_i)$ is an interval; then checking that the intersection of the n intervals is nonempty comes down to checking that the *pairwise* intersections are nonempty, which corresponds to the case $n = 2$.

3.4 Interval-preserving relations

In this section, we define interval-preserving relations and interval-preserving structures, and state some of their basic properties.

Linear orders. A *partial order* $\leq$ over a set E is a reflexive, transitive and antisymmetric relation $\leq \subseteq E \times E$. A *linear order* or *total order* is a partial order such that for all $e, f \in E$, we have $e \leq f$ or $f \leq e$. We also call linear order the pair $(E, \leq)$.

For $F \subseteq E$, we also denote by $\leq$ the restriction of $\leq$ to F. Then, if $(E, \leq)$ is a linear order, $(F, \leq)$ is also one. We write $e < f$ if $e \leq f$ and $f \neq e$. Moreover, for $e \in E$, we write $e < F$ if for all $f \in F$, $e < f$, and $F < e$ if for all $f \in F$, $f < e$. For $F, G \subseteq E$, we write $F < G$ if for all $f \in F$ and $g \in G$, $f < g$. We define similarly $e \leq F$, $F \leq e$, and $F \leq G$.

Figure 3.1: Definition of interval-preserving relations (arrows represent $\mathcal{R}$).

Interval-preserving relations. Let $(E, \leq)$ be a linear order. An *interval* of $(E, \leq)$ is a set $I \subseteq E$ such that for all $e \leq f \leq g$ with $e, g \in I$, we have $f \in I$. For $e, f \in E$, we denote by $[a, b)$ the interval $\{c \in E \mid a \leq c < b\}$, and similarly for the intervals $[a, b]$, $(a, b]$, (a, b). We call a relation $\mathcal{R} \subseteq E \times E$ over a linear order $(E, \leq)$ *interval-preserving* if:

interval-preserving relation

- For all intervals I of $(E, \leq)$, $\mathcal{R}(I)$ is an interval of $(\mathcal{R}(E), \leq)$.
- For all intervals J of $(E, \leq)$, $\mathcal{R}^{-1}(J)$ is an interval of $(\mathcal{R}^{-1}(E), \leq)$.

In other terms, for all $e_1 \mathcal{R} f_1$ and $e_2 \mathcal{R} f_2$ with $e_1, e_2 \in I$, for all $f_1 \leq f \leq f_2$, if there exists some $e \in E$ such that $e \mathcal{R} f$, then there exists one in I (cf. Figure 3.1). Note that we do not require that *all* elements between f_1 and f_2 are in $\mathcal{R}(I)$, but only those which are in the image of $\mathcal{R}$. Notice also that we may have $e_1 \leq e_2$ or $e_2 \leq e_1$. The second condition is symmetric: for all $e_1 \mathcal{R} f_1$ and $e_2 \mathcal{R} f_2$ with $f_1, f_2 \in J$, for all $e_1 \leq e \leq e_2$, if there exists some $f \in F$ such that $e \mathcal{R} f$, then there exists one in J.

Example 3.6. For any linear order $(E, \leq)$ and partial function $f : E \to E$, if f is increasing or decreasing then the relation $\{(e, f(e)) \mid e \in \mathrm{dom}(f)\}$ is interval-preserving.

As another example, consider a structure $M = (E, \leq, (P^M)_{P \in \mathsf{Prop}})$ such that $\leq$ is a linear order. For $P, Q \in \mathsf{Prop}$, let

$$\mathsf{until}_{P,Q} = \{(e, f) \in E \times E \mid e < f \wedge f \in Q^M \wedge \forall e < g < f, g \in P^M\}.$$

Then $\mathsf{until}_{P,Q}$ is interval-preserving.

The following lemma states some simple closure properties of interval-preserving relations.

Lemma 3.7. *Let $(E, \leq)$ be a linear order, and $\mathcal{R}, \mathcal{R}_1, \mathcal{R}_2 \subseteq E \times E$ interval-preserving relations. Then:*

1. *$\mathcal{R}^{-1}$ is interval-preserving.*
2. *$\mathcal{R}_1 \cap \mathcal{R}_2$ is interval-preserving.*

3. $\mathcal{R}_1 \cdot \mathcal{R}_2$ is interval-preserving.

Proof. Part 1 follows from the fact that $(\mathcal{R}^{-1})^{-1} = \mathcal{R}$.

Let us prove 2. Since $(\mathcal{R}_1 \cap \mathcal{R}_2)^{-1} = \mathcal{R}_1^{-1} \cap \mathcal{R}_2^{-1}$, by symmetry, it suffices to prove that for all intervals I of $(E, \leq)$, $(\mathcal{R}_1 \cap \mathcal{R}_2)(I)$ is an interval of $((\mathcal{R}_1 \cap \mathcal{R}_2)(E), \leq)$. Let $e_1, e_2 \in I$ and $f_1 \leq f \leq f_2$ such that $(e_1, f_1), (e_2, f_2) \in (\mathcal{R}_1 \cap \mathcal{R}_2)$ and $(e, f) \in (\mathcal{R}_1 \cap \mathcal{R}_2)$ for some $e \in E$. If $e \in I$, then we are done. Otherwise, suppose for instance that $e < e_1 \leq e_2$ (the other cases, $e < e_2 \leq e_1$ or $e_1 \leq e_2 < e$ or $e_2 \leq e_1 < e$, are similar). Since $\mathcal{R}_1$ is interval-preserving, there exists $e_1 \leq e' \leq e_2$ such that $e' \; \mathcal{R}_1 \; f$. Then, since $e < e_1 \leq e'$ and $\mathcal{R}_1^{-1}(f)$ is an interval of $(\mathcal{R}_1^{-1}(E), \leq)$, we obtain $e_1 \; \mathcal{R}_1 \; f$. Similarly, $e_1 \; \mathcal{R}_2 \; f$. Hence $e_1 \; (\mathcal{R}_1 \cap \mathcal{R}_2) \; f$.

Let us show that 2 implies 3. Again, by symmetry, it suffices to prove that for all intervals I of $(E, \leq)$, $(\mathcal{R}_1 \cdot \mathcal{R}_2)(I)$ is an interval of $((\mathcal{R}_1 \cdot \mathcal{R}_2)(E), \leq)$. Let $\mathcal{R}_3 \subseteq E \times E$ denote the relation $\mathcal{R}_1(E) \times E$. It is interval-preserving, since for all intervals I of $(E, \leq)$, $\mathcal{R}_3(I)$ is either empty (if $I \cap \mathcal{R}_1(E) = \emptyset$) or equal to $E = \mathcal{R}_3(E)$, and similarly for $\mathcal{R}_3^{-1}(I)$. Moreover, we have

$$\begin{aligned}(\mathcal{R}_1 \cdot \mathcal{R}_2)(E) &= \{g \in E \mid \exists e, f \in E.\ (e, f) \in \mathcal{R}_1 \wedge (f, g) \in \mathcal{R}_2\} \\ &= \{g \in E \mid \exists f \in E.\ f \in \mathcal{R}_1(E) \wedge (f, g) \in \mathcal{R}_2\} \\ &= \{g \in E \mid \exists f \in E.\ (f, g) \in \mathcal{R}_3 \wedge (f, g) \in \mathcal{R}_2\} \\ &= (\mathcal{R}_2 \cap \mathcal{R}_3)(E)\,.\end{aligned}$$

Now, let I be some interval of $(E, \leq)$, and J be the smallest interval of $(E, \leq)$ containing $\mathcal{R}_1(I)$:

$$J := \{f \in E \mid \exists f_1, f_2 \in \mathcal{R}_1(I), f_1 \leq f \leq f_2\}\,.$$

Note that J is indeed an interval of $(E, \leq)$. Moreover, by definition of $\mathcal{R}_1$ being interval-preserving, we have $J \cap \mathcal{R}_1(E) \subseteq \mathcal{R}_1(I)$, hence $J \cap \mathcal{R}_1(E) = \mathcal{R}_1(I)$. Then

$$\begin{aligned}(\mathcal{R}_1 \cdot \mathcal{R}_2)(I) &= \mathcal{R}_2(\mathcal{R}_1(I)) \\ &= \mathcal{R}_2(J \cap \mathcal{R}_1(E)) \\ &= \{f \in E \mid \exists e \in J.\ e \in \mathcal{R}_1(E) \wedge (e, f) \in \mathcal{R}_2\} \\ &= \{f \in E \mid \exists e \in J.\ (e, f) \in \mathcal{R}_3 \wedge (e, f) \in \mathcal{R}_2\} \\ &= (\mathcal{R}_2 \cap \mathcal{R}_3)(J)\,.\end{aligned}$$

Then, according to 2, $(\mathcal{R}_1 \cdot \mathcal{R}_2)(I)$ is an interval of $((\mathcal{R}_2 \cap \mathcal{R}_3)(E), \leq)$, i.e., an interval of $((\mathcal{R}_1 \cdot \mathcal{R}_2)(E), \leq)$. □

Interval-preserving structures. Let $\Sigma = (\mathsf{Prop}, \mathsf{Rel})$ be a signature with a distinguished symbol $\leq \in \mathsf{Rel}$, and $M = (E^M, (P^M)_{P \in \mathsf{Prop}}, (\alpha^M)_{\alpha \in \mathsf{Rel}})$ a Σ-structure. We say that M is *interval-preserving* if $\leq^M$ is a linear order over E^M, and for all $\alpha \in \mathsf{Rel}$, α^M is interval-preserving.

3.5 An interval-preserving fragment of star-free PDL

In this section, we define a fragment of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ in which all path formulas are interval-preserving, that is, path formulas define interval-preserving relations when interpreted in interval-preserving Σ-structures. We will show in the next section that this fragment captures $\mathrm{FO}[\Sigma]$ (over interval-preserving Σ-structures).

interval-preserving formula

Definition 3.8. A path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ is called *interval-preserving* if for all interval-preserving Σ-structures M, $[\![\pi]\!]^M$ is interval-preserving.

Notice that for all φ, the formula $\{\varphi\}?$ is interval-preserving. By definition, all atomic path formulas $\alpha \in \mathsf{Rel}$ are also interval-preserving. By Lemma 3.7, all $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ formulas constructed without the boolean operators $+$ and c (but possibly with $\cap$) are interval-preserving. However, the complement or the union of interval-preserving relations are not in general interval-preserving, since the complement or union of intervals are not always intervals.

In order to define an (expressive enough) interval-preserving fragment of $\mathrm{PDL}_{\mathsf{sf}}$, we thus need to restrict the use of the complement operator. The next lemma shows that when the complement is applied only to formulas of the form $\lambda \cdot \pi \cdot \mu$ with $\lambda, \mu \in \{\leq, \leq^{-1}\}$ it results in interval-preserving formulas. As in previous examples, we denote the formula $\leq^{-1}$ by $\geq$.

Lemma 3.9. *For all path formulas $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, the formulas*

$$(\leq \cdot \pi \cdot \leq)^{\mathsf{c}} \qquad (\leq \cdot \pi \cdot \geq)^{\mathsf{c}} \qquad (\geq \cdot \pi \cdot \leq)^{\mathsf{c}} \qquad (\geq \cdot \pi \cdot \geq)^{\mathsf{c}}$$

are interval-preserving.

Proof. Let us show that $(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}$ is interval-preserving. We have

$$(e, f) \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!] \quad \text{if and only if} \quad f < [\![\leq \cdot \pi]\!](e)\,.$$

We can then make the following observations: if $(e, f) \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]$,

- For all $f' \leq f$, $(e, f') \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]$. Indeed, we have
$$f' \leq f < [\![\leq \cdot \pi]\!](e)\,.$$
- For all $e' \geq e$, $(e', f) \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]$. Indeed, we have
$$f < [\![\leq \cdot \pi]\!](e) \supseteq [\![\leq \cdot \pi]\!](e')\,.$$

This proves that $(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}$ is interval-preserving. Indeed, for all intervals I, for all $e_1 \leq e \leq e_2$, we have

$$e_2 \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!](I) \implies e \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!](I) \qquad \text{and}$$
$$e_1 \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]^{-1}(I) \implies e \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]^{-1}(I)\,.$$

Therefore, $[\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!](I)$ and $[\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]^{-1}(I)$ are intervals.

This also implies that $(\geq \cdot \pi \cdot \geq)^{\mathsf{c}}$ is interval-preserving. Indeed, we have $(\pi^{-1})^{\mathsf{c}} \equiv (\pi^{\mathsf{c}})^{-1}$ for all π, so in particular,

$$(\geq \cdot \pi \cdot \geq)^{\mathsf{c}} \equiv ((\leq \cdot \pi^{-1} \cdot \leq)^{\mathsf{c}})^{-1}.$$

The remaining cases are similar. For $(\leq \cdot \pi \cdot \geq)^{\mathsf{c}}$, we have

$$(e, f) \in [\![(\leq \cdot \pi \cdot \geq)^{\mathsf{c}}]\!] \quad \text{if and only if} \quad [\![\leq \cdot \pi]\!](e) < f\,.$$

Therefore, if $(e, f) \in [\![(\leq \cdot \pi \cdot \geq)^{\mathsf{c}}]\!]$:

- For all $f \leq f'$, $(e, f') \in [\![(\leq \cdot \pi \cdot \geq)^{\mathsf{c}}]\!]$.
- For all $e' \geq e$, $(e', f) \in [\![(\leq \cdot \pi \cdot \geq)^{\mathsf{c}}]\!]$.

Finally, for $(\geq \cdot \pi \cdot \leq)^{\mathsf{c}}$, we have

$$(e, f) \in [\![(\geq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!] \quad \text{if and only if} \quad f < [\![\geq \cdot \pi]\!](e)\,.$$

Therefore, if $(e, f) \in [\![(\geq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]$:

- For all $f' \leq f$, $(e, f') \in [\![(\geq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]$.
- For all $e' \leq e$, $(e', f) \in [\![(\geq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]$. □

Note that Lemma 3.9 holds even if π is not interval-preserving.

Corollary 3.10. *Any path formula π over the following syntax is interval-preserving:*

$$\varphi ::= \mathit{true} \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\, \varphi$$
$$\pi ::= \alpha \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cap \pi \mid (\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}\,,$$

where $P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel}$, and $\lambda, \mu \in \{\leq, \geq\}$.

The restriction of complement operators to formulas of the form $\lambda \cdot \pi \cdot \mu$ (where $\lambda, \mu \in \{\leq, \geq\}$) is not as strong as it may seem. Indeed, in interval-preserving Σ-structures, the formulas $\lambda \cdot \pi \cdot \mu$ suffice to characterize π:

Lemma 3.11. *For all interval-preserving $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, over interval-preserving structures,*

$$\pi \equiv (\leq \cdot \pi \cdot \leq) \cap (\leq \cdot \pi \cdot \geq) \cap (\geq \cdot \pi \cdot \leq) \cap (\geq \cdot \pi \cdot \geq) \cap (\{\langle\pi\rangle\}? \cdot \top \cdot \{\langle\pi^{-1}\rangle\}?)\,.$$

Proof. The direct implication is obvious. Conversely, let

$$(e,f) \in [\![(\leq \cdot \pi \cdot \leq) \cap (\leq \cdot \pi \cdot \geq) \cap (\geq \cdot \pi \cdot \leq) \cap (\geq \cdot \pi \cdot \geq) \cap (\{\langle \pi \rangle\}? \cdot \top \cdot \{\langle \pi^{-1} \rangle\}?)]\!] .$$

Since $(e,f) \in [\![(\leq \cdot \pi \cdot \leq)]\!]$, there exists $(e_1,f_1) \in [\![\pi]\!]$ such that $e \leq e_1$ and $f_1 \leq f$. Similarly, there exists (e_2,f_2) such that $e \leq e_2$ and $f \leq f_2$. Let I be the closed interval delimited by e_1 and e_2, i.e., $I = [e_1,e_2]$ or $I = [e_2,e_1]$. We have $f_1 \leq f \leq f_2$ and $f \in [\![\langle \pi^{-1} \rangle]\!]$. Since π is interval-preserving, there exists $e' \in I$ such that $(e',f) \in [\![\pi]\!]$. Note that $e' \geq e$.

Similarly, since $(e,f) \in [\![(\geq \cdot \pi \cdot \leq) \cap (\geq \cdot \pi \cdot \geq)]\!]$, there exists $e'' \leq e$ such that $(e'',f) \in [\![\pi]\!]$. Since $e'' \leq e \leq e'$ and $e \in [\![\langle \pi \rangle]\!]$, we obtain $(e,f) \in [\![\pi]\!]$. □

Using Lemma 3.11, it is not very difficult to prove that at the level of sentences or event formulas, $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ and the fragment defined in Corollary 3.10 have the same expressive power. In fact, we can even further restrict this fragment by limiting the use of the intersection to $\mathsf{Loop}(\pi)$ formulas, as defined below. The fact that the resulting logic is expressively equivalent to $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ is proved in Section 3.6.

Definition 3.12. Let $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ be the set of formulas $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$

$$\begin{aligned} \xi &::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi \\ \varphi &::= \mathit{true} \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle \pi \rangle\, \varphi \mid \mathsf{Loop}(\pi) \\ \pi &::= \alpha \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid (\lambda \cdot \pi \cdot \mu)^{\mathsf{c}} , \end{aligned}$$

where $P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel}$, and $\lambda, \mu \in \{\leq, \geq\}$.

It follows from Corollary 3.10 that all path formulas of $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ are interval-preserving.

Lemma 3.13. *All path formulas $\pi \in \mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ are interval-preserving.*

3.6 Equivalence of FO and $\mathrm{PDL}_{\mathsf{sf}}$ over interval-preserving structures

The main result of the chapter is an effective translation of $\mathrm{FO}[\Sigma]$ formulas into finite positive boolean combinations of formulas in $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$, over interval-preserving structures:

Theorem 3.14. *Every formula $\Phi \in \mathrm{FO}[\Sigma]$ with at least one free variable is equivalent, over interval-preserving Σ-structures, to a finite positive boolean combination of formulas of the form $\widetilde{\pi}(x,y)$, where $x,y \in \mathrm{Free}(\Phi)$ and $\pi \in \mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$.*

Note that the equivalent formula may also contain subformulas of the form $\widetilde{\pi}(x,x)$.

Before proving Theorem 3.14, we state some of its consequences.

Corollary 3.15. *Over interval-preserving Σ-structures,*

1. *Every sentence $\Phi \in \mathrm{FO}[\Sigma]$ is equivalent to a $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ sentence.*

2. *Every formula $\Phi(x) \in \mathrm{FO}[\Sigma]$ with a single free variable is equivalent to a $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ event formula.*

3. *Every formula $\Phi(x, y) \in \mathrm{FO}[\Sigma]$ with two free variables is equivalent to a $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ path formula (or to a finite positive boolean combination of $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ path formulas).*

Proof. We first prove 2, which immediately implies 1. By Theorem 3.14, $\Phi(x)$ is equivalent to a finite positive boolean combination of formulas of the form $\widetilde{\pi}(x, x)$, which are themselves equivalent to the formulas $\mathsf{Loop}(\widetilde{\pi})$. The combination of these $\mathsf{Loop}(\widetilde{\pi})$ formulas is then an event formula of $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$.

For 3, we know that $\Phi(x, y)$ is equivalent to a finite positive boolean combination of formulas of the form $\widetilde{\pi}(x, y)$, $\widetilde{\pi}(y, x)$, $\widetilde{\pi}(x, x)$, or $\widetilde{\pi}(y, y)$. We can replace any subformula $\widetilde{\pi}(y, x)$ with $\widetilde{\pi^{-1}}(x, y)$, and any subformula $\widetilde{\pi}(x, x)$ with $\widetilde{\pi_1}(x, y) \vee \widetilde{\pi_2}(x, y)$, where $\pi_1 = (\{\mathsf{Loop}(\pi)\}? \cdot \leq)$ and $\pi_2 = (\{\mathsf{Loop}(\pi)\}? \cdot \geq)$, and similarly for formulas $\widetilde{\pi}(y, y)$. We obtain an equivalent positive boolean combination of formulas of the form $\widetilde{\pi}(x, y)$. Since $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ allows union and intersection of path formulas, this is equivalent to a $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ formula. □

Another consequence of Theorem 3.14 and Lemma 3.3 is that the class of interval-preserving Σ-structures has the strong 3-variable property:

Theorem 3.16. *Over interval-preserving Σ-structures, any formula in $\mathrm{FO}[\Sigma]$ is equivalent to a finite boolean combination of formulas in $\mathrm{FO}^3[\Sigma]$.*

The remainder of the section is devoted to the proof of Theorem 3.14.

Eliminating negations. In order to deal with negation in the inductive translation from $\mathrm{FO}[\Sigma]$ to $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, we need to show that the complement of a $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ path formula can be expressed as a positive boolean combination of $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ path formulas. This is an immediate consequence of Lemma 3.11:

Lemma 3.17. *For all path formulas $\pi \in \mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$, π^{c} is equivalent, over interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ formulas.*

Proof. By Lemma 3.11, over interval-preserving Σ-structures,

$$\pi \equiv (\leq \cdot \pi \cdot \leq) \cap (\leq \cdot \pi \cdot \geq) \cap (\geq \cdot \pi \cdot \leq) \cap (\geq \cdot \pi \cdot \geq) \cap (\{\langle \pi \rangle\}? \cdot \top \cdot \{\langle \pi^{-1} \rangle\}?)\,.$$

Therefore,

$$\pi^{\mathsf{c}} \equiv (\leq \cdot \pi \cdot \leq)^{\mathsf{c}} + (\leq \cdot \pi \cdot \geq)^{\mathsf{c}} + (\geq \cdot \pi \cdot \leq)^{\mathsf{c}} + (\geq \cdot \pi \cdot \geq)^{\mathsf{c}} + (\{\langle \pi \rangle\}? \cdot \top \cdot \{\langle \pi^{-1} \rangle\}?)^{\mathsf{c}},$$

and we only need to show that $(\{\langle \pi \rangle\}? \cdot \top \cdot \{\langle \pi^{-1} \rangle\}?)^{\mathsf{c}}$ is equivalent to a finite union of $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ path formulas. This is the case, as

$$\begin{aligned}
&(\{\langle \pi \rangle\}? \cdot \top \cdot \{\langle \pi^{-1} \rangle\}?)^{\mathsf{c}} \\
&\quad \equiv \{\neg\langle \pi \rangle\}? \cdot \top + \top \cdot \{\neg\langle \pi^{-1} \rangle\}? \\
&\quad \equiv (\{\neg\langle \pi \rangle\}? \cdot \leq) + (\{\neg\langle \pi \rangle\}? \cdot \geq) + (\leq \cdot \{\neg\langle \pi^{-1} \rangle\}?) + (\geq \cdot \{\neg\langle \pi^{-1} \rangle\}?)\,. \qquad \square
\end{aligned}$$

Existential quantification. The elimination of existential quantifiers relies on the simple lemma below:

Lemma 3.18. *Let $(E, \leq)$ be a linear order, and $I_1, \ldots, I_n$ intervals of $(E, \leq)$. Then $\bigcap_{1 \leq i \leq n} I_i \neq \emptyset$ if and only if for all $1 \leq i, j \leq n$, $I_i \cap I_j \neq \emptyset$.*

Proof. We show that there exists k and ℓ such that $\bigcap_{1 \leq i \leq n} I_i = I_k \cap I_\ell$, which implies the result. We define relations $\sqsubseteq_{\text{left}}$ and $\sqsubseteq_{\text{right}}$ over $\{I_1, \ldots, I_n\}$ which, intuitively, compare respectively the left and right bounds of the intervals:

$$\begin{aligned}
I \sqsubseteq_{\text{left}} J \qquad &\text{if} \qquad \forall e \in J, \exists f \in I, f \leq e \\
I \sqsubseteq_{\text{right}} J \qquad &\text{if} \qquad \forall e \in I, \exists f \in J, e \leq f\,.
\end{aligned}$$

It is easy to check that $\sqsubseteq_{\text{left}}$ and $\sqsubseteq_{\text{right}}$ are transitive and total, that is, for all I, J, $I \sqsubseteq_{\text{left}} J$ or $J \sqsubseteq_{\text{left}} I$ (and similarly for $\sqsubseteq_{\text{right}}$). Thus, there exist k and ℓ such that $I_i \sqsubseteq_{\text{left}} I_k$ for all i, and $I_\ell \sqsubseteq_{\text{right}} I_i$ for all i. Then for all $e \in I_k \cap I_\ell$, for all i, there exist $f, f' \in I_i$ such that $f \leq e \leq f'$. Since I_i is an interval, we obtain $e \in I_i$. Hence $I_k \cap I_\ell = \bigcap_{1 \leq i \leq n} I_i$. $\square$

The next lemma follows from an application of Lemma 3.18 to intervals of the form $[\![\pi_i]\!](e_i)$.

Lemma 3.19. *Let $n \geq 1$. For all path formulas $\pi_1, \ldots, \pi_n$ and all event formulas φ in $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$, the $\mathrm{FO}[\Sigma]$ formula*

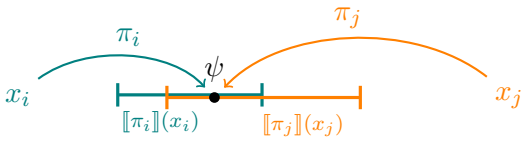
$$\Phi(x_1, \ldots, x_n) = \exists x. \left(\widetilde{\varphi}(x) \wedge \bigwedge_{1 \leq i \leq n} \widetilde{\pi}_i(x_i, x) \right) \qquad (x_i \neq x \text{ for all } i)$$

is equivalent, over interval-preserving Σ-structures, to a finite positive boolean combination of formulas of the form $\widetilde{\pi}(x_j, x_k)$, with $1 \leq j, k \leq n$ and $\pi \in \mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$.

Proof. We define a $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ event formula $\psi = \varphi \wedge \bigwedge_{1\le i\le n} \langle \pi_i^{-1}\rangle$, and

$$\Psi(x_1,\dots,x_n) = \bigwedge_{1\le i,j\le n} (\pi_i \cdot \widetilde{\{\psi\}?} \cdot \pi_j^{-1})(x_i,x_j)\,.$$

Intuitively, $\Psi(x_1,\dots,x_n)$ states that for all i,j, the intersection of $[\![\pi_i]\!](x_i)$ and $[\![\pi_j]\!](x_j)$ contains at least one event satisfying ψ:



Let us show that, over interval-preserving Σ-structures,

$$\Phi(x_1,\dots,x_n) \equiv \Psi(x_1,\dots,x_n)\,.$$

Let M be an interval-preserving Σ-structure, and $\nu : \{x_1,\dots,x_n\} \to E^M$. For all $1 \le i \le n$, let $I_i = [\![\pi_i]\!](\nu(x_i)) \cap [\![\psi]\!] = \{e \mid e \in [\![\psi]\!] \wedge (\nu(x_i), e) \in [\![\pi_i]\!]\}$. Let us show that I_i is an interval of $([\![\psi]\!], \le)$. First, since π_i is interval-preserving and the singleton $\{\nu(x_i)\}$ is an interval, $[\![\pi_i]\!](\nu(x_i))$ is an interval of $([\![\langle \pi_i^{-1}\rangle]\!], \le)$. Thus, I_i is an interval of $([\![\langle \pi_i^{-1}\rangle]\!] \cap [\![\psi]\!], \le)$. But since $[\![\langle \pi_i^{-1}\rangle]\!] \supseteq [\![\psi]\!]$, this is simply $([\![\psi]\!], \le)$. Besides, it is easy to verify that

$$M,\nu \models \Phi(x_1,\dots,x_n) \quad\iff\quad \bigcap_{1\le i\le n} I_i \neq \emptyset\,.$$

Applying Lemma 3.18, we obtain

$$\begin{aligned} M,\nu \models \Phi \quad &\iff \quad \text{for all } 1 \le i,j \le n, I_i \cap I_j \neq \emptyset \\ &\iff \quad \text{for all } 1 \le i,j \le n, (\nu(x_i),\nu(x_j)) \in [\![\pi_i \cdot \{\psi\}? \cdot \pi_j^{-1}]\!] \\ &\iff \quad M,\nu \models \Psi\,. \end{aligned}$$

□

Translation from FO[Σ] to $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$. We are now ready to give the proof of Theorem 3.14.

Proof of Theorem 3.14. We assume that $\Phi \in \mathrm{FO}[\Sigma]$ is in prenex normal form, and prove the result by induction. The translation of atomic formulas $\alpha(x,y)$ is straightforward; moreover, $P(x) \equiv \widetilde{\{P\}?}(x,x)$, and $(x = y) \equiv \widetilde{\{true\}?}(x,y)$. Using Lemma 3.17 to eliminate negations, we obtain the result for all quantifier-free formulas.

The case $\Phi = \forall x.\Psi \equiv \neg\exists x.\neg\Psi$ reduces to the case of existential quantification, applying again Lemma 3.17 to eliminate negations.

We are left with the case $\Phi = \exists x.\Psi$. If x is not free in Ψ, then $\Phi \equiv \Psi$ (since Ψ has at least one free variable) and we are done by induction. Otherwise, assume that $\mathrm{Free}(\Psi) = \{x_1, \ldots, x_n\}$ with $n > 1$ and $x = x_n$. By induction, Ψ is equivalent to a positive boolean combination of formulas of the form $\widetilde{\pi}(x_i, x_j)$ with $\pi \in \mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$. We replace $\widetilde{\pi}(x_i, x_j)$ with $\widetilde{\pi^{-1}}(x_j, x_i)$ whenever $j < i$, and bring the resulting formula into disjunctive normal form. Each conjunct is then of the form $\Upsilon = \Upsilon_1 \wedge \Upsilon_2 \wedge \Upsilon_3$, where Υ_1 uses only variables from $\{x_1, \ldots, x_{n-1}\}$, $\Upsilon_2 = \bigwedge_i \widetilde{\pi}_i(y_i, x)$ with $y_i = x_j$ for some $1 \leq j < n$, and $\Upsilon_3 = \bigwedge_j \widetilde{\pi}_j(x, x)$. Note that $\Upsilon_3 \equiv \widetilde{\varphi}(x)$, where $\varphi = \bigwedge_j \mathsf{Loop}(\pi_j)$. Then $\exists x.\Psi$ is equivalent to a finite disjunction of formulas

$$\exists x.\Upsilon \;\equiv\; \Upsilon_1 \wedge \exists x.\,(\Upsilon_2 \wedge \widetilde{\varphi}(x))$$

with Υ_1 and Υ_2 as above. If Υ_2 is empty, then we replace $\exists x.\widetilde{\varphi}(x)$ with the formula

$$(\leq \cdot\, \{\varphi\}? \cdot \geq)(x_1, x_1) \vee (\geq \cdot\, \{\varphi\}? \cdot \leq)(x_1, x_1)\,.$$

Otherwise, we apply Lemma 3.19 to $\exists x.\,(\Upsilon_2 \wedge \widetilde{\varphi}(x))$. In all cases, we obtain an equivalent formula which is a finite positive boolean combination of formulas $\widetilde{\pi}(x_i, x_j)$ with $1 \leq i, j < n$ and $\pi \in \mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$. □

3.7 Applications

Various classes of structures studied in the literature are interval-preserving, or easily reducible to interval-preserving structures.

Linear orders. It follows from Theorem 3.16 that any class of linear orders has the 3-variable property. This was initially proven in [52] using Ehrenfeucht-Fraïssé games. The case of formulas with one free variable is also implied by Kamp's theorem [54, 32] stating that LTL is expressively complete for first-order logic.

Real-time systems. Let $\mathbb{A} \subseteq \mathbb{R}$ (for instance, $\mathbb{A} = \{1\}, \mathbb{N}, \mathbb{Q}$). The structure $(\mathbb{R}, <, +\mathbb{A}) = (\mathbb{R}, <, (\{(r, r+a) \mid r \in \mathbb{R}\})_{a \in \mathbb{A}})$ has the 3-variable property. This gives a new proof of a result from [3], which was again proven through Ehrenfeucht-Fraïssé games.

Polynomials over $\mathbb{R}$. Theorem 3.16 also allows us to answer an open question from [3], namely, whether structures over the real numbers with polynomial functions have the 3-variable property:

Suppose that $\mathsf{Rel} = \{\leq\} \cup \mathcal{P}$, where $\mathcal{P}$ is a set of polynomials $p : \mathbb{R} \to \mathbb{R}$. Let $M_{\mathcal{P}}$ be the $(\emptyset, \mathsf{Rel})$-structure $(\mathbb{R}, \leq, (p^{M_{\mathcal{P}}})_{p \in \mathcal{P}})$, where $\leq$ is the usual ordering of the real numbers, and $p^{M_{\mathcal{P}}} = \{(x, p(x)) \mid x \in \mathbb{R}\}$ for all $p \in \mathcal{P}$. Given an interpretation $h : \mathsf{Prop} \to 2^{\mathbb{R}}$ of the unary predicates, we denote by $(M_{\mathcal{P}}, h)$ the Σ-structure

$(\mathbb{R}, \leq, (p^{M_\mathcal{P}})_{p\in\mathcal{P}}, (h(P))_{P\in\mathsf{Prop}})$. We also denote by $\mathcal{C}_\mathbb{R}[\Sigma]$ the set of all structures $(M_\mathcal{P}, h)$.

Theorem 3.20. *$\mathcal{C}_\mathbb{R}[\Sigma]$ has the strong 3-variable property, i.e., any $\mathrm{FO}[\Sigma]$ formula is equivalent over $\mathcal{C}_\mathbb{R}[\Sigma]$ to a finite boolean combination of formulas in $\mathrm{FO}^3[\Sigma]$.*

Proof. The idea is to decompose each polynomial function into a finite number of monotone (and thus, interval-preserving) partial functions.

Let $p \in \mathcal{P}$, and $m_1 < \cdots < m_n$ its local extrema. We denote by $p_{(-\infty,m_1)}$, $p_{[m_1,m_2)}, \ldots, p_{[m_n,+\infty)}$ the (monotone) restrictions of p to intervals delimitated by its local extrema, and $\mathcal{P}'_p$ the set of these partial functions. Let $\mathcal{P}' = \bigcup_{p\in\mathcal{P}} \mathcal{P}'_p$, and $\Sigma' = (\mathsf{Prop}, \{\leq\} \uplus \mathcal{P}')$. Similarly to the definition of the structure $M_\mathcal{P}$, we let $M_{\mathcal{P}'} = (\mathbb{R}, \leq, ({p_I}^{M_{\mathcal{P}'}})_{p_I\in\mathcal{P}'})$, where $\leq$ is the usual ordering of the real numbers, and ${p_I}^{M_{\mathcal{P}'}} = \{(x, p(x)) \mid x \in I\}$. We denote by $\mathcal{C}_\mathbb{R}[\Sigma']$ the set of Σ'-structures $(M_{\mathcal{P}'}, h)$ with $h : \mathsf{Prop} \to 2^\mathbb{R}$ (defined analogously to $(M_\mathcal{P}, h)$). Note that all structures in $\mathcal{C}_\mathbb{R}[\Sigma']$ are interval-preserving.

We say that two formulas $\Phi \in \mathrm{FO}[\Sigma]$ and $\Psi \in \mathrm{FO}[\Sigma']$ with free variables included in a set V are equivalent, written $\Phi \equiv \Psi$, when for all $h : \mathsf{Prop} \to 2^\mathbb{R}$ and $\nu : V \to \mathbb{R}$, we have $(M_\mathcal{P}, h), \nu \models \Phi$ if and only if $(M_{\mathcal{P}'}, h), \nu \models \Psi$.

Let $\Phi \in \mathrm{FO}[\Sigma]$. By definition of the formulas p_I, the formula $\Psi \in \mathrm{FO}[\Sigma']$ obtained by replacing each atomic formula $p(x, y)$ by $\bigvee_{p_I\in\mathcal{P}'_p} p_I(x, y)$ is equivalent to Φ. Applying Theorem 3.16 to Ψ, we obtain another formula $\Psi' \in \mathrm{FO}[\Sigma']$ such that $\Psi' \equiv_{\mathcal{C}_\mathbb{R}[\Sigma']} \Psi$ and Ψ' is a finite boolean combination of formulas in $\mathrm{FO}^3[\Sigma']$.

Following Example 2.3, one can construct for each $p_I \in \mathcal{P}'$ a formula "$x \in I$" of $\mathrm{FO}^3[\Sigma]$ such that $(M_\mathcal{P}, h), \nu \models x \in I$ if and only if $\nu(x) \in I$. Consider now the formula $\Phi' \in \mathrm{FO}[\Sigma]$ obtained by replacing each atomic formula $p_I(x, y)$ in Ψ' by $x \in I \wedge p(x, y)$. Then $\Phi' \equiv \Psi'$, hence $\Phi \equiv_{\mathcal{C}_\mathbb{R}[\Sigma]} \Phi'$. Moreover, Φ' is a finite boolean combination of formulas in $\mathrm{FO}^3[\Sigma]$. □

Mazurkiewicz traces. Mazurkiewicz traces [24] are a common way to describe executions of concurrent systems. We assume a finite alphabet A, and a reflexive and symmetric relation $D \subseteq A\times A$ called the dependence relation. A *(Mazurkiewicz) trace* over (A, D) is a labeled partial order $(E, \leq, \lambda)$ such that

- E is a finite or countably infinite set.
- $\leq \subseteq E \times E$ is a partial order such that for all $e \in E$, $\{f \in E \mid f \leq e\}$ is finite. We let $<$ denote the strict part of $\leq$, and write $e \lessdot f$ if $e < f$ and there are no f' with $e < f' < f$.
- $\lambda : E \to A$ is the labeling function.
- for all $e, f \in E$ such that $(\lambda(e), \lambda(f)) \in D$, we have $e \leq f$ or $f \leq e$.

- for all $e, f \in E$ such that $e \lessdot f$, $(\lambda(e), \lambda(f)) \in D$.

Let $\Sigma = (A, \leq)$ be the signature associated with traces over (A, D). It was proven in [23], as a corollary of the expressive completeness of an extension of LTL to traces, that every *sentence* in $\mathrm{FO}[\Sigma]$ is equivalent, over traces, to a sentence in $\mathrm{FO}^3[\Sigma]$. Using Theorem 3.16, we can give a new proof of this, and in fact, show a slightly stronger result, where we also consider formulas with free variables.

Traces are partially ordered rather than linearly ordered, so we cannot directly apply our results about interval-preserving structures. However, like in the example of polynomial functions, we can define an alternative signature Σ' and a faithful description of traces as interval-preserving Σ'-structures that allow us to apply Theorem 3.16.

Theorem 3.21. *The class of traces over (A, D) has the strong 3-variable property: any $\mathrm{FO}[\Sigma]$ formula is equivalent, over traces, to a finite boolean combination of formulas in $\mathrm{FO}^3[\Sigma]$.*

Proof. For a trace $(E, \leq, \lambda)$ and $a, b \in A$, we denote by $\leq_{a,b}$ the restriction of $\leq$ to events labeled a and b, respectively:

$$e \leq_{a,b} f \qquad \text{if} \qquad e \leq f \text{ and } \lambda(e) = a \text{ and } \lambda(f) = b\,.$$

We also define a linear order $\sqsubseteq \subseteq E \times E$ as follows: we fix a linear order $\sqsubseteq \subseteq A \times A$, and we let

$$e \sqsubseteq f \qquad \text{if} \qquad \lambda(e) \sqsubset \lambda(f) \text{ or } (\lambda(e) = \lambda(f) \wedge e \leq f)\,.$$

This is indeed a linear order over A, since for all e, f such that $\lambda(e) = \lambda(f)$, we have $(\lambda(e), \lambda(f)) \in D$, which implies that $e \leq f$ or $f \leq e$.

We let $\Sigma' = (A, \{\sqsubseteq\} \cup \{\leq_{a,b} \mid a, b \in A\})$.

Claim 3.22. For all traces $M = (E, \leq, \lambda)$, the Σ'-structure induced by M is interval-preserving (with respect to $\sqsubseteq$).

Proof. Let $a, b \in A$, and I be an interval of $(E, \sqsubseteq)$. Let us show that $[\![\leq_{a,b}]\!](I)$ is an interval of $([\![\langle \leq_{a,b}^{-1} \rangle]\!], \sqsubseteq)$, the symmetric condition is proven similarly. So let $f_1, f_2 \in [\![\leq_{a,b}]\!](I)$, and let $f \in [\![\langle \leq_{a,b}^{-1} \rangle]\!]$ such that $f_1 \sqsubseteq f \sqsubseteq f_2$. Let us show that for $e_1 \in I$ such that $e_1 \leq_{a,b} f_1$, we also have $e_1 \leq_{a,b} f$. By definition of $\leq_{a,b}$, we have $\lambda(e_1) = a$ and $\lambda(f) = \lambda(f_1) = b$. In particular, $f_1 \sqsubseteq f$ implies $f_1 \leq f$. By transitivity of $\leq$, we obtain $e_1 \leq f$, i.e., $e_1 \leq_{a,b} f$. □

Let $\Phi \in \mathrm{FO}[\Sigma]$, and let $\Phi' \in \mathrm{FO}[\Sigma']$ be the formula obtained from Φ by replacing every formula $x \leq y$ by $\bigvee_{a,b \in A} x \leq_{a,b} y$. By Theorem 3.16 and Claim 3.22, there exists a finite boolean combination $\Psi' \in \mathrm{FO}[\Sigma']$ of $\mathrm{FO}^3[\Sigma']$ formulas such that Φ'

and Ψ' are equivalent over traces. Define $\Psi \in \mathrm{FO}[\Sigma]$ as the formula obtained from Ψ' by replacing every formula $x \sqsubseteq y$ with

$$\bigvee_{a \sqsubset b} a(x) \wedge b(y) \vee \bigvee_{a \in A} a(x) \wedge a(y) \wedge a \leq y \,,$$

and every formula $x \leq_{a,b} y$ with $x \leq y \wedge a(x) \wedge b(y)$. Then Ψ is a finite boolean combination of $\mathrm{FO}^3[\Sigma]$ formulas, and Φ and Ψ are equivalent over traces. □

Theorem 3.21 is in fact true for all labeled partial orders such that: (i) the set of labels is finite, and (ii) for every label a, the restriction of the partial order to a-labeled elements is linear. These are called pomsets without auto-concurrency in [26], semi-words in [79, 22], P-traces in [5].

Message Sequence Charts. In Chapter 5, we will see that MSCs, which represent behaviors of message-passing systems, can also be interpreted as interval-preserving structures. Like traces, MSCs are labeled partial orders in which elements with a same label are totally ordered, but in addition, they are also equipped with another binary relation called the message relation.

3.8 The case of complete linear orders

In this section, we show that over *complete* linear orders, we can further restrict the use of complement, while keeping the full expressive power of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ (i.e., of $\mathrm{FO}[\Sigma]$). This results in a simpler syntax, closer to known temporal logics, and without an explicit complement operation. This is interesting from an algorithmic point of view, as the complement operator is usually a source of high complexities. To better understand this, we can look at the example of words. Over words, star-free regular expressions and LTL are both known to have the same expressive power as first-order logic. However, the complexity of translating a star-free regular expression into finite automata is non-elementary [80], while it is PSPACE for LTL [78, 91]. The non-elementary complexity for star-free regular expressions comes from the use of a complement operation, which is similar to the complement of path formulas in star-free PDL. On the other hand, while negation is also allowed in LTL, it is closer to negation at the level of *event* formulas in star-free PDL, in the sense that both are applied to unary, rather than binary, predicates. Our aim in this section is to define a fragment of star-free PDL which is still expressively equivalent to first-order logic (over complete linear orders), but somewhat closer in its syntax to LTL.

This section is largely independent from the remainder of the thesis, and not required to understand subsequent chapters. We will later define a similar fragment when studying the special case of MSCs, and the present section highlights the fact

that the equivalence of star-free PDL and its complement-free variant can be proven in a more general setting. However, as the proofs are much simpler for MSCs, we will not need to rely on the results presented here.

Definition 3.23. Let $\Sigma = (\mathsf{Prop}, \mathsf{Rel})$ be a signature with $\leq \in \mathsf{Rel}$. A Σ-structure $M = (E^M, (P^M)_{P\in\mathsf{Prop}}, (\alpha^M)_{\alpha\in\mathsf{Rel}})$ is *complete* if $\leq^M$ is a complete linear order, i.e., a linear order such that any nonempty subset $F \subseteq E^M$ with an upper bound has a least upper bound $\sup F$. *complete* Σ*-structure*

Note that in a complete Σ-structure M, every nonempty subset $F \subseteq E^M$ with a lower bound also has a greatest lower bound $\inf F$.

Example 3.24. The structure $(\mathbb{R}, \leq, +\mathbb{A})$ for $\mathbb{A} \subseteq \mathbb{R}$ is a complete Σ-structure. Discrete Σ-structures, where every element has a successor and a predecessor, are also complete.

3.8.1 A fragment of $\mathsf{PDL}_{\mathsf{sf}}[\Sigma]$ without complement

We start by defining a fragment $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ without a complement operator, but with additional "atomic" path formulas.

Path formulas for Until and Since. We define two more path formulas, $<_\varphi$ and $>_\varphi$, similar to LTL strict until and strict since modalities, with the following semantics. Given an event formula $\varphi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ and a Σ-structure M, $<_\varphi, >_\varphi$

$$[\![<_\varphi]\!]^M = \left\{(e,f) \in E^M \times E^M \mid e < f \wedge \forall g \in E^M.\, \Big(e < g < f \implies g \in [\![\varphi]\!]^M\Big)\right\}$$
$$[\![>_\varphi]\!]^M = \left\{(e,f) \in E^M \times E^M \mid f < e \wedge \forall g \in E^M.\, \Big(f < g < e \implies g \in [\![\varphi]\!]^M\Big)\right\}.$$

Note that $>_\varphi \equiv (<_\varphi)^{-1}$. The formulas $<_\varphi$ and $>_\varphi$ are easily definable in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$:

$$<_\varphi \equiv < \cap (< \cdot \{\neg\varphi\}? \cdot <)^{\mathsf{c}}$$
$$>_\varphi \equiv > \cap (> \cdot \{\neg\varphi\}? \cdot >)^{\mathsf{c}}.$$

Remark 3.25. The usual LTL strict until SU and strict since SS modalities can be defined as

$$\varphi\,\mathsf{SU}\,\psi \equiv \langle <_\varphi\rangle\,\psi \qquad \text{and} \qquad \varphi\,\mathsf{SS}\,\psi \equiv \langle >_\varphi\rangle\,\psi\,.$$

For the non-strict versions,

$$\varphi\,\mathsf{U}\,\psi \equiv \psi \vee (\varphi \wedge \langle <_\varphi\rangle\,\psi) \qquad \text{and} \qquad \varphi\,\mathsf{S}\,\psi \equiv \psi \vee (\varphi \wedge \langle >_\varphi\rangle\,\psi)\,.$$

Sup and inf of a path formula. For all $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ we introduce path formulas $\mathsf{sup}\ \pi$ and $\mathsf{inf}\ \pi$ with

sup π, inf π

$$[\![\mathsf{sup}\ \pi]\!]^M = \{(e,f) \in E^M \times E^M \mid f = \sup [\![\pi]\!]^M(e)\}$$
$$[\![\mathsf{inf}\ \pi]\!]^M = \{(e,f) \in E^M \times E^M \mid f = \inf [\![\pi]\!]^M(e)\}\,.$$

In particular, $[\![\mathsf{sup}\ \pi]\!](e)$ is non-empty if and only if $\sup[\![\pi]\!](e)$ is well-defined, in which case it is a singleton (and similarly for $\mathsf{inf}\ \pi$).

The formulas $\mathsf{sup}\ \pi$ and $\mathsf{inf}\ \pi$ could be defined as follows in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$:

$$\mathsf{sup}\ \pi \equiv (\pi \cdot >)^{\mathsf{c}} \cap ((\pi \cdot >)^{\mathsf{c}} \cdot <)^{\mathsf{c}}$$
$$\mathsf{inf}\ \pi \equiv (\pi \cdot <)^{\mathsf{c}} \cap ((\pi \cdot <)^{\mathsf{c}} \cdot >)^{\mathsf{c}}\,.$$

Intuitively, the formula $(\pi \cdot >)^{\mathsf{c}}$ relates e to upper bounds of $[\![\pi]\!](e)$, and $((\pi \cdot >)^{\mathsf{c}} \cdot <)^{\mathsf{c}}$ selects the minimal element among these lower bounds. The case of $\mathsf{inf}\ \pi$ is similar.

Syntax of $\mathbf{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$. In $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$, we do not allow any complement operators, but we allow formulas $<_\varphi$ and $>_\varphi$, and a very restricted use of inf and sup: they appear only in formulas $\mathsf{inf}\ (\lambda \cdot \alpha)$ or $\mathsf{sup}\ (\lambda \cdot \alpha)$ where $\alpha \in \mathsf{Rel}$ and λ is an order formula in $\{\leq, <, \geq, >\}$. We can see these $\mathsf{inf}\ (\lambda \cdot \alpha)$ and $\mathsf{sup}\ (\lambda \cdot \alpha)$ formulas as additional atomic formulas, defined only from the relations in Rel. We also restrict the use of the converse operation to "atomic" formulas, which can already be done in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ or $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$.

$\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$

Definition 3.26. The set of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ formulas is defined as follows:

$$\begin{aligned}
\xi &::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi \\
\varphi &::= \mathit{true} \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi) \\
\sigma &::= \alpha \mid \mathsf{inf}\ (\lambda \cdot \alpha) \mid \mathsf{sup}\ (\lambda \cdot \alpha) \\
\pi &::= \sigma \mid \sigma^{-1} \mid \{\varphi\}? \mid <_\varphi \mid >_\varphi \mid \pi \cdot \pi\,,
\end{aligned}$$

where $P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel} \setminus \{\leq\}$, $\lambda \in \{\leq, <, \geq, >\}$.

Since all operators of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ are definable in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, we can see the logic $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ as a fragment of $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, and we sometimes identify $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ formulas with their equivalent in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$. So we may talk for instance about $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ formulas π^{c} or $\pi_1 + \pi_2$ for $\pi, \pi_1, \pi_2 \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$.

Macros. Note that we can also define $<$ and $>$ in $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$:

$$< \equiv <_{\mathit{true}} \qquad > \equiv >_{\mathit{true}}\,.$$

And while $\leq$, $\geq$ and $\top$ are not formulas of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$, they are equivalent to finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Finally, π^{-1} is also definable in $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$. For simplicity, we do not distinguish between π^{-1} (which is syntactically in $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ but not in $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$) and the equivalent formula in $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$.

Lemma 3.27. *For all $\pi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$, there exists $\pi' \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ such that $\pi^{-1} \equiv \pi'$.*

Proof. We can define π' inductively using the following equivalences:

$$(\pi_1^{-1})^{-1} \equiv \pi_1 \qquad \{\varphi\}?^{-1} \equiv \{\varphi\}? \qquad <_\varphi^{-1} \equiv >_\varphi \qquad (\pi_1 \cdot \pi_2)^{-1} \equiv \pi_2^{-1} \cdot \pi_1^{-1}\,. \quad \square$$

Properties of $\mathbf{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$. An important remark is that all $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas are interval-preserving. In fact, the new formulas $\mathsf{inf}\ (\lambda \cdot \alpha)$ or $\mathsf{sup}\ (\lambda \cdot \alpha)$ satisfy a stronger property: they define non-increasing or non-decreasing functions from an interval of E^M to E^M.

Lemma 3.28. *Let $\pi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ of the form $\pi = \mathsf{sup}\ (\lambda \cdot \alpha)$ or $\pi = \mathsf{inf}\ (\lambda \cdot \alpha)$, where $\alpha \in \mathsf{Rel} \setminus \{\leq\}$ and $\lambda \in \{\leq, <, \geq, >\}$. In any Σ-structure M,*

1. *If M is complete, then the set $[\![\langle\pi\rangle]\!]$ is an interval of $(E^M, \leq^M)$.*
2. *If π is of the form*

   $$\pi = \mathsf{sup}\ (\geq \cdot\, \alpha)\,, \quad \pi = \mathsf{sup}\ (> \cdot\, \alpha)\,, \quad \pi = \mathsf{inf}\ (\leq \cdot\, \alpha)\,, \text{or} \quad \pi = \mathsf{inf}\ (< \cdot\, \alpha)\,,$$

   then for all $(e_1, f_1), (e_2, f_2) \in [\![\pi]\!]$ such that $e_1 \leq e_2$, we have $f_1 \leq f_2$.
3. *If π is of the form*

   $$\pi = \mathsf{sup}\ (\leq \cdot\, \alpha)\,, \quad \pi = \mathsf{sup}\ (< \cdot\, \alpha)\,, \quad \pi = \mathsf{inf}\ (\geq \cdot\, \alpha)\,, \text{or} \quad \pi = \mathsf{inf}\ (> \cdot\, \alpha)\,,$$

   then for all $(e_1, f_1), (e_2, f_2) \in [\![\pi]\!]$ such that $e_1 \leq e_2$, we have $f_2 \leq f_1$.

Proof. We distinguish four cases.

Case 1: suppose that $\pi = \mathsf{sup}\ (\lambda \cdot \alpha)$ for $\lambda \in \{\geq, >\}$.

Let $(e_1, f_1), (e_2, f_2) \in [\![\mathsf{sup}\ (\lambda \cdot \alpha)]\!]$ such that $e_1 \leq e_2$. We have $f_1 = \sup[\![\lambda \cdot \alpha]\!](e_1)$ and $f_2 = \sup[\![\lambda \cdot \alpha]\!](e_2)$. Since

$$[\![\lambda \cdot \alpha]\!](e_1) \subseteq [\![\lambda \cdot \alpha]\!](e_2) \quad \text{and} \quad [\![\lambda \cdot \alpha]\!](e_2) \leq f_2\,,$$

we have

$$[\![\lambda \cdot \alpha]\!](e_1) \leq f_2\,, \quad \text{i.e.,} \quad f_1 \leq f_2\,.$$

This proves property 2.

Similarly, for all $e_1 \leq e \leq e_2$, we have

$$[\![\lambda \cdot \alpha]\!](e_1) \subseteq [\![\lambda \cdot \alpha]\!](e) \subseteq [\![\lambda \cdot \alpha]\!](e_2)\,.$$

In particular, $[\![\lambda \cdot \alpha]\!](e) \leq f_2$. If, in addition, $[\![\lambda \cdot \alpha]\!](e) \neq \emptyset$, then if M is a complete Σ-structure, $\sup[\![\lambda \cdot \alpha]\!](e)$ is well-defined. Otherwise, if $[\![\lambda \cdot \alpha]\!](e) = \emptyset$, then we also have $[\![\lambda \cdot \alpha]\!](e_1) = \emptyset$, and

$$\sup[\![\lambda \cdot \alpha]\!](e) = \sup \emptyset = \sup[\![\lambda \cdot \alpha]\!](e_1) = f_1\,.$$

In any case, $\sup[\![\lambda \cdot \alpha]\!](e)$ is well-defined, i.e., $e \in [\![\langle\pi\rangle]\!]$. This proves property 1.

Case 2: suppose that $\pi = \mathsf{inf}\ (\lambda \cdot \alpha)$ for $\lambda \in \{\geq, >\}$. Similarly to the previous case, for all $(e_1, f_1), (e_2, f_2) \in [\![\mathsf{inf}\ (\lambda \cdot \alpha)]\!]$ such that $e_1 \leq e_2$, we have

$$[\![\lambda \cdot \alpha]\!](e_1) \subseteq [\![\lambda \cdot \alpha]\!](e_2) \implies f_2 \leq [\![\lambda \cdot \alpha]\!](e_1)$$

hence $f_2 \leq f_1$. This proves property 3. If M is complete, then for all $e_1 \leq e \leq e_2$,

- $f_2 \leq [\![\lambda \cdot \alpha]\!](e)$, therefore, if $[\![\lambda \cdot \alpha]\!](e) \neq \emptyset$, $\inf[\![\lambda \cdot \alpha]\!](e)$ is well-defined;
- If $[\![\lambda \cdot \alpha]\!](e) = \emptyset$, then $\inf[\![\lambda \cdot \alpha]\!](e) = \inf \emptyset = \inf[\![\lambda \cdot \alpha]\!](e_1) = f_1$.

Thus, $e \in [\![\langle\pi\rangle]\!]$, which proves property 1.

Case 3: suppose that $\pi = \mathsf{sup}\ (\lambda \cdot \alpha)$ for $\lambda \in \{\leq, <\}$. For all $(e_1, f_1), (e_2, f_2) \in [\![\mathsf{sup}\ (\lambda \cdot \alpha)]\!]$ such that $e_1 \leq e_2$, we have

$$[\![\lambda \cdot \alpha]\!](e_2) \subseteq [\![\lambda \cdot \alpha]\!](e_1) \implies [\![\lambda \cdot \alpha]\!](e_2) \leq f_1\,,$$

hence $f_2 \leq f_1$. This proves property 3. If M is complete, then for all $e_1 \leq e \leq e_2$,

- $[\![\lambda \cdot \alpha]\!](e) \leq f_1$, therefore, if $[\![\lambda \cdot \alpha]\!](e) \neq \emptyset$, $\sup[\![\lambda \cdot \alpha]\!](e)$ is well-defined;
- If $[\![\lambda \cdot \alpha]\!](e) = \emptyset$, then $\sup[\![\lambda \cdot \alpha]\!](e) = \sup \emptyset = \sup[\![\lambda \cdot \alpha]\!](e_2) = f_2$.

Thus, $e \in [\![\langle\pi\rangle]\!]$, which proves property 1.

Case 4: suppose that $\pi = \mathsf{inf}\ (\lambda \cdot \alpha)$ for $\lambda \in \{\leq, <\}$. For all $(e_1, f_1), (e_2, f_2) \in [\![\mathsf{inf}\ (\lambda \cdot \alpha)]\!]$ such that $e_1 \leq e_2$, we have

$$[\![\lambda \cdot \alpha]\!](e_2) \subseteq [\![\lambda \cdot \alpha]\!](e_1) \implies f_1 \leq [\![\lambda \cdot \alpha]\!](e_2)\,,$$

hence $f_1 \leq f_2$. This proves property 2. If M is complete, then for all $e_1 \leq e \leq e_2$,

- $f_1 \leq [\![\lambda \cdot \alpha]\!](e)$, therefore, if $[\![\lambda \cdot \alpha]\!](e) \neq \emptyset$, $\inf[\![\lambda \cdot \alpha]\!](e)$ is well-defined;

- If $[\![\lambda \cdot \alpha]\!](e) = \emptyset$, then $\inf [\![\lambda \cdot \alpha]\!](e) = \inf \emptyset = \inf [\![\lambda \cdot \alpha]\!](e_2) = f_2$.

Thus, $e \in [\![\langle \pi \rangle]\!]$, which proves property 1. □

Lemma 3.29. *All path formulas $\pi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ are interval-preserving.*

Proof. Recall that

$$<_\varphi \equiv < \cap (< \cdot \{\neg\varphi\}? \cdot <)^{\mathsf{c}}$$

Moreover, $<$ is interval-preserving, and by Lemma 3.9, so is $(< \cdot \{\neg\varphi\}? \cdot <)^{\mathsf{c}} \equiv (\leq \cdot (< \cdot \{\neg\varphi\}? \cdot <) \cdot \leq)^{\mathsf{c}}$. Therefore, by Lemma 3.7, $<_\varphi$ and $>_\varphi \equiv <_\varphi^{-1}$ are also interval-preserving.

In addition, by Lemma 3.28, for all $\lambda \in \{\leq, <, \geq, >\}$ and $\alpha \in \mathsf{Rel} \setminus \{\leq\}$, the formulas $\mathsf{sup}\ (\lambda \cdot \alpha)$ and $\mathsf{inf}\ (\lambda \cdot \alpha)$ define non-increasing or non-decreasing partial functions. Therefore, they are interval-preserving.

Since interval-preserving formulas are closed under concatenation and converse (Lemma 3.7), all path formulas $\pi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ are thus interval-preserving. □

3.8.2 Main result

The aim of this section is to show that over the class of complete and interval-preserving Σ-structures, $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ and $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ are expressively equivalent:

Theorem 3.30. *Over complete interval-preserving Σ-structures,*

1. *Any $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ sentence is equivalent to a $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ sentence.*
2. *Any $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ event formula is equivalent to a $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ event formula.*
3. *Any $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ path formula is equivalent to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.*

By Theorem 3.14, this also implies that over complete and interval-preserving Σ-structures, $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ has the same expressive power as $\mathrm{FO}[\Sigma]$.

Remark 3.31. For many classes of complete interval-preserving Σ-structures, it is possible to also remove the $\mathsf{inf}\ (\lambda \cdot \alpha)$ and $\mathsf{sup}\ (\lambda \cdot \alpha)$ formulas from the syntax of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ without losing in expressivity. This is for instance the case over structures $(\mathbb{R}, <, +\mathbb{A})$, since

$$\mathsf{inf}\ (\leq \cdot +a) \equiv \mathsf{inf}\ (< \cdot +a) \equiv \mathsf{sup}\ (\geq \cdot +a) \equiv \mathsf{sup}\ (> \cdot +a) \equiv +a$$
$$\mathsf{inf}\ (\geq \cdot +a) \equiv \mathsf{inf}\ (> \cdot +a) \equiv \mathsf{sup}\ (\leq \cdot +a) \equiv \mathsf{sup}\ (< \cdot +a) \equiv \{\mathit{false}\}? \,.$$

More generally, formulas $\mathsf{inf}\ (\lambda \cdot \alpha)$ and $\mathsf{sup}\ (\lambda \cdot \alpha)$ can be eliminated over structures where $[\![\alpha]\!]$ defines a total increasing or decreasing bijection (and in fact, one can show that every interval-preserving bijection is either increasing or decreasing). They are also unnecessary over message sequence charts (see Chapter 5).

Consequences of Theorem 3.30. Theorem 3.30 provides a logic which is also expressively equivalent to FO[Σ] over complete interval-preserving Σ-structures, but which has better algorithmic properties than $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$. Of course, most decision problems for FO[Σ], and therefore $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ or $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$, are undecidable. However, for specific classes of structures and problems for which this is not the case, having a logic without a complement operator results in better complexities. We discussed the case of words at the beginning of the section, and Chapter 5 and 6 provide further examples of this.

$\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ is also closer to classic temporal logics. Recall that $\langle <_\varphi \rangle\, \psi$ corresponds to a strict until formula $\varphi\, \mathsf{SU}\, \psi$, and $\langle >_\varphi \rangle\, \psi$ to the strict since $\varphi\, \mathsf{SS}\, \psi$. In addition, formulas $\langle \pi_1 \cdot \pi_2 \rangle\, \varphi$ can be unfolded as $\langle \pi_1 \cdot \pi_2 \rangle\, \varphi \equiv \langle \pi_1 \rangle\, (\langle \pi_2 \rangle\, \varphi)$. Combining these observations, the syntax of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ event formulas could be equivalently described as

$$\varphi ::= \mathit{true} \mid P \mid \varphi \vee \varphi \mid \varphi\, \mathsf{SU}\, \psi \mid \varphi\, \mathsf{SS}\, \psi \mid \neg\varphi \mid \langle \sigma \rangle\, \varphi \mid \langle \sigma^{-1} \rangle\, \varphi \mid \mathsf{Loop}(\pi)\,,$$

and seen as a (one-dimensional) temporal logic with an infinite number of $\mathsf{Loop}(\pi)$ modalities (on the other hand, if Rel is finite, we only need a finite number of $\langle \sigma \rangle$ modalities).

Given a class $\mathcal{C}$ of complete interval-preserving Σ-structures, the problem of the existence of an expressively complete temporal logic over $\mathcal{C}$, with a finite set of FO-definable modalities, can then be reformulated as whether it is possible to bound the length of π (that is, the number of top-level concatenations) in $\mathsf{Loop}(\pi)$ formulas.

Example 3.32. Over $(\mathbb{R}, \leq, +\mathbb{Q})$, all formulas of Metric Temporal Logic (MTL) (cf. Example 3.1) can be defined in $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ using loop formulas of length at most 4. For instance,

$$\phi\, \mathsf{U}_{(q,r)}\, \psi \equiv \mathsf{Loop}(<_\varphi \cdot \{\varphi\}? \cdot (+q)^{-1}) \wedge \langle +q \rangle\, \mathsf{Loop}(<_\varphi \cdot \{\psi\}? \cdot <_{\mathit{true}} \cdot +(r-q)^{-1})\,.$$

MTL was shown to be expressively complete for first-order logic [49, 48]. Similar arguments could be applied to do the proof at the level of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ instead of working directly with first-order formulas.

On the other hand, there are classes of structures for which the length of $\mathsf{Loop}(\pi)$ formulas cannot be bounded. For instance, over $(\mathbb{R}, \leq, +\mathbb{Z})$, no temporal logic with modalities definable by first-order formulas of bounded quantifier depth can be expressively complete for first-order logic [45]. To regain expressive completeness, one has to add to MTL an infinite set of counting modalities $C_n(\varphi)$, which states that there are at least n positions satisfying φ in the next unit of time [48]. This could be written in $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ as

$$C_n(\varphi) \equiv \mathsf{Loop}\left(\leq \cdot (\{\varphi\}? \cdot <)^{n-1} \cdot \{\varphi\}? \cdot \leq \cdot (+1)^{-1}\right)\,,$$

that is, a loop formula of length $2n+2$.

Outline of the proof. The remainder of the section is devoted to the proof of Theorem 3.30. We focus on 3., which implies 2. and 1. In order to prove the result, we show that finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas are closed under the operations of $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$. In particular, we need to prove that for all finite unions π of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ paths formulas, and for all $\lambda, \mu \in \{\leq, \geq\}$, the formula $(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$ is equivalent, over complete interval-preserving structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas. This is proven by induction on π. Section 3.8.3 deals with the inductive cases $\pi = \pi_1 \cdot \pi_2$ and $\pi = \pi_1 + \pi_2$, and Section 3.8.4 with all the base cases. We put everything together in Section 3.8.5.

3.8.3 Splitting formulas with complement operators

The most important step towards proving Theorem 3.30 is to show that formulas of the form $(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$, where π is a concatenation of several path formulas, can be broken down into "smaller" formulas $(\lambda' \cdot \pi' \cdot \mu')^{\mathsf{c}}$.

Lemma 3.33. *For all interval-preserving path formulas $\pi_1, \pi_2 \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, over complete interval-preserving Σ-structures,*

$$\begin{aligned}(\pi_1 \cdot \pi_2)^{\mathsf{c}} \equiv\ & (\{\neg\langle \pi_1 \cdot \pi_2\rangle\}? \cdot \top) + (\top \cdot \{\neg\langle (\pi_1 \cdot \pi_2)^{-1}\rangle\}?) + \\ & (\pi_1 \cdot \{\langle \pi_2\rangle\}? \cdot >)^{\mathsf{c}} \cdot (\geq \cdot \{\langle \pi_1^{-1}\rangle\}? \cdot \pi_2)^{\mathsf{c}} + \\ & (\pi_1 \cdot \{\langle \pi_2\rangle\}? \cdot \geq)^{\mathsf{c}} \cdot (> \cdot \{\langle \pi_1^{-1}\rangle\}? \cdot \pi_2)^{\mathsf{c}} + \\ & (\pi_1 \cdot \{\langle \pi_2\rangle\}? \cdot <)^{\mathsf{c}} \cdot (\leq \cdot \{\langle \pi_1^{-1}\rangle\}? \cdot \pi_2)^{\mathsf{c}} + \\ & (\pi_1 \cdot \{\langle \pi_2\rangle\}? \cdot \leq)^{\mathsf{c}} \cdot (< \cdot \{\langle \pi_1^{-1}\rangle\}? \cdot \pi_2)^{\mathsf{c}}.\end{aligned}$$

Proof. First, notice that $(e, f) \in [\![\pi_1 \cdot \pi_2]\!]$ if and only if there exists $g \in [\![\pi_1\{\langle \pi_2\rangle\}?]\!](e)$ such that $g \in [\![\pi_2^{-1}\{\langle \pi_1^{-1}\rangle\}?]\!](f)$. Therefore,

$$(e, f) \in [\![(\pi_1 \cdot \pi_2)^{\mathsf{c}}]\!] \iff [\![\pi_1\{\langle \pi_2\rangle\}?]\!](e) \cap [\![\pi_2^{-1}\{\langle \pi_1^{-1}\rangle\}?]\!](f) = \emptyset\,.$$

We write $\overline{\pi}_1 = \pi_1\{\langle \pi_2\rangle\}?$, and $\overline{\pi}_2 = \pi_2^{-1}\{\langle \pi_1^{-1}\rangle\}?$. Note that $\overline{\pi}_1$ and $\overline{\pi}_2$ are interval-preserving. Moreover, $[\![\langle \overline{\pi}_1^{-1}\rangle]\!] = [\![\langle \pi_1^{-1}\rangle \wedge \langle \pi_2\rangle]\!] = [\![\langle \overline{\pi}_2^{-1}\rangle]\!]$. So $[\![\overline{\pi}_1]\!](e)$ and $[\![\overline{\pi}_2]\!](f)$ are both intervals of $([\![\langle \overline{\pi}_1^{-1}\rangle]\!], \leq)$. Thus,

$$[\![\overline{\pi}_1]\!](e) \cap [\![\overline{\pi}_2]\!](f) = \emptyset \iff \Big([\![\overline{\pi}_1]\!](e) < [\![\overline{\pi}_2]\!](f) \text{ or } [\![\overline{\pi}_2]\!](f) < [\![\overline{\pi}_1]\!](e)\Big)\,.$$

Now, in a complete Σ-structure, $[\![\overline{\pi}_1]\!](e) < [\![\overline{\pi}_2]\!](f)$ if and only if $[\![\overline{\pi}_1]\!](e) = \emptyset$, $[\![\overline{\pi}_2]\!](f) = \emptyset$, or there exists g (for instance, $g = \sup[\![\overline{\pi}_1]\!](e)$) such that

$$[\![\overline{\pi}_1]\!](e) \leq g < [\![\overline{\pi}_2]\!](f) \quad \text{or} \quad [\![\overline{\pi}_1]\!](e) < g \leq [\![\overline{\pi}_2]\!](f)\,,$$

i.e.,

$$g \in [\![(\overline{\pi}_1 \cdot >)^{\mathsf{c}}]\!](e) \cap [\![(\overline{\pi}_2 \cdot \leq)^{\mathsf{c}}]\!](f) \quad \text{or} \quad g \in [\![(\overline{\pi}_1 \cdot \geq)^{\mathsf{c}}]\!](e) \cap [\![(\overline{\pi}_2 \cdot <)^{\mathsf{c}}]\!](f)\,.$$

It follows that

$$[\![\overline{\pi}_1]\!](e) < [\![\overline{\pi}_2]\!](f) \iff (e,f) \in [\![\{\neg\langle\overline{\pi}_1\rangle\}? \cdot \top + \top \cdot \{\neg\langle\overline{\pi}_2\rangle\}? + \sigma]\!]\,,$$

where

$$\begin{aligned}
\sigma &= (\overline{\pi}_1 \cdot >)^{\mathsf{c}} \cdot ((\overline{\pi}_2 \cdot \leq)^{\mathsf{c}})^{-1} + (\overline{\pi}_1 \cdot \geq)^{\mathsf{c}} \cdot ((\overline{\pi}_2 \cdot <)^{\mathsf{c}})^{-1} \\
&= (\pi_1\{\langle\pi_2\rangle\}? \cdot >)^{\mathsf{c}} \cdot ((\pi_2^{-1}\{\langle\pi_1^{-1}\rangle\}? \cdot \leq)^{\mathsf{c}})^{-1} + \\
&\quad (\pi_1\{\langle\pi_2\rangle\}? \cdot \geq)^{\mathsf{c}} \cdot ((\pi_2^{-1}\{\langle\pi_1^{-1}\rangle\}? \cdot <)^{\mathsf{c}})^{-1} \\
&\equiv (\pi_1\{\langle\pi_2\rangle\}? \cdot >)^{\mathsf{c}} \cdot (\geq \cdot \{\langle\pi_1^{-1}\rangle\}? \cdot \pi_2)^{\mathsf{c}} + \\
&\quad (\pi_1\{\langle\pi_2\rangle\}? \cdot \geq)^{\mathsf{c}} \cdot (> \cdot \{\langle\pi_1^{-1}\rangle\}? \cdot \pi_2)^{\mathsf{c}}\,.
\end{aligned}$$

Similarly,

$$[\![\overline{\pi}_2]\!](f) < [\![\overline{\pi}_1]\!](e) \iff (e,f) \in [\![\{\neg\langle\overline{\pi}_1\rangle\}? \cdot \top + \top \cdot \{\neg\langle\overline{\pi}_2\rangle\}? + \sigma']\!]\,,$$

where

$$\begin{aligned}
\sigma' = (\pi_1 \cdot \{\langle\pi_2\rangle\}? \cdot <)^{\mathsf{c}} \cdot (\leq \cdot \{\langle\pi_1^{-1}\rangle\}? \cdot \pi_2)^{\mathsf{c}} + \\
(\pi_1 \cdot \{\langle\pi_2\rangle\}? \cdot \leq)^{\mathsf{c}} \cdot (< \cdot \{\langle\pi_1^{-1}\rangle\}? \cdot \pi_2)^{\mathsf{c}}\,.
\end{aligned}$$

Finally, notice that

$$\begin{aligned}
\{\neg\langle\overline{\pi}_1\rangle\}? \cdot \top &\equiv \{\neg\langle\pi_1 \cdot \pi_2\rangle\}? \cdot \top \\
\top \cdot \{\neg\langle\overline{\pi}_2\rangle\}? &\equiv \top \cdot \{\langle(\pi_1 \cdot \pi_2)^{-1}\rangle\}?\,.
\end{aligned}$$

□

Notice that by applying Lemma 3.33 to formulas $\pi_1' = \lambda \cdot \pi_1$ and $\pi_2' = \pi_2 \cdot \mu$, we obtain a decomposition of $(\lambda \cdot (\pi_1 \cdot \pi_2) \cdot \mu)^{\mathsf{c}}$ in terms of formulas of the form

$$(\lambda' \cdot (\{\varphi\}? \cdot \pi_i \cdot \{\psi\}?) \cdot \mu')^{\mathsf{c}} \qquad (i \in \{1,2\} \text{ and } \lambda', \mu' \in \{\leq, <, \geq, >\})\,.$$

Furthermore, the next lemma shows that we can get rid of the test formulas, so that the complement operator only occurs in subformulas of the form

$$(\lambda' \cdot \pi_i \cdot \mu')^{\mathsf{c}} \qquad (i \in \{1,2\} \text{ and } \lambda', \mu' \in \{\leq, <, \geq, >\})\,.$$

Lemma 3.34. *Let $\pi \in \mathrm{PDL}_{\mathsf{sf}}^{<\bullet}[\Sigma]$ such that for all $\lambda \in \{\leq, <, \geq, >\}$, the formula*

$$(\lambda \cdot \pi)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}_{\mathsf{sf}}^{<\bullet}[\Sigma]$ path formulas. Then for all $\varphi \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, for all $\lambda \in \{\leq, <, \geq, >\}$, the formula

$$(\lambda \cdot \{\varphi\}? \cdot \pi)^{\mathsf{c}}$$

is also equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}_{\mathsf{sf}}^{<\bullet}[\Sigma]$ path formulas.

Proof. Let $\varphi \in \mathrm{PDL}_{\mathsf{sf}}^{<\bullet}[\Sigma]$, and $\overline{\pi} = \{\varphi\}? \cdot \pi$.

Case $\lambda = \leq$. Let us show that this case reduces to the case $\lambda = <$. More precisely, we prove that

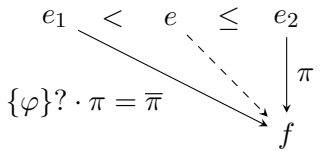
$$(\leq \cdot \overline{\pi})^{\mathsf{c}} \equiv (\leq \cdot \pi)^{\mathsf{c}} + \{\neg\langle\overline{\pi}\rangle\}? \cdot (< \cdot \overline{\pi})^{\mathsf{c}} + \top \cdot \{\neg\,\langle\overline{\pi}^{-1}\rangle\,\}? \,.$$

The right-to-left implication is easy to check. Conversely, let

$$(e, f) \in [\![(\leq \cdot \overline{\pi})^{\mathsf{c}}]\!] \setminus [\![(\leq \cdot \pi)^{\mathsf{c}} + \top \cdot \{\neg\,\langle\overline{\pi}^{-1}\rangle\,\}?]\!] \;.$$

Since $(e, f) \in [\![(\leq \cdot \overline{\pi})^{\mathsf{c}}]\!]$, we also have $(e, f) \in [\![(< \cdot \overline{\pi})^{\mathsf{c}}]\!]$. So we only need to prove that $e \in [\![\neg\langle\overline{\pi}\rangle]\!] \equiv [\![\neg\langle\pi\rangle \vee \neg\varphi]\!]$. Suppose that $e \in [\![\langle\pi\rangle]\!]$, and let us show that $e \notin [\![\varphi]\!]$.

By assumption, $f \notin [\![\neg\langle\overline{\pi}^{-1}\rangle]\!]$, hence there exists e_1 such that $(e_1, f) \in [\![\overline{\pi}]\!]$. In addition, since $(e, f) \in [\![(\leq \cdot \overline{\pi})^{\mathsf{c}}]\!]$, we must have $e_1 < e$. Now, since $(e, f) \notin [\![(\leq \cdot \pi)^{\mathsf{c}}]\!]$, there is also $e \leq e_2$ such that $(e_2, f) \in [\![\pi]\!]$:



We then have

$$e_1 \leq e \leq e_2 \,, \quad e \in [\![\langle\pi\rangle]\!] \quad \text{and} \quad e_1, e_2 \in [\![\pi]\!]^{-1}(f) \,,$$

and since π is interval-preserving, we obtain $(e, f) \in [\![\pi]\!]$. Finally, we also know that $(e, f) \notin [\![\overline{\pi}]\!] = [\![\{\varphi\}? \cdot \pi]\!]$, hence $e \notin [\![\varphi]\!]$.

Case $\lambda = \geq$. Similarly, this case reduces to the case $\lambda = >$. We have

$$(\geq \cdot \overline{\pi})^{\mathsf{c}} \equiv (\geq \cdot \pi)^{\mathsf{c}} + \{\neg\langle\overline{\pi}\rangle\}? \cdot (> \cdot \overline{\pi})^{\mathsf{c}} + \top \cdot \{\neg\,\langle\overline{\pi}^{-1}\rangle\,\}? \,.$$

Case $\lambda = <$. Let us show that

$$\begin{aligned}(< \cdot \overline{\pi})^{\mathsf{c}} \equiv \Big(\{\neg\langle< \cdot \overline{\pi}\rangle\}? \cdot \top\Big) + \Big(\top \cdot \{\neg\langle(< \cdot \overline{\pi})^{-1}\rangle\}?\Big) + \\ (< \cdot \pi)^{\mathsf{c}} + \Big(<_{\neg\langle\overline{\pi}\rangle} \cdot (\leq \cdot \pi)^{\mathsf{c}}\Big) + \Big(<_{\neg\langle\overline{\pi}\rangle} \cdot \{\neg\langle\overline{\pi}\rangle\}? \cdot (< \cdot \pi)^{\mathsf{c}}\Big) \,.\end{aligned}$$

The right-to-left implication is easy to check. Conversely, let

$$(e, f) \in [\![(< \cdot \overline{\pi})^{\mathsf{c}}]\!] \setminus [\![\{\neg\langle< \cdot \overline{\pi}\rangle\}? \cdot \top + \top \cdot \{\neg\langle(< \cdot \overline{\pi})^{-1}\rangle\}?]\!] \,.$$

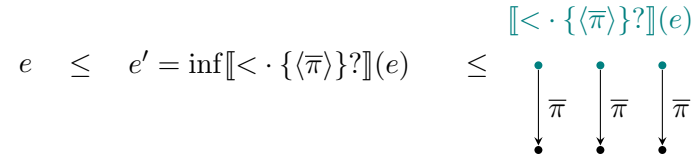
It follows from the fact that $(e, f) \notin [\![\{\neg\langle< \cdot \overline{\pi}\rangle\}? \cdot \top]\!]$ that

$$e < [\![< \cdot \{\langle\overline{\pi}\rangle\}?]\!](e) \neq \emptyset \,.$$

Therefore, in a complete Σ-structure,

$$e' = \inf[\![< \cdot \{\langle\overline{\pi}\rangle\}?]\!](e)$$

is well-defined. Note that $e \leq e'$, and that if $e < e'$, then $(e, e') \in [\![<_{\neg\langle\overline{\pi}\rangle}]\!]$, as illustrated in the figure below:



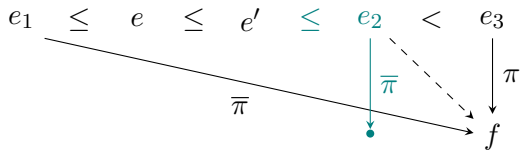
We want to prove that $(e, f) \in [\![(< \cdot \pi)^{\mathsf{c}}]\!]$ or $(e, f) \in [\![<_{\neg\langle\overline{\pi}\rangle} \cdot (\leq \cdot \pi)^{\mathsf{c}}]\!]$ or $(e, f) \in [\![<_{\neg\langle\overline{\pi}\rangle} \cdot \{\neg\langle\overline{\pi}\rangle\}? \cdot (< \cdot \pi)^{\mathsf{c}}]\!]$. To do so, it suffices to show that one of the following holds:

$$\begin{aligned} &e = e' \text{ and } (e', f) \in [\![(< \cdot \pi)^{\mathsf{c}}]\!] && \text{or} \\ &e < e' \text{ and } (e', f) \in [\![(\leq \cdot \pi)^{\mathsf{c}}]\!] && \text{or} \\ &e < e' < [\![< \cdot \{\langle\overline{\pi}\rangle\}?]\!](e) \text{ and } (e', f) \in [\![(< \cdot \pi)^{\mathsf{c}}]\!]\,. \end{aligned}$$

Let us first prove that

$$(e', f) \in [\![(< \cdot \pi)^{\mathsf{c}}]\!]\,.$$

Suppose towards a contradiction that there exists $e' < e_3$ such that $(e_3, f) \in [\![\pi]\!]$. Since $e' = \inf[\![< \cdot \{\langle\overline{\pi}\rangle\}?]\!](e)$, there exists $e_2 \in [\![\langle\overline{\pi}\rangle]\!]$ such that $e' \leq e_2 < e_3$. Moreover, since $f \in [\![\langle(< \cdot \overline{\pi})^{-1}\rangle]\!]$, there exists e_1 such that $(e_1, f) \in [\![\overline{\pi}]\!]$. Since $(e, f) \in [\![(< \cdot \overline{\pi})^{\mathsf{c}}]\!]$, we must have $e_1 \leq e$:



We then have

$$e_1 \leq e_2 \leq e_3\,, \quad e_2 \in [\![\langle\overline{\pi}\rangle]\!] \quad \text{and} \quad e_1, e_3 \in [\![\pi]\!]^{-1}(f)\,,$$

and since π is interval-preserving, $(e_2, f) \in [\![\pi]\!]$, that is, $(e_2, f) \in [\![\overline{\pi}]\!]$. This contradicts the fact that $(e, f) \notin [\![< \cdot \overline{\pi}]\!]$. Hence,

$$(e', f) \in [\![(< \cdot \pi)^{\mathsf{c}}]\!]\,.$$

If $e = e'$ or $e < e' < [\![< \cdot \{\langle\overline{\pi}\rangle\}?]\!](e)$, there is nothing more to prove. Otherwise,

$$e < e' = \min[\![< \cdot \{\langle\overline{\pi}\rangle\}?]\!](e)\,.$$

Then $(e', f) \notin [\![\overline{\pi}]\!] = [\![\{\varphi\}? \cdot \pi]\!]$ but $e' \in [\![\varphi]\!]$, hence $(e', f) \notin [\![\pi]\!]$, that is, $(e', f) \in [\![(\leq \cdot \pi)^{\mathsf{c}}]\!]$.

Case $\lambda = >$. This case is symmetric to the previous one: we have

$$(> \cdot \overline{\pi})^{\mathsf{c}} \equiv \left(\{\neg\langle > \cdot \overline{\pi}\rangle\}? \cdot \top\right) + \left(\top \cdot \{\neg\langle (> \cdot \overline{\pi})^{-1}\rangle\}?\right) + (> \cdot \pi)^{\mathsf{c}} + \left(>_{\neg\langle\overline{\pi}\rangle} \cdot (\geq \cdot \pi)^{\mathsf{c}}\right) + \left(>_{\neg\langle\overline{\pi}\rangle} \cdot \{\neg\langle\overline{\pi}\rangle\}? \cdot (> \cdot \pi)^{\mathsf{c}}\right). \qquad \square$$

Putting together Lemmas 3.33 and 3.34, we obtain the following.

Lemma 3.35. *Let $\pi_1, \pi_2 \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ such that for all $\lambda, \mu \in \{\leq, <, \geq, >\}$,*

$$(\lambda \cdot \pi_1 \cdot \mu)^{\mathsf{c}} \qquad \textit{and} \qquad (\lambda \cdot \pi_2 \cdot \mu)^{\mathsf{c}}$$

are equivalent, over complete interval-preserving Σ-structures, to finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas. Then for all $\lambda, \mu \in \{\leq, <, \geq, >\}$,

$$(\lambda \cdot (\pi_1 \cdot \pi_2) \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Proof. By Lemma 3.33 applied to $\pi'_1 = \lambda \cdot \pi_1$ and $\pi'_2 = \pi_2 \cdot \mu$, it is sufficient to prove that for all $\lambda, \mu \in \{\leq, <, \geq, >\}$ and $\varphi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$,

$$(\lambda \cdot \pi_1 \cdot \{\varphi\}? \cdot \mu)^{\mathsf{c}} \qquad \text{and} \qquad (\lambda \cdot \{\varphi\}? \cdot \pi_2 \cdot \mu)^{\mathsf{c}}$$

are equivalent to finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

For $(\lambda \cdot \{\varphi\}? \cdot \pi_2 \cdot \mu)^{\mathsf{c}}$, this follows from Lemma 3.34 applied to $\pi' = \pi_2 \cdot \mu$.

For $(\lambda \cdot \pi_1 \cdot \{\varphi\}? \cdot \mu)^{\mathsf{c}}$, the situation is symmetric, so we can go through converses of path formulas to obtain the same result. More precisely, let $\pi' = \pi_1^{-1} \cdot \lambda^{-1}$. For all λ',

$$(\lambda' \cdot \pi')^{\mathsf{c}} \equiv \left(\left(\lambda \cdot \pi_1 \cdot (\lambda')^{-1}\right)^{-1}\right)^{\mathsf{c}} \equiv \left((\lambda \cdot \pi_1 \cdot (\lambda')^{-1})^{\mathsf{c}}\right)^{-1}.$$

Since $(\lambda \cdot \pi_1 \cdot (\lambda')^{-1})^{\mathsf{c}}$ is equivalent to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas, so is its converse, and thus $(\lambda' \cdot \pi')^{\mathsf{c}}$. Therefore, π' satisfies the conditions of Lemma 3.34. So for all $\varphi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ and $\mu \in \{\leq, <, \geq, >\}$, taking $\lambda' = \mu^{-1}$, we obtain that

$$\left(\mu^{-1} \cdot \{\varphi\}? \cdot \pi'\right)^{\mathsf{c}}$$

is equivalent to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas. Therefore, so is its converse,

$$\left(\left(\mu^{-1} \cdot \{\varphi\}? \cdot \pi'\right)^{\mathsf{c}}\right)^{-1} \equiv \left(\left(\mu^{-1} \cdot \{\varphi\}? \cdot \pi'\right)^{-1}\right)^{\mathsf{c}} \equiv (\lambda \cdot \pi_1 \cdot \{\varphi\}? \cdot \mu)^{\mathsf{c}}. \qquad \square$$

We can prove a similar result for $\pi_1 + \pi_2$:

Lemma 3.36. *Let $\pi_1, \pi_2 \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ such that for all $\lambda, \mu \in \{\leq, <, \geq, >\}$,*

$$(\lambda \cdot \pi_1 \cdot \mu)^{\mathsf{c}} \qquad \textit{and} \qquad (\lambda \cdot \pi_2 \cdot \mu)^{\mathsf{c}}$$

are equivalent, over complete interval-preserving Σ-structures, to finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas. Then for all $\lambda, \mu \in \{\leq, <, \geq, >\}$,

$$(\lambda \cdot (\pi_1 + \pi_2) \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Proof. The proof relies on the following observation:

Claim 3.37. For all $\pi_1, \pi_2 \in \mathrm{PDL}_{\mathsf{sf}}[\Sigma]$,

$$\begin{aligned}((\pi_1 + \pi_2) \cdot \leq)^{\mathsf{c}} \equiv \{\neg\mathsf{Loop}((\pi_1 \cdot \leq)^{\mathsf{c}} \cdot (\pi_2 \cdot \leq)^{-1})\}? \cdot (\pi_1 \cdot \leq)^{\mathsf{c}} + \\ \{\neg\mathsf{Loop}((\pi_2 \cdot \leq)^{\mathsf{c}} \cdot (\pi_1 \cdot \leq)^{-1})\}? \cdot (\pi_2 \cdot \leq)^{\mathsf{c}} \\ ((\pi_1 + \pi_2) \cdot \geq)^{\mathsf{c}} \equiv \{\neg\mathsf{Loop}((\pi_1 \cdot \geq)^{\mathsf{c}} \cdot (\pi_2 \cdot \geq)^{-1})\}? \cdot (\pi_1 \cdot \geq)^{\mathsf{c}} + \\ \{\neg\mathsf{Loop}((\pi_2 \cdot \geq)^{\mathsf{c}} \cdot (\pi_1 \cdot \geq)^{-1})\}? \cdot (\pi_2 \cdot \geq)^{\mathsf{c}}\,.\end{aligned}$$

Proof. We prove the first equivalence, the second is similar. We have

$$((\pi_1 + \pi_2) \cdot \leq)^{\mathsf{c}} \equiv (\pi_1 \cdot \leq)^{\mathsf{c}} \cap (\pi_2 \cdot \leq)^{\mathsf{c}}\,.$$

Note that for all e, $[\![(\pi_1 \cdot \leq)^{\mathsf{c}}]\!](e)$ and $[\![(\pi_2 \cdot \leq)^{\mathsf{c}}]\!](e)$ are downward-closed. Indeed,

$$f \in [\![(\pi_i \cdot \leq)^{\mathsf{c}}]\!](e) \iff f < [\![\pi_i]\!](e)\,.$$

Thus, we must have

$$[\![(\pi_1 \cdot \leq)^{\mathsf{c}}]\!](e) \subseteq [\![(\pi_2 \cdot \leq)^{\mathsf{c}}]\!](e) \quad \text{or} \quad [\![(\pi_2 \cdot \leq)^{\mathsf{c}}]\!](e) \subseteq [\![(\pi_1 \cdot \leq)^{\mathsf{c}}]\!](e)\,,$$

or equivalently,

$$e \in \left[\!\!\left[\neg\mathsf{Loop}((\pi_1 \cdot \leq)^{\mathsf{c}} \cdot (\pi_2 \cdot \leq)^{-1})\right]\!\!\right] \quad \text{or} \quad e \in \left[\!\!\left[\neg\mathsf{Loop}((\pi_2 \cdot \leq)^{\mathsf{c}} \cdot (\pi_1 \cdot \leq)^{-1})\right]\!\!\right]\,.$$

In the first case, we have

$$\begin{aligned}[\![((\pi_1 + \pi_2) \cdot \leq)^{\mathsf{c}}]\!](e) &= [\![(\pi_1 \cdot \leq)^{\mathsf{c}}]\!] \cap [\![(\pi_2 \cdot \leq)^{\mathsf{c}}]\!] = [\![(\pi_1 \cdot \leq)^{\mathsf{c}}]\!](e) \\ &= \left[\!\!\left[\{\neg\mathsf{Loop}((\pi_1 \cdot \leq)^{\mathsf{c}} \cdot (\pi_2 \cdot \leq)^{-1})\}? \cdot (\pi_1 \cdot \leq)^{\mathsf{c}}\right]\!\!\right](e)\,,\end{aligned}$$

and in the second case,

$$\begin{aligned}[\![((\pi_1 + \pi_2) \cdot \leq)^{\mathsf{c}}]\!](e) &= [\![(\pi_1 \cdot \leq)^{\mathsf{c}}]\!] \cap [\![(\pi_2 \cdot \leq)^{\mathsf{c}}]\!] = [\![(\pi_2 \cdot \leq)^{\mathsf{c}}]\!](e) \\ &= \left[\!\!\left[\{\neg\mathsf{Loop}((\pi_2 \cdot \leq)^{\mathsf{c}} \cdot (\pi_1 \cdot \leq)^{-1})\}? \cdot (\pi_2 \cdot \leq)^{\mathsf{c}}\right]\!\!\right](e)\,. \qquad \square\end{aligned}$$

Let $\pi_1, \pi_2 \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ such that for all $\lambda, \mu \in \{\leq, <, \geq, >\}$,

$$(\lambda \cdot \pi_1 \cdot \mu)^{\mathsf{c}} \qquad \text{and} \qquad (\lambda \cdot \pi_2 \cdot \mu)^{\mathsf{c}}$$

are equivalent, over complete interval-preserving Σ-structures, to finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Let $\lambda, \mu \in \{\leq, <, \geq, >\}$. If $\mu = \{\leq\}$, then we apply the first equivalence of Claim 3.37 to $\pi'_1 = \lambda \cdot \pi_1$ and $\pi'_2 = \lambda \cdot \pi_2$. We obtain a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas equivalent to

$$((\lambda \cdot \pi_1 + \lambda \cdot \pi_2) \cdot \leq)^{\mathsf{c}} \equiv (\lambda \cdot (\pi_1 + \pi_2) \cdot \mu)^{\mathsf{c}}\,.$$

If $\mu = \{<\}$, then we apply the first equivalence of Claim 3.37 to $\pi'_1 = \lambda \cdot \pi_1 \cdot <$ and $\pi'_2 = \lambda \cdot \pi_2 \cdot <$. Note that in that case, we have

$$(\pi'_i \cdot \leq)^{\mathsf{c}} \equiv (\lambda \cdot \pi_i \cdot <)^{\mathsf{c}} \equiv (\lambda \cdot \pi_i \cdot \mu)^{\mathsf{c}}\,,$$

so we obtain again a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas equivalent to

$$((\lambda \cdot \pi_1 \cdot < + \lambda \cdot \pi_2 \cdot <) \cdot \leq)^{\mathsf{c}} \equiv (\lambda \cdot (\pi_1 + \pi_2) \cdot < \cdot \leq)^{\mathsf{c}} \equiv (\lambda \cdot (\pi_1 + \pi_2) \cdot \mu)^{\mathsf{c}}\,.$$

Similarly, if $\mu = \{\geq\}$, then we apply the second equivalence of Claim 3.37 to $\pi'_1 = \lambda \cdot \pi_1$ and $\pi'_2 = \lambda \cdot \pi_2$, and if $\mu = >$, then we apply the second equivalence of Claim 3.37 to $\pi'_1 = \lambda \cdot \pi_1 \cdot >$ and $\pi'_2 = \lambda \cdot \pi_2 \cdot >$. □

3.8.4 Complements for base path formulas

Recall that we want to prove by induction that for all finite unions π of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ paths formulas, and for all $\lambda, \mu \in \{\leq, \geq\}$, the formula $(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$ is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas. We have solved the inductive cases $\pi = \pi_1 \cdot \pi_2$ and $\pi = \pi_1 + \pi_2$ in Lemmas 3.35 and 3.36, provided we generalize the result to all $\lambda, \mu \in \{\leq, <, \geq, >\}$. We also know that finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas are closed under converse. We are left with the base cases: we want to show that for all $\pi = \alpha$, $\pi = \mathsf{sup}\ (\lambda \cdot \alpha)$, $\pi = \mathsf{inf}\ (\lambda \cdot \alpha)$, $\pi = \{\varphi\}?$, or $\pi = <_\varphi$, formulas of the form $(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$ are expressible as finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ formulas.

Lemma 3.38. *For all $\alpha \in \mathsf{Rel} \setminus \{\leq\}$ and $\lambda, \mu \in \{\leq, <, \geq, >\}$, the formula*

$$(\lambda \cdot \alpha \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Proof. This follows from the fact that for all path formula π, over complete Σ-structures,

$$\begin{aligned}
(\lambda\cdot\pi\cdot<)^{\mathsf{c}} &\equiv \{\neg\langle\lambda\cdot\pi\cdot<\rangle\}?\cdot\top + \mathsf{inf}\ (\lambda\cdot\pi)\cdot\geq\\
(\lambda\cdot\pi\cdot\leq)^{\mathsf{c}} &\equiv \{\neg\langle\lambda\cdot\pi\cdot\leq\rangle\}?\cdot\top + \mathsf{inf}\ (\lambda\cdot\pi)\cdot> +\\
&\qquad \{\neg\mathsf{Loop}((\lambda\cdot\pi)\cdot(\mathsf{inf}\ (\lambda\cdot\pi))^{-1})\}?\cdot\mathsf{inf}\ (\lambda\cdot\pi)\\
(\lambda\cdot\pi\cdot>)^{\mathsf{c}} &\equiv \{\neg\langle\lambda\cdot\pi\cdot>\rangle\}?\cdot\top + \mathsf{sup}\ (\lambda\cdot\pi)\cdot\leq\\
(\lambda\cdot\pi\cdot\geq)^{\mathsf{c}} &\equiv \{\neg\langle\lambda\cdot\pi\cdot\geq\rangle\}?\cdot\top + \mathsf{sup}\ (\lambda\cdot\pi)\cdot< +\\
&\qquad \{\neg\mathsf{Loop}((\lambda\cdot\pi)\cdot(\mathsf{sup}\ (\lambda\cdot\pi))^{-1})\}?\cdot\mathsf{sup}\ (\lambda\cdot\pi)\,.
\end{aligned}$$

Let us prove the first two equivalences; the last two are symmetric. As usual, we trivially have

$$[\![\{\neg\langle\lambda\cdot\pi\cdot>\rangle\}?\cdot\top]\!]\subseteq[\![(\lambda\cdot\pi\cdot>)^{\mathsf{c}}]\!]\,.$$

Recall also that

$$(e,f)\in[\![(\lambda\cdot\pi\cdot<)^{\mathsf{c}}]\!]\iff f\leq[\![\lambda\cdot\pi]\!](e)\,.$$

If $(e,f)\in[\![\mathsf{inf}\ (\lambda\cdot\pi)\cdot\geq]\!]$, then

$$f\leq\inf[\![\lambda\cdot\pi]\!](e)\leq[\![\lambda\cdot\pi]\!](e)\,,$$

therefore, $(e,f)\in[\![(\lambda\cdot\pi\cdot<)^{\mathsf{c}}]\!]$. Conversely, let

$$(e,f)\in[\![(\lambda\cdot\pi\cdot<)^{\mathsf{c}}]\!]\setminus[\![\{\neg\langle\lambda\cdot\pi\cdot<\rangle\}?\cdot\top]\!]\,.$$

We have $f\leq[\![\lambda\cdot\pi]\!](e)\neq\emptyset$. Since the Σ-structure is complete, $\inf[\![\lambda\cdot\pi]\!](e)$ is well defined, and $f\leq\inf[\![\lambda\cdot\pi]\!](e)$. Therefore,

$$(e,f)\in[\![\mathsf{inf}\ (\lambda\cdot\pi)\cdot\geq]\!]\,.$$

Thus,

$$(\lambda\cdot\pi\cdot<)^{\mathsf{c}}\equiv\{\neg\langle\lambda\cdot\pi\cdot<\rangle\}?\cdot\top+\mathsf{inf}\ (\lambda\cdot\pi)\cdot\geq\,.$$

Let us move to the proof of the second equation. As in the previous case,

$$(e,f)\in[\![(\lambda\cdot\pi\cdot\leq)^{\mathsf{c}}]\!]\iff f<[\![\lambda\cdot\pi]\!](e)\,,$$

which implies

$$[\![\{\neg\langle\lambda\cdot\pi\cdot\leq\rangle\}?\cdot\top+\mathsf{inf}\ (\lambda\cdot\pi)\cdot>]\!]\subseteq[\![(\lambda\cdot\pi\cdot\leq)^{\mathsf{c}}]\!]\,.$$

On the other hand, we may have $\inf[\![\lambda\cdot\pi]\!](e)<[\![\lambda\cdot\pi]\!](e)$ or $\inf[\![\lambda\cdot\pi]\!](e)\in[\![\lambda\cdot\pi]\!](e)$, so the inclusion may be strict. Let

$$(e,f)\in[\![(\lambda\cdot\pi\cdot\leq)^{\mathsf{c}}]\!]\setminus[\![\{\neg\langle\lambda\cdot\pi\cdot\leq\rangle\}?\cdot\top+\mathsf{inf}\ (\lambda\cdot\pi)\cdot>]\!]\,.$$

Since $f < [\![\lambda \cdot \pi]\!](e) \neq \emptyset$, $\inf [\![\lambda \cdot \pi]\!](e)$ is well-defined, and $f \leq \inf [\![\lambda \cdot \pi]\!](e)$. Since $(e, f) \notin [\![\mathsf{inf}\ (\lambda \cdot \pi) \cdot >]\!]$, this means that $f = \inf [\![\lambda \cdot \pi]\!](e)$. The condition $f < [\![\lambda \cdot \pi]\!](e)$ can then be written as

$$e \in \left[\!\!\left[\neg \mathsf{Loop}((\lambda \cdot \pi) \cdot (\mathsf{inf}\ (\lambda \cdot \pi))^{-1})\right]\!\!\right] .$$

Therefore, $(e, f) \in [\![\{\neg \mathsf{Loop}((\lambda \cdot \pi) \cdot (\mathsf{inf}\ (\lambda \cdot \pi))^{-1})\}? \cdot \mathsf{inf}\ (\lambda \cdot \pi)]\!]$. This proves the left-to-right implication of the second equivalence. Conversely,

$$(e, f) \in [\![\{\neg \mathsf{Loop}((\lambda \cdot \pi) \cdot (\mathsf{inf}\ (\lambda \cdot \pi))^{-1})\}? \cdot \mathsf{inf}\ (\lambda \cdot \pi)]\!] \quad \text{implies} \quad f < [\![\lambda \cdot \pi]\!](e)\,,$$

that is, $(e, f) \in [\![(\lambda \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]$. Thus,

$$\begin{aligned}(\lambda \cdot \pi \cdot \leq)^{\mathsf{c}} \equiv\ & \{\neg\langle \lambda \cdot \pi \cdot \leq\rangle\}? \cdot \top + \mathsf{inf}\ (\lambda \cdot \pi) \cdot > + \\ & \{\neg \mathsf{Loop}((\lambda \cdot \pi) \cdot (\mathsf{inf}\ (\lambda \cdot \pi))^{-1})\}? \cdot \mathsf{inf}\ (\lambda \cdot \pi)\,.\end{aligned}$$ □

The next two lemmas deal with the case of formulas of the form $\pi = \mathsf{sup}\ (\lambda \cdot \alpha)$ or $\pi = \mathsf{inf}\ (\lambda \cdot \alpha)$. According to Lemma 3.28, we can distinguish two cases, depending on whether $[\![\pi]\!]$ is non-decreasing or non-increasing.

Lemma 3.39. *Let π be a path formula of the form*

$$\pi = \mathsf{sup}\ (\geq \cdot \alpha)\,, \quad \pi = \mathsf{sup}\ (> \cdot \alpha)\,, \quad \pi = \mathsf{inf}\ (\leq \cdot \alpha)\,, \text{ or } \quad \pi = \mathsf{inf}\ (< \cdot \alpha)\,,$$

for some $\alpha \in \mathsf{Rel} \setminus \{\leq\}$. Then for all $\lambda \in \{\leq, <, \geq, >\}$, the formula

$$(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$$

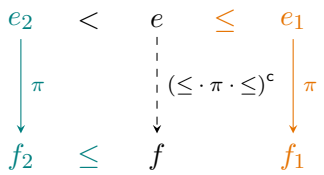
is equivalent, over complete Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Proof. The proof relies on Lemma 3.28. First, using the fact that $[\![\langle \pi \rangle]\!]$ is an interval, we obtain:

Claim 3.40. For all $(e, f) \in [\![(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}]\!]$, at least one of the following holds:

1. $[\![\lambda \cdot \pi \cdot \mu]\!](e) = \emptyset$
2. $\left[\!\!\left[(\lambda \cdot \pi \cdot \mu)^{-1}\right]\!\!\right](f) = \emptyset$
3. $[\![\pi]\!](e) \neq \emptyset$.

Proof. Let us illustrate the proof in the case $\lambda = \leq$ and $\mu = \leq$. Let $(e, f) \in [\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]$ satisfying neither condition 1 nor 2. We have $[\![\leq \cdot \pi \cdot \leq]\!](e) \neq \emptyset$, hence there exists e_1, f_1 such that $e \leq e_1$ and $(e_1, f_1) \in [\![\pi]\!]$. We also have $[\![(\leq \cdot \pi \cdot \leq)^{-1}]\!](f) \neq \emptyset$, hence there exists e_2, f_2 such that $f_2 \leq f$ and $(e_2, f_2) \in [\![\pi]\!]$. Moreover, since $(e, f) \notin [\![\leq \cdot \pi \cdot \leq]\!]$, we cannot have $e \leq e_2$. That is, $e_2 < e$:



Since $[\![\langle\pi\rangle]\!]$ is an interval containing e_1 and e_2, we obtain $[\![\pi]\!](e) \neq \emptyset$.

In the general case, we obtain similarly $(e_1, f_1), (e_2, f_2) \in [\![\pi]\!]$ such that $e \ \lambda \ e_1$ and $f_2 \ \mu \ f$, which implies $e \ \lambda^{\mathsf{c}} \ e_2$. So we have $e_2 \ \lambda' \ e \ \lambda \ e_1$ for some $(\lambda', \lambda) \in \{(\leq, <), (<, \leq), (\geq, >), (>, \geq)$. In any case, e belongs to the interval $[\![\langle\pi\rangle]\!]$. □

Note that conditions 1 and 2 of Claim 3.40 always imply $(e, f) \in [\![(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}]\!]$, and their disjunction hold if and only if

$$(e, f) \in \left[\!\!\left[\{\neg\langle\lambda \cdot \pi \cdot \mu\rangle\}? \cdot \top + \top \cdot \{\neg\langle(\lambda \cdot \pi \cdot \mu)^{-1}\rangle\}?\right]\!\!\right] .$$

So to decompose $(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$ as a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas, we only need to find a suitable characterization of $[\![(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}]\!](e)$ in the case where $[\![\pi]\!](e) \neq \emptyset$. To do so, we use the fact that $[\![\pi]\!]$ (seen as a partial function) is non-decreasing.

Case $\lambda = \leq$ and $\mu = \leq$. If $[\![\pi]\!](e) \neq \emptyset$, then, since $[\![\pi]\!]$ is non-decreasing, the unique element of $[\![\pi]\!](e)$ is the minimum of $[\![\leq \cdot \pi]\!](e)$. Therefore,

$$[\![(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}]\!](e) = \{f \mid f < [\![\leq \cdot \pi]\!](e)\} = \{f \mid f < [\![\pi]\!](e)\} = [\![\pi \cdot >]\!](e) .$$

So we obtain

$$(\leq \cdot \pi \cdot \leq)^{\mathsf{c}} \equiv \{\neg\langle\leq \cdot \pi \cdot \leq\rangle\}? \cdot \top + \top \cdot \{\neg\langle(\leq \cdot \pi \cdot \leq)^{-1}\rangle\}? + (\pi \cdot >) .$$

Case $\lambda = \geq$ and $\mu = \geq$. Symmetrically,

$$(\geq \cdot \pi \cdot \geq)^{\mathsf{c}} \equiv \{\neg\langle\geq \cdot \pi \cdot \geq\rangle\}? \cdot \top + \top \cdot \{\neg\langle(\geq \cdot \pi \cdot \geq)^{-1}\rangle\}? + (\pi \cdot <) .$$

Case $\lambda = \leq$ and $\mu = <$. Similarly to the first case,

$$(\leq \cdot \pi \cdot <)^{\mathsf{c}} \equiv \{\neg\langle\leq \cdot \pi \cdot <\rangle\}? \cdot \top + \top \cdot \{\neg\langle(\leq \cdot \pi \cdot <)^{-1}\rangle\}? + (\pi \cdot \geq) .$$

Indeed, for e such that $[\![\pi]\!](e) \neq \emptyset$, we have, using as before the monotonicity of $[\![\pi]\!]$,

$$[\![(\leq \cdot \pi \cdot <)^{\mathsf{c}}]\!](e) = \{f \mid f \leq [\![\leq \cdot \pi]\!](e)\} = \{f \mid f \leq [\![\pi]\!](e)\} = [\![\pi \cdot \geq]\!](e) .$$

Case $\lambda = \geq$ and $\mu = >$. Symmetrically,

$$(\geq \cdot \pi \cdot >)^{\mathsf{c}} \equiv \{\neg\langle\geq \cdot \pi \cdot >\rangle\}? \cdot \top + \top \cdot \{\neg\langle(\geq \cdot \pi \cdot >)^{-1}\rangle\}? + (\pi \cdot \leq) .$$

Case $\lambda = <$ and $\mu = \leq$. This case is slightly different, since we no longer have $[\![\pi]\!] \subseteq [\![\lambda \cdot \pi]\!]$. We need to divide the case $[\![\pi]\!](e) \neq \emptyset$ into two sub-cases, according to whether the unique $f \in [\![\pi]\!](e)$ is also in $[\![< \cdot \pi]\!](e)$.

If $[\![\pi]\!](e) \neq \emptyset$ and $[\![\pi]\!](e) \subseteq [\![< \cdot \pi]\!](e)$, that is, if $e \in \left[\!\left[\mathsf{Loop}(< \cdot \pi \cdot \pi^{-1})\right]\!\right]$, we have, as in the first case,

$$[\![(< \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]\,(e) = [\![\pi \cdot >]\!]\,(e)\,.$$

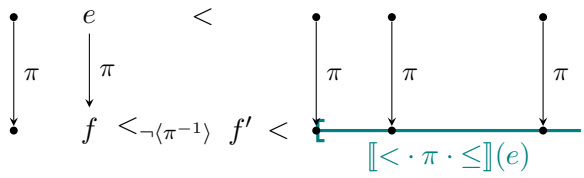
Assume that $[\![\pi]\!](e) \neq \emptyset$ but $[\![\pi]\!](e) \not\subseteq [\![< \cdot \pi]\!](e)$, that is, $e \in \left[\!\left[\neg\mathsf{Loop}(< \cdot \pi \cdot \pi^{-1})\right]\!\right]$. Let us show that

$$[\![(< \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]\,(e) = \left[\!\left[\pi \cdot \geq + \pi \cdot <_{\neg\langle\pi^{-1}\rangle} \cdot \{\neg\langle\pi^{-1}\rangle\}?\right]\!\right](e)\,.$$

Let f be the unique element in $[\![\pi]\!](e)$. Since $[\![\pi]\!]$ is non-increasing and $f \notin [\![< \cdot \pi]\!](e)$, we have $f < [\![< \cdot \pi]\!](e)$. So for all $f' \in [\![\pi \cdot \geq]\!](e)$, we have

$$f' \leq f < [\![< \cdot \pi]\!](e)\,,$$

that is, $f' \in [\![(< \cdot \pi \cdot \leq)^{\mathsf{c}}]\!](e)$. In addition, by monotonicity of $[\![\pi]\!]$, for all $f < f'$, we have $f' < [\![< \cdot \pi]\!](e)$ if and only if for all $f < g \leq f'$, $g \notin \langle\pi^{-1}\rangle$:



Moreover, we have ($f < f'$ and $g \notin [\![\langle\pi^{-1}\rangle]\!]$ for all $f < g \leq f'$) if and only if $(f, f') \in \left[\!\left[<_{\neg\langle\pi^{-1}\rangle} \cdot \{\neg\langle\pi^{-1}\rangle\}?\right]\!\right]$. So we have

$$[\![(< \cdot \pi \cdot \leq)^{\mathsf{c}}]\!]\,(e) = \left[\!\left[\pi \cdot \geq + \pi \cdot <_{\neg\langle\pi^{-1}\rangle} \cdot \{\neg\langle\pi^{-1}\rangle\}?\right]\!\right](e)\,.$$

Combining all cases, we obtain

$$\begin{aligned}(< \cdot \pi \cdot \leq)^{\mathsf{c}} \equiv\ & \{\neg\langle< \cdot \pi \cdot \leq\rangle\}? \cdot \top + \top \cdot \{\neg\langle(< \cdot \pi \cdot \leq)^{-1}\rangle\}? \quad + \\ & \{\mathsf{Loop}(< \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot > \quad + \\ & \{\neg\mathsf{Loop}(< \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot \geq \quad + \\ & \{\neg\mathsf{Loop}(< \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot <_{\neg\langle\pi^{-1}\rangle} \cdot \{\neg\langle\pi^{-1}\rangle\}? \ .\end{aligned}$$

Case $\lambda = >$ and $\mu = \geq$. Symmetrically,

$$\begin{aligned}(> \cdot \pi \cdot \geq)^{\mathsf{c}} \equiv\ & \{\neg\langle> \cdot \pi \cdot \geq\rangle\}? \cdot \top + \top \cdot \{\neg\langle(> \cdot \pi \cdot \geq)^{-1}\rangle\}? \quad + \\ & \{\mathsf{Loop}(> \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot < \quad + \\ & \{\neg\mathsf{Loop}(> \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot \leq \quad + \\ & \{\neg\mathsf{Loop}(> \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot >_{\neg\langle\pi^{-1}\rangle} \cdot \{\neg\langle\pi^{-1}\rangle\}? \ .\end{aligned}$$

Case $\lambda = <$ and $\mu = <$. Similarly to the case $\lambda = <$ and $\mu = \leq$,

$$(< \cdot \pi \cdot <)^{\mathsf{c}} \equiv \{\neg\langle < \cdot \pi \cdot <\rangle\}? \cdot \top + \top \cdot \{\neg\langle (< \cdot \pi \cdot <)^{-1}\rangle\}? + \pi \cdot \geq + \{\neg \mathsf{Loop}(< \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot <_{\neg\langle \pi^{-1}\rangle} .$$

Case $\lambda = >$ and $\mu = >$. Symmetrically,

$$(> \cdot \pi \cdot >)^{\mathsf{c}} \equiv \{\neg\langle > \cdot \pi \cdot >\rangle\}? \cdot \top + \top \cdot \{\neg\langle (> \cdot \pi \cdot >)^{-1}\rangle\}? + \pi \cdot \leq + \{\neg \mathsf{Loop}(> \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot >_{\neg\langle \pi^{-1}\rangle} .$$

Case $\lambda = \leq$ and $\mu = \geq$. Let us show that

$$(\leq \cdot \pi \cdot \geq)^{\mathsf{c}} \equiv \{\neg\langle \leq \cdot \pi \cdot \geq\rangle\}? \cdot \top + \top \cdot \{\neg\langle (\leq \cdot \pi \cdot \geq)^{-1}\rangle\}? .$$

As usual, the right-to-left inclusion is trivial. Conversely, let

$$(e, f) \notin \left[\!\!\left[\{\neg\langle \leq \cdot \pi \cdot \geq\rangle\}? \cdot \top + \top \cdot \{\neg\langle (\leq \cdot \pi \cdot \geq)^{-1}\rangle\}? \right]\!\!\right] .$$

Let us show that $(e, f) \in [\![\leq \cdot \pi \cdot \geq]\!]$. Since $e \in [\![\langle \leq \cdot \pi \cdot \geq\rangle]\!]$, there exists $(e_1, f_1) \in [\![\pi]\!]$ such that $e \leq e_1$. If $f_1 \geq f$, then we are done. So, suppose $f < f_1$. Since $f \in [\![\langle (\leq \cdot \pi \cdot \geq)^{-1}\rangle]\!]$, there exists $(e_2, f_2) \in [\![\pi]\!]$ such that $f \leq f_2$:

$$\begin{array}{ccccccc} e & \leq & e_1 & & & & e_2 \\ & & \downarrow \pi & & & & \downarrow \pi \\ & & f_1 & < & f & \leq & f_2 \end{array}$$

We then have $f_1 < f_2$. By monotonicity, $e_1 < e_2$, hence, $e < e_2$ and $(e, f) \in [\![\leq \cdot \pi \cdot \geq]\!]$.

Case $(\lambda, \mu) \in \{(\leq, >), (<, \geq), (<, >), (\geq, \leq), (\geq, <), (>, \leq), (>, <)\}$. Let us show that the prof of the previous case can be generalized, and

$$(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}} \equiv \{\neg\langle \lambda \cdot \pi \cdot \mu\rangle\}? \cdot \top + \top \cdot \{\neg\langle (\lambda \cdot \pi \cdot \mu)^{-1}\rangle\}? .$$

We use the notation

$$\underline{\lambda} = \begin{cases} \leq & \text{if } \lambda \in \{\leq, <\} \\ \geq & \text{if } \lambda \in \{\geq, >\} . \end{cases}$$

Let

$$(e, f) \notin \left[\!\!\left[\{\neg\langle \lambda \cdot \pi \cdot \mu\rangle\}? \cdot \top + \top \cdot \{\neg\langle (\lambda \cdot \pi \cdot \mu)^{-1}\rangle\}? \right]\!\!\right] .$$

There exist $(e_1, f_1), (e_2, f_2) \in [\![\pi]\!]$ such that $e \ \lambda \ e_1$ and $f_2 \ \mu \ f$. If $f_1 \ \mu \ f$, then $(e, f) \in [\![\lambda \cdot \pi \cdot \mu]\!]$. Otherwise, we have $f \ \underline{\mu} \ f_1$. Then,

$$f_2 \ \mu \ f \ \underline{\mu} \ f_1 \,, \quad \text{therefore,} \quad f_2 \ \mu \ f_1 \,.$$

By monotonicity of $[\![\pi]\!]$, we obtain

$$e_2 \ \mu \ e_1 \,, \quad \text{and thus,} \quad e_1 \ \underline{\lambda} \ e_2 \,.$$

Therefore, $e \ \lambda \ e_2$, and $(e, f) \in [\![\lambda \cdot \pi \cdot \mu]\!]$. □

Lemma 3.41. *Let π be a $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formula of the form*

$$\pi = \mathsf{sup}\ (\leq \cdot \alpha) \,, \quad \pi = \mathsf{sup}\ (< \cdot \alpha) \,, \quad \pi = \mathsf{inf}\ (\geq \cdot \alpha) \,, \text{or} \quad \pi = \mathsf{inf}\ (> \cdot \alpha) \,.$$

for some $\alpha \in \mathsf{Rel} \setminus \{\leq\}$. Then for all $\lambda \in \{\leq, <, \geq, >\}$, the formula

$$(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Proof. The proof is similar to the previous one, except we now have non-increasing functions instead of non-decreasing ones. Since $[\![\langle\pi\rangle]\!]$ is an interval, we still have (with the exact same proof):

Claim 3.42. For all $(e, f) \in [\![(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}]\!]$, at least one of the following holds:

1. $[\![\lambda \cdot \pi \cdot \mu]\!](e) = \emptyset$
2. $\left[\!\left[(\lambda \cdot \pi \cdot \mu)^{-1}\right]\!\right](f) = \emptyset$
3. $[\![\pi]\!](e) \neq \emptyset$.

Since $[\![\pi]\!]$ is non-increasing rather than non-decreasing as in the previous lemma, the trivial and non-trivial cases are reversed. Apart from this, the arguments remain essentially the same. For this reason, we only give the proof for two cases. We use the same notation as before:

$$\underline{\lambda} = \begin{cases} \leq & \text{if } \lambda \in \{\leq, <\} \\ \geq & \text{if } \lambda \in \{\geq, >\} \,. \end{cases}$$

Case $\underline{\lambda} = \underline{\mu}$. Let us show that in that case,

$$(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}} \equiv \{\neg\langle \lambda \cdot \pi \cdot \mu\rangle\}? \cdot \top + \top \cdot \{\neg\langle (\lambda \cdot \pi \cdot \mu)^{-1}\rangle\}? \,.$$

Let

$$(e, f) \notin \left[\!\!\left[\{\neg\langle \lambda \cdot \pi \cdot \mu\rangle\}? \cdot \top + \top \cdot \{\neg\langle (\lambda \cdot \pi \cdot \mu)^{-1}\rangle\}?\right]\!\!\right] .$$

There exist $(e_1, f_1), (e_2, f_2) \in [\![\pi]\!]$ such that $e \ \lambda \ e_1$ and $f_2 \ \mu \ f$. If $f_1 \ \mu \ f$, then $(e, f) \in [\![\lambda \cdot \pi \cdot \mu]\!]$. Otherwise, we have $f \ \underline{\mu} \ f_1$. Then,

$$f_2 \ \mu \ f \ \underline{\mu} \ f_1 \,, \quad \text{therefore,} \quad f_2 \ \mu \ f_1 \,.$$

Since $[\![\pi]\!]$ is non-increasing, we obtain

$$e_1 \ \underline{\mu} \ e_2 \,, \quad \text{i.e.,} \quad e_1 \ \underline{\lambda} \ e_2 \,.$$

Therefore, $e \ \lambda \ e_2$, and $(e, f) \in [\![\lambda \cdot \pi \cdot \mu]\!]$.

Case $\lambda = <$ and $\mu = >$. Let us show that

$$\begin{aligned}(< \cdot \pi \cdot >)^{\mathsf{c}} \equiv \{\neg\langle < \cdot \pi \cdot \geq\rangle\}? \cdot \top + \top \cdot \{\neg\langle (< \cdot \pi \cdot \geq)^{-1}\rangle\}? + \\ \pi \cdot \leq + \{\neg \mathsf{Loop}(< \cdot \pi \cdot \pi^{-1})\}? \cdot \pi \cdot >_{\neg\langle \pi^{-1}\rangle} \,.\end{aligned}$$

Let e such that $[\![\pi]\!](e) \neq \emptyset$, and f the unique element in $[\![\pi]\!](e)$. Since $[\![\pi]\!]$ is non-increasing, we have

$$[\![< \cdot \pi]\!](e) \leq f \,.$$

In particular, for all $f' \in [\![\pi \cdot \leq]\!](e)$, $[\![< \cdot \pi]\!](e) \leq f \leq f'$, hence $f' \in [\![(< \cdot \pi \cdot >)^{\mathsf{c}}]\!](e)$.

Suppose that in addition, $f \notin [\![< \cdot \pi]\!](e)$, that is, $e \in [\![\neg \mathsf{Loop}(< \cdot \pi \cdot \pi^{-1})]\!]$, and let $f' \in \left[\!\!\left[\pi \cdot >_{\neg\langle \pi^{-1}\rangle}\right]\!\!\right](e)$, that is, $f' \in \left[\!\!\left[>_{\neg\langle \pi^{-1}\rangle}\right]\!\!\right](f)$. Suppose towards a contradiction that $f' \notin [\![(< \cdot \pi \cdot >)^{\mathsf{c}}]\!](e)$, i.e., there exists $(e_1, f_1) \in [\![\pi]\!]$ such that $e < e_1$ and $f' < f_1$. Since $[\![\pi]\!]$ is non-increasing and $e < e_1$, we have $f_1 \leq f$. Since $f \notin [\![< \cdot \pi]\!](e)$, this implies $f_1 < f$. So there exists f_1 such that $f' < f_1 < f$ and $f_1 \in [\![\langle \pi^{-1}\rangle]\!]$, which contradicts $f' \in \left[\!\!\left[>_{\neg\langle \pi^{-1}\rangle}\right]\!\!\right](f)$.

Conversely, let $f' \in [\![(< \cdot \pi \cdot >)^{\mathsf{c}}]\!](e) \setminus [\![\pi \cdot \leq]\!](e)$. Since $f' \notin [\![\pi \cdot \leq]\!](e)$, we have $f' < f$. Let us show that

$$f' \in \left[\!\!\left[\pi \cdot >_{\neg\langle \pi^{-1}\rangle}\right]\!\!\right](e) \,, \quad \text{i.e.,} \quad f' \in \left[\!\!\left[>_{\neg\langle \pi^{-1}\rangle}\right]\!\!\right](f) \,.$$

Suppose towards a contradiction that this is not the case, i.e., there exists $(e_1, f_1) \in [\![\pi]\!]$ such that $f' < f_1 < f$. Since $[\![\pi]\!]$ is non-increasing and $f_1 < f$, we have $e < e_1$, and thus $(e, f) \in [\![< \cdot \pi \cdot >]\!]$, a contradiction. □

Lemma 3.43. *For all event formulas $\varphi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ and $\lambda, \mu \in \{\leq, <, \geq, >\}$, the formula*

$$(\lambda \cdot \{\varphi\}? \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Proof. We apply Lemma 3.34 with $\pi = \mu$. So we only need to prove that for all $\lambda, \mu \in \{\leq, <, \geq, >\}$,

$$(\lambda \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas. We have

$$\begin{array}{ll}
(\leq \cdot \leq)^{\mathsf{c}} \equiv > & (\leq \cdot \geq)^{\mathsf{c}} \equiv \{\mathit{false}\}? \\
(\leq \cdot <)^{\mathsf{c}} \equiv \geq & (\leq \cdot >)^{\mathsf{c}} \equiv \top \cdot \{\neg\langle < \rangle\}? \\
(< \cdot \leq)^{\mathsf{c}} \equiv \geq & (< \cdot \geq)^{\mathsf{c}} \equiv \{\neg\langle < \rangle\}? \cdot \top \\
(< \cdot <)^{\mathsf{c}} \equiv \geq + <_{\mathit{false}} & (< \cdot >)^{\mathsf{c}} \equiv \{\neg\langle < \rangle\}? \cdot \top + \top \cdot \{\neg\langle < \rangle\}?\,,
\end{array}$$

and the remaining cases are obtained by taking the converse of all formulas. □

Lemma 3.44. *For all event formulas $\varphi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ and $\lambda, \mu \in \{\leq, <, \geq, >\}$, the formula*

$$(\lambda \cdot <_\varphi \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Proof. Again, we need to distinguish several cases.

Case $(\lambda, \mu) \in \{(\leq, \geq), (<, \geq), (\leq, >), (<, >)\}$. Let us show that in that case,

$$(\lambda \cdot <_\varphi \cdot \mu)^{\mathsf{c}} \equiv \{\neg\langle \lambda \cdot <_\varphi \cdot \mu \rangle\}? \cdot \top + \top \cdot \{\neg\langle (\lambda \cdot <_\varphi \cdot \mu)^{-1} \rangle\}?\,.$$

The right-to-left inclusion is easy to check. Conversely, let us show that for all $e \in [\![\langle \lambda \cdot <_\varphi \cdot \mu \rangle]\!]$ and $f \in [\![\langle (\lambda \cdot <_\varphi \cdot \mu)^{-1} \rangle]\!]$, we have $(e, f) \in [\![\lambda \cdot <_\varphi \cdot \mu]\!]$.

Since $e \in [\![\langle \lambda \cdot <_\varphi \cdot \mu \rangle]\!]$, there exist g_1, h_1, f_1 such that $e\ \lambda\ g_1$, $(g_1, h_1) \in [\![<_\varphi]\!]$, and $h_1\ \mu\ f_1$. Similarly, since $f \in [\![\langle (\lambda \cdot <_\varphi \cdot \mu)^{-1} \rangle]\!]$, there exist e_2, g_2, h_2 such that $e_2\ \lambda\ g_2$, $(g_2, h_2) \in [\![<_\varphi]\!]$, and $h_2\ \mu\ f$. We distinguish three cases:

- Suppose $g_1 \leq g_2$. Since $\lambda \in \{\leq, <\}$, $e\ \lambda\ g_1 \leq g_2$ implies $e\ \lambda\ g_2$. We then have

  $$e\ \lambda\ g_2\,, \quad (g_2, h_2) \in [\![<_\varphi]\!]\,, \quad h_2\ \mu\ f\,,$$

  and thus, $(e, f) \in [\![\lambda \cdot <_\varphi \cdot \mu]\!]$.

- Suppose $h_1 \geq h_2$. Similarly,

$$e \; \lambda \; g_1 \,, \quad (g_1, h_1) \in [\![<_\varphi]\!] \,, \quad h_1 \; \mu \; f \,,$$

and thus, $(e, f) \in [\![\lambda \cdot <_\varphi \cdot \mu]\!]$.

- Otherwise, we have $g_2 < g_1 < h_1 < h_2$, which implies $(g_1, h_2) \in [\![<_\varphi]\!]$. Then

$$e \; \lambda \; g_1 \,, \quad (g_1, h_2) \in [\![<_\varphi]\!] \,, \quad h_2 \; \mu \; f \,,$$

and thus, $(e, f) \in [\![\lambda \cdot <_\varphi \cdot \mu]\!]$.

Case $(\lambda, \mu) \in \{(\geq, \leq), (>, \leq), (\geq, <), (>, <)\}$. Similarly to the previous case, we have

$$(\lambda \cdot <_\varphi \cdot \mu)^{\mathsf{c}} \equiv \{\neg\langle \lambda \cdot <_\varphi \cdot \mu \rangle\}? \cdot \top + \top \cdot \{\neg\langle (\lambda \cdot <_\varphi \cdot \mu)^{-1} \rangle\}? \,.$$

Case $(\lambda, \mu) \in \{(\leq, \leq), (<, \leq), (\leq, <), (<, <)\}$. We can first observe that

$$\leq \cdot <_\varphi \cdot \leq \; \equiv \; \leq \cdot \{\langle <_\varphi \rangle\}? \cdot < \,. \tag{3.1}$$

Indeed, if $(e, f) \in [\![\leq \cdot <_\varphi \cdot \leq]\!]$, then there exist $e \leq g < h \leq f$ such that $(g, h) \in [\![<_\varphi]\!]$. We then have $e \leq g < f$ and $g \in [\![\langle <_\varphi \rangle]\!]$, hence $(e, f) \in [\![\leq \cdot \{\langle <_\varphi \rangle\}? \cdot <]\!]$. Conversely, if $(e, f) \in [\![\leq \cdot \{\langle <_\varphi \rangle\}? \cdot <]\!]$, then there exist g, h such that $e \leq g < f$ and $(g, h) \in [\![<_\varphi]\!]$. If $e \leq g < h \leq f$, then we are done. Otherwise, $e \leq g < f < h$, and we also have $(g, f) \in [\![<_\varphi]\!]$, hence $(e, f) \in [\![\leq \cdot <_\varphi \cdot \leq]\!]$.

By Lemma 3.43, we conclude that

$$(\leq \cdot <_\varphi \cdot \leq)^{\mathsf{c}} \equiv (\leq \cdot \{\langle <_\varphi \rangle\}? \cdot <)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Since $< \equiv < \cdot \leq$, Equation (3.1) also implies that

$$< \cdot <_\varphi \cdot \leq \; \equiv \; < \cdot \{\langle <_\varphi \rangle\}? \cdot < \,,$$

and again, by Lemma 3.43, $(< \cdot <_\varphi \cdot \leq)^{\mathsf{c}}$ is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

Finally, by Equation (3.1),

$$\leq \cdot <_\varphi \cdot < \; \equiv \; \leq \cdot \{\langle <_\varphi \rangle\}? \cdot < \cdot < \qquad \text{and} \qquad < \cdot <_\varphi \cdot < \; \equiv \; < \cdot \{\langle <_\varphi \rangle\}? \cdot < \cdot < \,,$$

and by Lemma 3.34, all we need to conclude is to check that for all $\lambda \in \{\leq, <, \geq, >\}$, the formula

$$(\lambda \cdot < \cdot <)^{\mathsf{c}}$$

is equivalent to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas. This is straightforward:

$$(\leq \cdot < \cdot <)^{\mathsf{c}} \equiv \geq + <_{\mathit{false}} \qquad (< \cdot < \cdot <)^{\mathsf{c}} \equiv \geq + <_{\mathit{false}} + <_{\mathit{false}} \cdot <_{\mathit{false}}$$
$$(\geq \cdot < \cdot <)^{\mathsf{c}} \equiv \top \cdot \{\neg\langle > \cdot > \rangle\}? \qquad (> \cdot < \cdot <)^{\mathsf{c}} \equiv \top \cdot \{\neg\langle > \cdot > \rangle\}? + \{\neg\langle > \rangle\}? \cdot \top \,.$$

Case $(\lambda, \mu) \in \{(\geq, \geq), (>, \geq), (\geq, >), (>, >)\}$. We have

$$\begin{aligned}
(\geq \cdot <_\varphi \cdot \geq)^{\mathsf{c}} &\equiv \{\neg \mathsf{Loop}(\geq \cdot <_\varphi \cdot \geq)\}? \cdot \leq + < \cdot \{\neg\varphi\}? \cdot < + \\
&\quad \{\neg\langle <_\varphi\rangle \wedge \neg\langle >_\varphi\rangle\}? \cdot >_{\neg\langle >_\varphi\rangle} \cdot \{\neg\langle >_\varphi\rangle\}? \\
(> \cdot <_\varphi \cdot \geq)^{\mathsf{c}} &\equiv \{\neg \mathsf{Loop}(> \cdot <_\varphi \cdot \geq)\}? \cdot \leq + \leq \cdot \{\neg\varphi\}? \cdot < + \\
&\quad \{\neg\langle >_\varphi\rangle\}? \cdot >_{\neg\langle >_\varphi\rangle} \cdot \{\neg\langle >_\varphi\rangle\}? \\
(\geq \cdot <_\varphi \cdot >)^{\mathsf{c}} &\equiv \{\neg \mathsf{Loop}(\geq \cdot <_\varphi \cdot >)\}? \cdot \leq + < \cdot \{\neg\varphi \vee \neg\langle <_\varphi\rangle\}? \cdot \leq + \\
&\quad \{\neg\langle <_\varphi\rangle \wedge \neg\langle >_\varphi\rangle\}? \cdot >_{\neg\langle >_\varphi\rangle} \\
(> \cdot <_\varphi \cdot >)^{\mathsf{c}} &\equiv \{\neg \mathsf{Loop}(> \cdot <_\varphi \cdot >)\}? \cdot \leq + \leq \cdot \{\neg\varphi \vee \neg\langle <_\varphi\rangle\}? \cdot \leq + \\
&\quad \{\neg\langle >_\varphi\rangle\}? \cdot >_{\neg\langle >_\varphi\rangle} .
\end{aligned}$$

We prove the first equivalence, the others are similar.

Let $e \leq f$. If there is no $f_1 \geq e$ such that $(e, f_1) \in [\![\geq \cdot <_\varphi \cdot \geq]\!]$, i.e., if $(e, e) \notin [\![\geq \cdot <_\varphi \cdot \geq]\!]$, then $(e, f) \in [\![(\geq \cdot <_\varphi \cdot \geq)^{\mathsf{c}}]\!]$ and $(e, f) \in [\![\{\neg \mathsf{Loop}(\geq \cdot <_\varphi \cdot \geq)\}? \cdot \leq]\!]$. Now, assume that $e \leq f$, and that $e \in [\![\mathsf{Loop}(\geq \cdot <_\varphi \cdot \geq)]\!]$. Let us show that in that case, $(e, f) \in [\![(\geq \cdot <_\varphi \cdot \geq)^{\mathsf{c}}]\!]$ if and only if $(e, f) \in [\![< \cdot \{\neg\varphi\}? \cdot <]\!]$, i.e.,

$$(e, f) \in [\![\geq \cdot <_\varphi \cdot \geq]\!] \quad \iff \quad \forall e'.\, (e < e' < f \implies e' \in [\![\varphi]\!]) .$$

If $e = f$, then both are true since $e \in [\![\mathsf{Loop}(\geq \cdot <_\varphi \cdot \geq)]\!]$. If $e < f$, then the right-to-left inclusion follows from the fact that $[\![<_\varphi]\!] \subseteq [\![\geq \cdot <_\varphi \cdot \geq]\!]$. Conversely, if $e < f$ and $(e, f) \in [\![\geq \cdot <_\varphi \cdot \geq]\!]$, then there exist g, h such that

$$g \leq e < f \leq h \qquad \text{and} \qquad (g, h) \in [\![<_\varphi]\!] .$$

Therefore, we also have $(e, f) \in [\![<_\varphi]\!]$.

Finally, let us prove that if $f < e$, then $(e, f) \in [\![(\geq \cdot <_\varphi \cdot \geq)^{\mathsf{c}}]\!]$ if and only if $(e, f) \in [\![\{\neg\langle <_\varphi\rangle \wedge \neg\langle >_\varphi\rangle\}? \cdot >_{\neg\langle >_\varphi\rangle} \cdot \{\neg\langle >_\varphi\rangle\}?]\!]$, i.e.,

$$(e, f) \in [\![\geq \cdot <_\varphi \cdot \geq]\!] \iff \begin{cases} e \in [\![\langle <_\varphi\rangle]\!] \\ \text{or} \\ \exists h_1, f \leq h_1 \leq e \wedge h_1 \in [\![\langle >_\varphi\rangle]\!] . \end{cases}$$

Assume that $(e, f) \in [\![\geq \cdot <_\varphi \cdot \geq]\!]$. There exist g, h such that

$$e \geq g , \quad (g, h) \in [\![<_\varphi]\!] , \quad h \geq f .$$

If $h \leq e$, then we can take $h_1 = h$: we have

$$f \leq h \leq e \quad \text{and} \quad h \in [\![\langle >_\varphi\rangle]\!] .$$

Otherwise, $e < h$, and since $g \leq e < h$ and $(g, h) \in [\![<_\varphi]\!]$, we also have $(e, h) \in [\![<_\varphi]\!]$. Hence, $e \in [\![\langle <_\varphi\rangle]\!]$.

Conversely, if there exists h such that $(e,h) \in [\![<_\varphi]\!]$, then we have

$$e \geq e\,, \quad (e,h) \in [\![<_\varphi]\!]\,, \quad h \geq f\,,$$

hence $(e,f) \in [\![\geq \cdot <_\varphi \cdot \geq]\!]$. If there exist g_1, h_1 such that $f \leq h_1 \leq e$ and $(g_1,h_1) \in [\![<_\varphi]\!]$, then

$$e \geq g_1\,, \quad (g_1,h_1) \in [\![<_\varphi]\!]\,, \quad h_1 \geq f\,,$$

hence $(e,f) \in [\![\geq \cdot <_\varphi \cdot \geq]\!]$. □

3.8.5 Proof of Theorem 3.30

We now have all the ingredients to prove Theorem 3.30.

Proof of Theorem 3.30. We show that every $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ path formula is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas, which implies 1. and 2.

The only "missing" atomic path formula in $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ is $\leq$, and we have

$$\leq \equiv \{\mathit{true}\}? + <_{\mathit{true}}\,.$$

Moreover, finite unions of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas are closed (up to equivalence) under converse and concatenation. Therefore, we only need to prove that for all finite union π of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas, for all $\lambda, \mu \in \{\leq, \geq\}$,

$$(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas. We generalize this to $\lambda, \mu \in \{\leq, <, \geq, >\}$. By Lemma 3.36, we only need to prove that this is the case for $\pi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$, rather than finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ formulas.

So, let us show by induction on π that for all $\pi \in \mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$, for all $\lambda, \mu \in \{\leq, <, \geq, >\}$,

$$(\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$$

is equivalent, over complete interval-preserving Σ-structures, to a finite union of $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ path formulas.

For $\pi = \alpha$, we apply Lemma 3.38. For $\pi = \inf(\lambda \cdot \alpha)$ or $\pi = \inf(\lambda \cdot \alpha)$, we apply Lemma 3.39 or 3.41. For $\pi = \sigma^{-1}$, we apply the induction hypothesis on σ, and the fact that

$$(\lambda \cdot \sigma^{-1} \cdot \mu)^{\mathsf{c}} \equiv \left(\left(\mu^{-1} \cdot \sigma \cdot \lambda^{-1}\right)^{\mathsf{c}}\right)^{-1}\,.$$

For $\pi = \{\varphi\}?$, we apply Lemma 3.43. For $\pi = <_\varphi$, we apply Lemma 3.44. For $\pi = >_\varphi$, we apply the result for $<_\varphi$, and the fact that $>_\varphi \equiv (<_\varphi)^{-1}$. For $\pi = \pi_1 \cdot \pi_2$, we apply Lemma 3.35. □

Chapter 4

Communicating Finite-State Machines

This chapter provides preliminaries on communicating finite-state machines (CFMs). In a CFM, a fixed number of finite-state processes communicate by exchanging messages through unbounded FIFO channels [14]. A CFM accepts/generates message-sequence charts (MSCs), which are similar to UML's sequence diagrams [4] and standardized by the International Telecommunication Union [53]. MSCs are equipped with Lamport's happened-before relation: an event e happens before an event f if there is a "message flow" path from e to f [58]. Additional binary predicates connect (i) the emission of a message with its reception, and (ii) successive events executed by one and the same process.

We first define MSCs, then CFMs, and give an overview of known logic-automata connections for CFMs.

4.1 Message Sequence Charts

4.1.1 Definition

Set of processes Procs *and of channels* Ch

We consider message-passing systems in which processes communicate through unbounded FIFO channels. We fix a nonempty finite set of *processes* Procs and a nonempty finite set of *atomic propositions* Prop, with $\mathsf{Procs} \cap \mathsf{Prop} = \emptyset$. For all $p, q \in \mathsf{Procs}$ such that $p \neq q$, there is a channel (p, q) that allows p to send messages to q. The set of channels is denoted Ch.

The executions of these systems are represented by MSCs, which are labeled graphs such as in Figure 4.1. Nodes of the graph are *events* executed by processes, and partitioned according to their process location. In addition, each event is labeled by a set of atomic propositions, which may provide more information such as "enter critical section" or "output some value". There are two types of edges, describing causal dependencies between events: process edges and message edges. An MSC also has to satisfy some conditions, such as FIFO behavior and acyclicity.

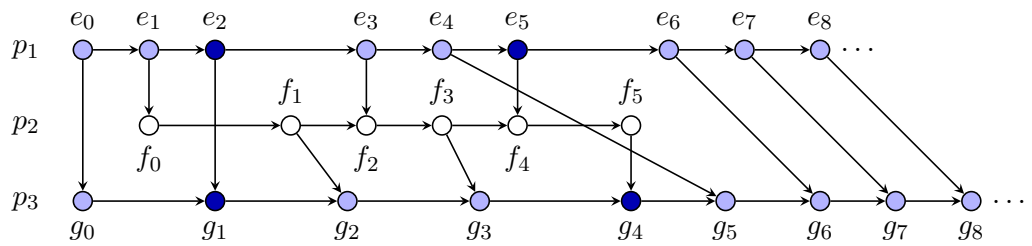


Figure 4.1: An (infinite) message sequence chart (MSC)

Message Sequence Chart

Definition 4.1. A *message sequence chart (MSC)* over Procs and Prop is a structure $M = (E, \rightarrow, \lhd, loc, \lambda)$, where

- E is a nonempty, finite or countably infinite set of *events*,
- $\rightarrow \subseteq E \times E$ is a set of *process edges*,
- $\lhd \subseteq E \times E$ is a set of *message edges*,
- $loc : E \rightarrow \mathsf{Procs}$ maps each event $e \in E$ to its *process location*, that is, the process which executes e,
- $\lambda : E \rightarrow 2^{\mathsf{Prop}}$ is a *labeling function* indicating the set of atomic propositions which hold at a given event.

Relations $\leq_{\mathsf{proc}}, \leq$

For a process $p \in \mathsf{Procs}$, we denote by $E_p := \{e \in E \mid loc(e) = p\}$ the set of events located on p. We call $\leq_{\mathsf{proc}} := \rightarrow^*$ the *process order*, and $\leq := (\rightarrow \cup \lhd)^*$ the *happened-before* relation. We also write $<_{\mathsf{proc}} := \rightarrow^+$ and $< := (\rightarrow \cup \lhd)^+$. We require that the following holds:

- $\rightarrow$ connects successive events on a same process, that is, $\rightarrow \subseteq \bigcup_{p \in \mathsf{Procs}} E_p \times E_p$, and each process is sequential: the restriction of $\leq_{\mathsf{proc}}$ to E_p is a total order, and $\rightarrow$ is its direct successor relation. In addition, every event $e \in E$ has a "finite past", i.e., $\{f \in E \mid f \leq_{\mathsf{proc}} e\}$ is finite.
- For all $(e, f) \in \lhd$, we have $(loc(e), loc(f)) \in Ch$, and each event is part of at most one message edge. If $e \lhd f$, we call e a *send event* and f a *receive event*. An event which is neither a send nor a receive event is called *internal*.

FIFO assumption

- Message edges respect a FIFO behavior: for all $(p, q) \in Ch$ and $(e, f), (e', f') \in \lhd \cap (E_p \times E_q)$, we have $e \leq e'$ if and only if $f \leq f'$.
- $\leq$ is a partial order (intuitively, messages cannot travel backwards in time).

Let $\mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ denote the set of MSCs over Procs and Prop. An MSC is *finite* if its set of events E is finite, and *infinite* otherwise. We denote the set of finite MSCs by $\mathbb{MSC}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})$, and the set of infinite MSCs by $\mathbb{MSC}^{\omega}(\mathsf{Procs}, \mathsf{Prop})$.

Note than in an infinite MSC, at least one of the processes executes infinitely many events, but some other process may execute finitely many, that is, E_p may be finite or infinite.

It is worth noting that, when Procs is a singleton, an MSC with events $e_1 \to e_2 \to e_3 \to \dots$ can be identified with the (finite or infinite) word $\lambda(e_1)\lambda(e_2)\lambda(e_3)\dots$ over 2^{Prop}.

Example 4.2. Consider the (infinite) MSC from Figure 4.1 over $\mathsf{Procs} = \{p_1, p_2, p_3\}$ and $\mathsf{Prop} = \{P, Q\}$. We denote labels $\lambda(e)$ of events e with $\bullet = \{P, Q\}$, $\circ = \{P\}$, and $\circ = \emptyset$. For instance, $\lambda(e_3) = \circ = \{P\}$, while $\lambda(f_5) = \emptyset$. Process locations are indicated on the left: $E_{p_1} = \{e_i \mid i \in \mathbb{N}\}$, $E_{p_2} = \{f_0, \dots, f_5\}$, $E_{p_3} = \{g_i \mid i \in \mathbb{N}\}$. The process relation is given by $e_i \to e_{i+1}$ and $g_i \to g_{i+1}$ for all $i \in \mathbb{N}$, as well as $f_i \to f_{i+1}$ for all $i \in \{0, \dots, 4\}$. Concerning the message relation, we have $e_1 \lhd f_0$, $e_4 \lhd g_5$, etc. Moreover, $e_2 \le f_3$, but neither $e_2 \le f_1$ nor $f_1 \le e_2$.

4.1.2 Logics for MSCs

For any set $\mathsf{Rel} \subseteq \{\lhd, \to, \le_{\mathsf{proc}}, \le\}$, every MSC $M = (E, \to, \lhd, \mathit{loc}, \lambda)$ induces a $(\mathsf{Procs} \cup \mathsf{Prop}, \mathsf{Rel})$-structure. We use the same symbols for the name and interpretation of relations in Rel. The interpretation of unary predicates is given by $P^M = \{e \in E \mid P \in \lambda(e)\}$ for $P \in \mathsf{Prop}$, and $p^M = E_p$ for $p \in \mathsf{Procs}$.

For each signature $\Sigma = (\mathsf{Procs} \cup \mathsf{Prop}, \mathsf{Rel})$ with $\mathsf{Rel} \subseteq \{\lhd, \to, \le_{\mathsf{proc}}, \le\}$, we can consider the logics $\mathrm{MSO}[\Sigma]$, $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, etc. For readability, we write for instance $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \lhd, \to]$ instead of $\mathrm{FO}[\mathsf{Procs} \cup \mathsf{Prop}, \{\lhd, \to\}]$, and similarly for other logics. We also simply write $M \models \Phi$ or $M, \nu \models \Phi$ to denote the fact that the Σ-structure associated with M satisfies Φ, and define similarly $[\![\varphi]\!]^M$, $[\![\pi]\!]^M$ for $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ or $\mathrm{ICPDL}[\Sigma]$ formulas φ and π. For a sentence Φ in $\mathrm{MSO}[\Sigma]$, $\mathrm{ICPDL}[\Sigma]$ or $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$, we let $\mathbb{L}(\Phi) = \{M \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop}) \mid M \models \Phi\}$ be the *language* of Φ.

Given $\mathcal{F} \in \{\mathrm{FO}, \mathrm{EMSO}, \mathrm{MSO}, \mathrm{PDL}, \mathrm{ICPDL}, \mathrm{PDL}_{\mathsf{sf}}\}$ and $\Sigma = (\mathsf{Procs} \cup \mathsf{Prop}, \mathsf{Rel})$, we denote by $\mathcal{L}(\mathcal{F}[\Sigma]) = \{\mathbb{L}(\Phi) \mid \Phi \in \mathcal{F}[\Sigma] \text{ is a sentence}\}$ the set of $\mathcal{F}[\Sigma]$-definable MSC languages.

It is easy to see that some of these logics have the same expressive power. For instance, since transitive closure is definable in MSO,

$$\mathcal{L}(\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \to, \lhd, \le_{\mathsf{proc}}, \le]) = \mathcal{L}(\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \to, \lhd])\,.$$

On the other hand, since $\mathrm{FO}[\mathsf{Prop}, \to] \subsetneq \mathrm{FO}[\mathsf{Prop}, \le]$ already for words (which correspond to the case $|\mathsf{Procs}| = 1$), we have

$$\mathcal{L}(\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \to, \lhd]) \subsetneq \mathcal{L}(\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \le, \lhd])\,.$$

Moreover, it was shown in [13] that the message relation cannot be defined from other predicates:

$$\mathcal{L}(\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq]) \subsetneq \mathcal{L}(\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]) \,.$$

It is worth noting that $\leq_{\mathsf{proc}}$ can be recovered from $\leq$, and vice versa:

$$\mathcal{L}(\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]) = \mathcal{L}(\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]) \,.$$

More precisely, we have the following translations:

Lemma 4.3. *Let $k \in \mathbb{N}$.*

1. *For every formula $\Phi \in \mathrm{FO}^k[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$, there exists a formula $\Psi \in \mathrm{FO}^k[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ such that, over MSCs, $\Phi \equiv \Psi$.*

2. *For every formula $\Phi \in \mathrm{FO}^k[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$, there exists a formula $\Psi \in \mathrm{FO}^{\max(3,k)}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ such that, over MSCs, $\Phi \equiv \Psi$.*

Proof. The first translation is obtained by observing that over MSCs,

$$x \leq_{\mathsf{proc}} y \quad \equiv \quad x \leq y \wedge \bigvee_{p \in \mathsf{Procs}} p(x) \wedge p(y) \,.$$

For the other direction, notice that if there is a path from an event e to f, then there is one which enters and leaves each process at most once. Therefore,

$$x \leq y \quad \equiv \quad \bigvee_{1 \leq n \leq |\mathsf{Procs}|} \exists x_1, y_1, \ldots, x_n, y_n . \, x_1 = x \wedge y_n = y \wedge \bigwedge_{1 \leq i \leq n} x_i \leq_{\mathsf{proc}} y_i \wedge \bigwedge_{1 \leq i < n} y_i \lhd x_{i+1} \,.$$

The latter formula can be easily rewritten into an $\mathrm{FO}^3[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ formula (thus matching the number of variables claimed in Lemma 4.3), by alternating quantifications over x and over a variable $z \notin \{x, y\}$, instead of using a new name for each of the variables $x_1, y_1, \ldots, x_n, y_n$. For instance, if $|\mathsf{Procs}| = 2$,

$$x \leq y \quad \equiv \quad x \leq_{\mathsf{proc}} y \vee \exists z . \, (x \leq_{\mathsf{proc}} z \wedge \exists x . (z \lhd x \wedge x \leq_{\mathsf{proc}} y)) \,. \qquad \square$$

4.1.3 Bounded MSCs

Most decision problems related to MSCs, such as satisfiability or validity of a first-order formula, or emptiness of communicating finite-state machines (defined in the next section), are undecidable in the general case. It is then natural to consider decidable restrictions of these problems. One possible restriction is to assume a

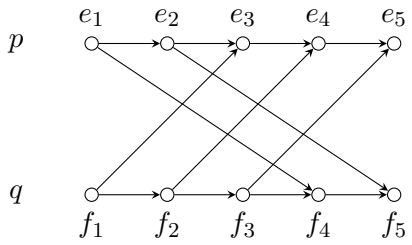


Figure 4.2: An existentially 2-bounded MSC

bound on the channel capacities [61, 44, 34]. There are in fact two classes of MSCs corresponding to this assumption: *existentially-bounded* and *universally-bounded* MSCs.

Bounded MSCs are defined in terms of linearizations:

linearization

Definition 4.4. A *linearization* of an MSC $M = (E, \rightarrow, \lhd, loc, \lambda)$ is a total order $\preceq \subseteq E \times E$ extending $\leq$ and of order type at most ω, i.e., a total order such that $\leq \subseteq \preceq$ and $\{e \in E \mid e \preceq f\}$ is finite for all $e \in E$. For $B \in \mathbb{N}$, a linearization $\preceq$ is *B-bounded* if, for all $g \in E$ and $(p, q) \in Ch$, $|\{(e, f) \in \lhd \cap (E_p \times E_q) \mid e \preceq g \prec f\}| \leq B$.

In other terms, a linearization $\preceq$ is B-bounded if the number of pending messages in any channel (p, q) never exceeds B if events occur in the order defined by $\preceq$. There are (at least) two natural definitions of bounded MSCs:

$\exists B/\forall B$-bounded MSC

Definition 4.5. An MSC M is *existentially B-bounded* ($\exists B$-bounded) if M has some B-bounded linearization. It is *universally B-bounded* ($\forall B$-bounded) if all its linearizations are B-bounded.

The set of $\exists B$-bounded MSCs is denoted by $\mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$, and the set of $\forall B$-bounded MSCs by $\mathbb{MSC}_{\forall B}(\mathsf{Procs}, \mathsf{Prop})$. Moreover, we let

$$\mathbb{MSC}^{\mathrm{fin}}_{\exists B}(\mathsf{Procs}, \mathsf{Prop}) := \mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop}) \cap \mathbb{MSC}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})$$
$$\mathbb{MSC}^{\mathrm{fin}}_{\forall B}(\mathsf{Procs}, \mathsf{Prop}) := \mathbb{MSC}_{\forall B}(\mathsf{Procs}, \mathsf{Prop}) \cap \mathbb{MSC}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})$$
$$\mathbb{MSC}^{\omega}_{\exists B}(\mathsf{Procs}, \mathsf{Prop}) := \mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop}) \cap \mathbb{MSC}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})$$
$$\mathbb{MSC}^{\omega}_{\forall B}(\mathsf{Procs}, \mathsf{Prop}) := \mathbb{MSC}_{\forall B}(\mathsf{Procs}, \mathsf{Prop}) \cap \mathbb{MSC}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})\,.$$

Example 4.6. The MSC from Figure 4.2 is $\exists 2$-bounded, as witnessed by the linearization $e_1 \prec e_2 \prec f_1 \prec e_3 \prec f_2 \prec e_4 \prec f_3 \prec e_5 \prec f_4 \prec f_5$. It is also $\forall 3$-bounded, but not $\forall 2$-bounded, since the linearization $e_1 \prec e_2 \prec f_1 \prec f_2 \prec f_3 \prec f_4 \prec f_5 \prec e_3 \prec e_4 \prec e_5$ is not 2-bounded.

Example 4.7. The MSC from Figure 4.1 is $\exists 1$-bounded, but it is not $\forall B$-bounded, no matter what B is.

4.2 Communicating finite-state machines

actions $\langle a\rangle$, $\langle a, !_q m\rangle$, $\langle a, ?_q m\rangle$

In a communicating finite-state machine [14], each process $p \in \mathsf{Procs}$ is represented by a finite-state transition system. It can perform internal actions of the form $\langle a\rangle$, where $a \in 2^{\mathsf{Prop}}$, or send/receive messages from a finite set of messages Msg. A send action $\langle a, !_q m\rangle$ of process p writes message $m \in Msg$ to channel (p, q), and performs $a \in 2^{\mathsf{Prop}}$. A receive action $\langle a, ?_q m\rangle$ reads message m from channel (q, p). Accordingly, we let

$$Act_p(Msg) := \{\langle a\rangle \mid a \in 2^{\mathsf{Prop}}\} \cup \{\langle a, !_q m\rangle \mid a \in 2^{\mathsf{Prop}}, m \in Msg, q \in \mathsf{Procs} \setminus \{p\}\} \cup \{\langle a, ?_q m\rangle \mid a \in 2^{\mathsf{Prop}}, m \in Msg, q \in \mathsf{Procs} \setminus \{p\}\}$$

denote the set of possible actions of process p.

communicating finite-state machine (CFM)

Definition 4.8. A *communicating finite-state machine (CFM)* over Procs and Prop is a tuple $\mathcal{A} = ((\mathcal{A}_p)_{p\in\mathsf{Procs}}, Msg, Acc)$ consisting of a finite set of messages Msg and a finite-state transition system $\mathcal{A}_p = (S_p, \iota_p, \Delta_p)$ for each process $p \in \mathsf{Procs}$, with finite set of states S_p, initial state $\iota_p \in S_p$, and transition relation $\Delta_p \subseteq S_p \times Act_p(Msg) \times S_p$. Moreover, we have an acceptance condition Acc, which is a *positive* boolean combination of atomic conditions $\langle p, s\rangle$ or $\langle p, s\rangle_\infty$ where $p \in \mathsf{Procs}$ and $s \in S_p$.

acceptance condition

Intuitively, $\langle p, s\rangle$ requires that process p terminates in state s (and, thus, executes only finitely many events), while $\langle p, s\rangle_\infty$ requires that process p enters state s infinitely often (which implies that p executes infinitely many events). This kind of "mixed" acceptance condition is quite convenient. Other, syntactically different acceptance criteria have been adopted in the literature, like Büchi or Muller conditions [56, 11]. However, they are all expressively equivalent. Note that here we consider nondeterministic automata.

Given a transition $t = (s, act, s') \in \Delta_p$, we let $source(t) = s$ and $target(t) = s'$ denote the source and target states of t. In addition, if $act = \langle a\rangle$, then t is an *internal transition* and we let $label(t) = a$. If $act = \langle a, !_q m\rangle$, then t is a *send transition* and we let $label(t) = a$, $msg(t) = m$, and $receiver(t) = q$. Finally, if $act = \langle a, ?_q m\rangle$, then t is a *receive transition* with $label(t) = a$, $msg(t) = m$, and $sender(t) = q$.

run of a CFM

A *run* ρ of $\mathcal{A}$ on an MSC $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ is a mapping associating with each event $e \in E_p$ a transition $\rho(e) \in \Delta_p$, and satisfying the following conditions:

1. for all events $e \in E$, we have $label(\rho(e)) = \lambda(e)$,
2. for all $\rightarrow$-minimal events $e \in E$, we have $source(\rho(e)) = \iota_p$, where $p = loc(e)$,
3. for all process edges $(e, f) \in \rightarrow$, we have $target(\rho(e)) = source(\rho(f))$,

4. for all internal events $e \in E$, $\rho(e)$ is an internal transition, and

5. for all message edges $e \lhd f$, $\rho(e)$ and $\rho(f)$ are respectively send and receive transitions such that $msg(\rho(e)) = msg(\rho(f))$, $receiver(\rho(e)) = loc(f)$, and $sender(\rho(f)) = loc(e)$.

We say that ρ is *accepting* if it satisfies the acceptance condition Acc, written $\rho \models Acc$. The relation $\rho \models Acc$ is defined inductively. Disjunction and conjunction are interpreted as usual. Moreover, we let $\rho \models \langle p, s\rangle$ if either $E_p = \emptyset$ and $s = \iota_p$, or E_p is a nonempty finite set and $s = target(\rho(e))$ where e is the last event of E_p. Finally, $\rho \models \langle p, s\rangle_\infty$ if $s = target(\rho(e))$ for infinitely many events $e \in E_p$ (which implies that E_p is infinite).

The *language* $\mathbb{L}(\mathcal{A})$ of $\mathcal{A}$ is the set of MSCs M such that there exists an accepting run of $\mathcal{A}$ on M. Moreover,

$$\mathcal{L}(\mathrm{CFM}[\mathsf{Procs}, \mathsf{Prop}]) := \{\mathbb{L}(\mathcal{A}) \mid \mathcal{A} \text{ is a CFM over } \mathsf{Procs} \text{ and } \mathsf{Prop}\}\,.$$

Following [44, 35, 56], we call a CFM $\mathcal{A} = ((\mathcal{A}_p)_{p\in\mathsf{Procs}}, Msg, Acc)$ *deterministic* if, for all processes p and transitions $t_1 = (s_1, act_1, s_1')$ and $t_2 = (s_2, act_2, s_2')$ of $\mathcal{A}_p$ such that $s_1 = s_2$ and $label(t_1) = label(t_2)$, the following hold:

- If t_1 and t_2 are internal transitions, then $s_1' = s_2'$.
- If t_1 and t_2 are send transitions such that $receiver(t_1) = receiver(t_2)$, then $s_1' = s_2'$ and $msg(t_1) = msg(t_2)$.
- If t_1 and t_2 are receive transitions such that $sender(t_1) = sender(t_2)$ and $msg(t_1) = msg(t_2)$, then $s_1' = s_2'$.

In particular, this implies that, for each MSC, there is at most one run.

Example 4.9. Consider the simple (deterministic) CFM $\mathcal{A}$ depicted in Figure 4.3. The set of processes is $\mathsf{Procs} = \{p_1, p_2, p_3\}$. Moreover, we have $Msg = \{\circ, \bullet\}$ and $\{\circ, \bullet, \bigcirc\} \subseteq 2^{\mathsf{Prop}}$. Process p_1 sends messages to p_2 and p_3. Each message can be either $\circ$ or $\bullet$, and the message sent is made "visible" in the labeling of the MSC. Process p_2 simply forwards every message it receives to p_3. In any case, the action is $\bigcirc$. Finally, p_3 receives and "outputs" messages from p_1 and p_2 in any order. Note that, in this example, there are no local transitions, i.e., every transition is either sending or receiving. As acceptance condition, we take $Acc = \langle p, s_{p_1}\rangle_\infty$, which says that p_1 executes infinitely many events. The MSC from Figure 4.1 is accepted by $\mathcal{A}$.

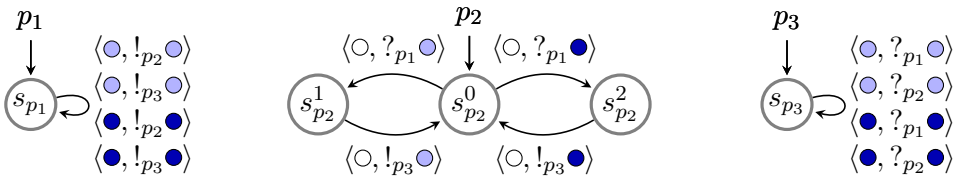


Figure 4.3: A communicating finite-state machine

4.3 Logical characterizations of CFMs

This section gives an overview of known logic-automata relations for CFMs.

Theorem 4.10 ([15, 27, 90])**.** *Suppose* $|\mathsf{Procs}| = 1$ *(i.e., CFMs are essentially finite automata). We have* $\mathcal{L}(\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq]) = \mathcal{L}(\mathrm{CFM}[\mathsf{Procs}, \mathsf{Prop}])$.

This classical result is known as the Büchi-Elgot-Trakhtenbrot theorem. It was first generalized to CFMs with universally bounded channels:

Theorem 4.11 ([44])**.** *For all* $B \in \mathbb{N}$ *and* $L \subseteq \mathbb{MSC}^{\mathrm{fin}}_{\forall B}(\mathsf{Procs}, \mathsf{Prop})$*, the following are equivalent:*

1. $L = \mathbb{L}(\mathcal{A})$ *for some CFM* $\mathcal{A}$*;*
2. $L = \mathbb{L}(\mathcal{A})$ *for some deterministic CFM* $\mathcal{A}$*;*
3. $L = \mathbb{L}(\Phi)$ *for some* $\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ *formula* Φ*.*

Moreover, there is a deterministic CFM $\mathcal{A}$ *such that* $\mathbb{L}(\mathcal{A}) = \mathbb{MSC}^{\mathrm{fin}}_{\forall B}(\mathsf{Procs}, \mathsf{Prop})$.

Kuske generalized the theorem to *infinite* universally bounded MSCs, while using a different proof technique.

Theorem 4.12 ([56])**.** *For all* $B \in \mathbb{N}$ *and* $L \subseteq \mathbb{MSC}_{\forall B}(\mathsf{Procs}, \mathsf{Prop})$*, the following are equivalent:*

1. $L = \mathbb{L}(\mathcal{A})$ *for some CFM* $\mathcal{A}$*;*
2. $L = \mathbb{L}(\mathcal{A})$ *for some deterministic CFM* $\mathcal{A}$*;*
3. $L = \mathbb{L}(\Phi)$ *for some* $\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ *formula* Φ*.*

In the case of finite MSCs, the logical characterization was lifted to existentially bounded MSCs by Genest et al.

Theorem 4.13 ([34])**.** *For all* $B \in \mathbb{N}$ *and* $L \subseteq \mathbb{MSC}^{\mathrm{fin}}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$*, the following are equivalent:*

1. $L = \mathbb{L}(\mathcal{A})$ *for some CFM* $\mathcal{A}$*;*

2. $L = \mathbb{L}(\Phi)$ *for some* $\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ *formula* Φ*.*

Moreover, there is a CFM $\mathcal{A}$ *such that* $\mathbb{L}(\mathcal{A}) = \mathbb{MSC}^{\mathrm{fin}}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$*.*

On the other hand, it turns out that deterministic CFMs are now strictly weaker:

Theorem 4.14 ([34])**.** *CFMs are inherently non-deterministic: There is a CFM* $\mathcal{A}$ *such that* $\mathbb{L}(\mathcal{A}) \subseteq \mathbb{MSC}^{\mathrm{fin}}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$ *and, for all deterministic CFMs* $\mathcal{A}'$*, we have* $\mathbb{L}(\mathcal{A}) \neq \mathbb{L}(\mathcal{A}')$*.*

The proofs of Theorems 4.11, 4.12, and 4.13 reduce message-passing systems to finite-state shared-memory systems so that involved results from Mazurkiewicz trace theory [24] can be applied. This generic approach is no longer applicable when the restriction on the channel capacity is dropped. In fact, in general, CFMs do not capture MSO logic:

Theorem 4.15 ([13])**.** *For all* $L \subseteq \mathbb{MSC}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})$*, the following are equivalent:*

1. $L = \mathbb{L}(\mathcal{A})$ *for some CFM* $\mathcal{A}$*;*

2. $L = \mathbb{L}(\Phi)$ *for some sentence* $\Phi \in \mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$*;*

However, $\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$ *is strictly more expressive than CFMs: There is an* $\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$ *sentence* Φ *such that* $\mathbb{L}(\Phi) \subseteq \mathbb{MSC}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})$ *and, for all CFMs* $\mathcal{A}$*, we have* $\mathbb{L}(\Phi) \neq \mathbb{L}(\mathcal{A})$*.*

The characterization from Theorem 4.15 was given for finite MSCs. Over infinite MSCs, $\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$ is strictly weaker than CFMs as it cannot express that *there are infinitely many events* to satisfy some property. This is already true for one process, i.e., finite automata and words. However, CFMs can be characterized by the logic $\mathrm{EMSO}^{\infty}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$, extending $\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$ by the quantifier $\exists^{\infty} x.\Phi$, which requires that there be *infinitely many events* x such that Φ holds.

Theorem 4.16 ([11])**.** *For all* $L \subseteq \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$*, the following are equivalent:*

1. $L = \mathbb{L}(\mathcal{A})$ *for some CFM* $\mathcal{A}$*;*

2. $L = \mathbb{L}(\Phi)$ *for some sentence* $\Phi \in \mathrm{EMSO}^{\infty}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$*.*

In terms of PDL, the following relations are known:

Theorem 4.17 ([12])**.** *1. For all sentence* $\xi \in \mathrm{PDL}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$*, there exists a CFM* $\mathcal{A}$ *such that* $\mathbb{L}(\xi) = \mathbb{L}(\mathcal{A})$*.*

2. *There exists a sentence* $\xi \in \mathrm{ICPDL}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$ *such that* $\mathbb{L}(\xi) \neq \mathbb{L}(\mathcal{A})$ *for all CFMs* $\mathcal{A}$.

Note that the happened-before relation is expressible in $\mathrm{PDL}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$ or $\mathrm{ICPDL}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$): we have $\leq \equiv (\lhd + \rightarrow)^*$.

Contributions. Theorem 4.15 shows that any sentence in $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$ or $\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd]$ can be translated into an equivalent CFM. However, until recently, it was open whether such a translation was still possible when the happened-before relation $\leq$ is added to the logic. We first answered this question positively for two-variable logics:

Theorem 4.18 ([8])**.** *For all* $L \subseteq \mathbb{MSC}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})$*, the following are equivalent:*

1. $L = \mathbb{L}(\mathcal{A})$ *for some CFM* $\mathcal{A}$*;*
2. $L = \mathbb{L}(\Phi)$ *for some sentence* $\Phi \in \mathrm{EMSO}^2[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd, \leq]$.

We solved the general case in [9] for finite MSCs, and in [10] for infinite MSCs:

Theorem 4.19 ([9, 10])**.** *For all* $L \subseteq \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ *the following are equivalent:*

1. $L = \mathbb{L}(\mathcal{A})$ *for some CFM* $\mathcal{A}$*;*
2. $L = \mathbb{L}(\Phi)$ *for some sentence* $\Phi \in \mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd, \leq]$.

More precisely, we introduced a variant $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ of PDL, that can be seen both as a fragment of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ or as a fragment of $\mathrm{ICPDL}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$, and we proved that

$$\mathcal{L}(\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd, \leq]) = \mathcal{L}(\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]) \subseteq \mathcal{L}(\mathrm{CFM}[\mathsf{Procs}, \mathsf{Prop}])\,.$$

The expressive power of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ and $\mathrm{PDL}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ are incomparable, so this is orthogonal to what was previously known about the relation between variants of PDL and CFMs (Theorem 4.17).

The proof of these results are spread over Chapters 5 and 6. We will not give more details about Theorem 4.18, since it is generalized by Theorem 4.19.

Another contribution concerns the case of existentially-bounded MSCs. Using Theorem 4.19, we give a new proof of Theorem 4.13, and extend it to infinite MSCs:

Theorem 4.20 ([10])**.** *For all* $B \in \mathbb{N}$ *and* $L \subseteq \mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$*, the following are equivalent:*

1. $L = \mathbb{L}(\mathcal{A})$ *for some CFM* $\mathcal{A}$*;*
2. $L = \mathbb{L}(\Phi)$ *for some* $\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ *formula* Φ.

The proof of Theorem 4.20 is presented in Chapter 6, Section 6.4.

Chapter 5

Logics for Message Sequence Charts

This chapter investigates the expressive power of several logics for MSCs. Our aim is to identify alternatives to first-order logic as a specification language, of similar expressive power but with better complexities. While these logics are interesting on their own, this is also a key step towards the translation from first-order logic to communicating finite-state machines presented in the next chapter.

We first show that results from Chapter 3 can be applied to MSCs. We then prove that the syntax of the resulting variant of star-free PDL can be simplified to remove all boolean operators from path formulas, including implicit intersections in the form of Loop formulas. Up to projection, the resulting logic is still equivalent to first-order logic; this is the main result of the chapter. Finally, we present an(other) extension of LTL to MSCs, based on classical temporal modalities for partial orders, and give a direct translation from this logic into star-free PDL.

5.1 MSCs as interval-preserving structures

Contrarily to interval-preserving structures which are linearly ordered, MSCs are naturally seen as partial orders, ordered by the happened-before relation $\leq$. In order to apply the results from Chapter 3, we need to equip MSCs with a linear order. To do so, we simply concatenate, in a fixed, arbitrary order, the total orders induced by $\leq_{\mathsf{proc}}$ on each process. Note that the resulting linear order is not a very meaningful relation for MSCs: it is simply a technical tool which, in a way, allows us to reason about the process order as if it were a linear order.

extension of $\leq_{\mathsf{proc}}$ into $\sqsubseteq$

Extension of the process order into a total order. Let us define more formally this linear order. We fix an arbitrary total order $\sqsubseteq_{\mathsf{Procs}}$ on Procs with strict part $\sqsubset_{\mathsf{Procs}}$. For all MSCs $M = (E, \rightarrow, \lhd, \mathit{loc}, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$, we then define a total

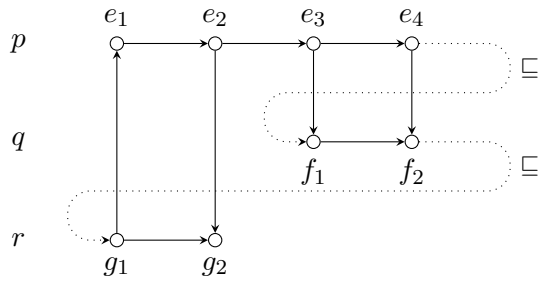


Figure 5.1: A possible extension of $\leq_{\mathsf{proc}}$ into a linear order $\sqsubseteq$

order $\sqsubseteq$ over E as follows: given two events $e, f \in E$, we let

$$e \sqsubseteq f \qquad \text{if} \qquad e \leq_{\mathsf{proc}} f \;\text{ or }\; loc(e) \sqsubset_{\mathsf{Procs}} loc(f)\,.$$

We denote by $\sqsubset$ the strict part of $\sqsubseteq$.

Note that $\sqsubseteq$ is not compatible with $\leq$: in general, there may be events e, f such that $e \leq f$ but $f \sqsubset e$. For instance, in the MSC from Figure 5.1, with the ordering $p \sqsubset_{\mathsf{Procs}} q \sqsubset_{\mathsf{Procs}} r$, we have $g_1 \leq e_1$ but $e_1 \sqsubset g_1$. Note also that we use the same order $\sqsubset_{\mathsf{Procs}}$ for all MSCs in $\mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$.

Interval-preserving message relations. The message relation $\lhd$ is not always interval-preserving with respect to $\sqsubseteq$. For instance, in the MSC from Figure 5.1, and using the ordering $p \sqsubset_{\mathsf{Procs}} q \sqsubset_{\mathsf{Procs}} r$, we have

$$f_1 \sqsubseteq f_2 \sqsubseteq g_2\,, \quad (e_3, f_1), (e_2, g_2) \in \lhd\,, \quad f_2 \in \lhd^{-1}(E)\,, \quad \text{but } f_2 \notin \lhd([e_2, e_3])\,.$$

However, the restriction of $\lhd$ to a particular channel is interval-preserving, thanks to the FIFO assumption. Given an MSC $M = (E, \rightarrow, \lhd, loc, \lambda)$ and a channel $(p, q) \in Ch$, we let

$$\lhd_{p,q} = \lhd \cap (E_p \times E_q)\,.$$

Lemma 5.1. *For all $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ and $(p, q) \in Ch$, $\lhd_{p,q}$ is interval-preserving (with respect to $\sqsubseteq$).*

Proof. Since the channels are FIFO, for all events e, f, g, e', f', g' such that $e \lhd_{p,q} e'$, $f \lhd_{p,q} f'$, and $g \lhd_{p,q} g'$, we have:

- if $e' \sqsubseteq f' \sqsubseteq g'$, then $e \sqsubseteq f \sqsubseteq g$. Therefore, for all intervals I of $(E, \preceq)$, $\lhd_{p,q}(I)$ is an interval of $(\lhd_{p,q}(E), \sqsubseteq)$.
- if $e \sqsubseteq f \sqsubseteq g$, then $e' \sqsubseteq f' \sqsubseteq g'$. Therefore, for all intervals I of $(E, \preceq)$, $\lhd_{p,q}^{-1}(I)$ is an interval of $(\lhd_{p,q}^{-1}(E), \sqsubseteq)$. □

Alternative signature for MSCs. We denote by $\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}$ the signature $\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}$

$$\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})} := (\mathsf{Procs} \cup \mathsf{Prop}, \{\sqsubseteq\} \cup \{\lhd_{p,q} \mid (p,q) \in Ch\})\,.$$

As an immediate consequence of Lemma 5.1, we have:

Lemma 5.2. *For all MSCs $M = (E, \to, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$, the induced $\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}$-structure*

$$\Big(E, (E_p)_{p\in\mathsf{Procs}}, (\llbracket P \rrbracket^M)_{P\in\mathsf{Prop}}, \sqsubseteq, (\lhd_{p,q})_{(p,q)\in Ch}\Big)$$

is a complete interval-preserving $\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}$-structure.

Note that $\leq_{\mathsf{proc}}$ (and therefore, by Lemma 4.3, $\leq$) can easily be recovered from $\sqsubseteq$, and vice versa:

Lemma 5.3. *Let $k \in \mathbb{N}$.*

1. *For every formula $\Phi \in \mathrm{FO}^k[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$, there exists a formula $\Psi \in \mathrm{FO}^k\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ such that over MSCs, $\Phi \equiv \Psi$.*

2. *For every formula $\Phi \in \mathrm{FO}^k\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$, there exists a formula $\Psi \in \mathrm{FO}^k[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ such that over MSCs, $\Phi \equiv \Psi$.*

Proof. Over $\mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$, we have

$$x \leq_{\mathsf{proc}} y \quad\equiv\quad x \sqsubseteq y \wedge \bigvee_{p\in\mathsf{Procs}} p(x) \wedge p(y)$$

and

$$x \sqsubseteq y \quad\equiv\quad x \leq_{\mathsf{proc}} y \vee \bigvee_{\substack{p,q\in\mathsf{Procs}\\ p\sqsubset_{\mathsf{Procs}} q}} p(x) \wedge q(y)\,.$$

For the message relation,

$$x \lhd y \quad\equiv\quad \bigvee_{(p,q)\in Ch} x \lhd_{p,q} y$$

and

$$x \lhd_{p,q} y \quad\equiv\quad x \lhd y \wedge p(x) \wedge q(y)\,. \qquad \square$$

Strong 3-variable properties for MSCs. As a direct application of Theorem 3.16, we have:

Corollary 5.4. *Over MSCs, any formula in* $\mathrm{FO}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ *is equivalent to a finite boolean combination of formulas in* $\mathrm{FO}^3\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$.

Recall that the translations given in Lemmas 5.3 and 4.3 between the logics $\mathrm{FO}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$, $\mathrm{FO}[\mathsf{Procs},\mathsf{Prop},\leq_{\mathsf{proc}},\lhd]$, and $\mathrm{FO}[\mathsf{Procs},\mathsf{Prop},\leq,\lhd]$ do not increase the number of variables used in a formula beyond 3. Together with Corollary 5.4, this implies that MSCs also have the strong 3-variable property when seen as $(\mathsf{Procs}\cup\mathsf{Prop},\{\leq_{\mathsf{proc}},\lhd\})$- or $(\mathsf{Procs}\cup\mathsf{Prop},\{\leq,\lhd\})$-structures:

Corollary 5.5. *Over MSCs, any formula in* $\mathrm{FO}[\mathsf{Procs},\mathsf{Prop},\leq_{\mathsf{proc}},\lhd]$ *is equivalent to a finite boolean combination of formulas in* $\mathrm{FO}^3[\mathsf{Procs},\mathsf{Prop},\leq_{\mathsf{proc}},\lhd]$.

Corollary 5.6. *Over MSCs, any formula in* $\mathrm{FO}[\mathsf{Procs},\mathsf{Prop},\leq,\lhd]$ *is equivalent to a finite boolean combination of formulas in* $\mathrm{FO}^3[\mathsf{Procs},\mathsf{Prop},\leq,\lhd]$.

Alternatively, this can be formulated in terms of $\mathrm{PDL}_{\mathsf{sf}}$:

Corollary 5.7. *Over MSCs, every formula* $\Phi\in\mathrm{FO}[\mathsf{Procs},\mathsf{Prop},\leq_{\mathsf{proc}},\lhd]$ *with at least one free variable is equivalent to a finite boolean combination of formulas of the form* $\widetilde{\pi}(x,y)$*, where* $x,y\in\mathrm{Free}(\Phi)$ *and* $\pi\in\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop},\leq_{\mathsf{proc}},\lhd]$.

Proof. We first translate the formula Φ into a $\mathrm{FO}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ formula, then apply Theorem 3.14. Each $\mathrm{PDL}_{\mathsf{sf}}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ path formula is then translated into $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop},\leq_{\mathsf{proc}},\lhd]$ as before:

$$\begin{aligned}\lhd_{p,q} &\equiv \{p\}?\cdot\lhd\cdot\{q\}?\\ \sqsubseteq &\equiv \leq_{\mathsf{proc}}+\sum_{p\sqsubset_{\mathsf{Procs}}q}\{p\}?\cdot\top\cdot\{q\}?\,,\end{aligned}$$

where $\top=\{\mathit{true}\}?+(\{\mathit{true}\}?)^{\mathsf{c}}$. □

5.2 Fragment of star-free PDL for MSCs

We have seen in Corollary 5.7 that first-order logic and star-free PDL have the same expressive power over MSCs. In addition, $\sqsubseteq$ is a complete order, so we could also apply Theorem 3.30 to obtain an equivalent logic without a complement operator. In fact, we can use an even simpler fragment, defined below.

5.2.1 Syntax

We define a fragment of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ where the complement of path formulas is not allowed (though we still have $\mathsf{Loop}(\pi)$ formulas). Instead, we add path formulas $(<_{\mathsf{proc}})_\varphi$ and $(>_{\mathsf{proc}})_\varphi$ which are similar to LTL *until* and *since* modalities, and local to a given process. This fragment can be seen as a variant of the logic $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}$ defined in Section 3.8, where formulas $<_\varphi$ are defined with respect to the process order $\leq_{\mathsf{proc}}$ rather than the total order $\sqsubseteq$. To maintain the possibility to move from one process to another, we also explicitly include $\top$ in the syntax of path formulas. For technical reasons, we choose to restrict the converse operation to atomic path formulas, though the converses of other formulas are still expressible.

Definition 5.8. Let $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ be the set of formulas $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}$

$$\begin{aligned}
\xi &::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi \\
\varphi &::= \mathit{true} \mid p \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi) \\
\pi &::= \top \mid \lhd_{p,q} \mid \lhd^{-1}_{p,q} \mid \{\varphi\}? \mid (<_{\mathsf{proc}})_\varphi \mid (>_{\mathsf{proc}})_\varphi \mid \pi \cdot \pi\,,
\end{aligned}$$

where $p \in \mathsf{Procs}$, $P \in \mathsf{Prop}$, $(p,q) \in \mathit{Ch}$.

The semantics of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ is inherited from the semantics of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}, (\lhd_{p,q})_{(p,q)\in \mathit{Ch}}]$, with additionally

$$\begin{aligned}
\Big[\!\Big[(<_{\mathsf{proc}})_\varphi\Big]\!\Big]^M &= \Big\{(e,f) \mid e <_{\mathsf{proc}} f \wedge \forall e <_{\mathsf{proc}} g <_{\mathsf{proc}} f, g \in [\![\varphi]\!]^M\Big\} \\
\Big[\!\Big[(>_{\mathsf{proc}})_\varphi\Big]\!\Big]^M &= \Big\{(e,f) \mid f <_{\mathsf{proc}} e \wedge \forall f <_{\mathsf{proc}} g <_{\mathsf{proc}} e, g \in [\![\varphi]\!]^M\Big\}\,.
\end{aligned}$$

We use the abbreviations $<_{\mathsf{proc}} := (<_{\mathsf{proc}})_{\mathit{true}}$ and $>_{\mathsf{proc}} := (>_{\mathsf{proc}})_{\mathit{true}}$, as well as $\leq_{\mathsf{proc}} := \{\mathit{true}\}? + <_{\mathsf{proc}}$, and $\geq_{\mathsf{proc}} := \{\mathit{true}\}? + >_{\mathsf{proc}}$. Note that $\leq_{\mathsf{proc}}$ and $\geq_{\mathsf{proc}}$ are not in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, but they are finite unions of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ path formulas. We also use formulas $\rightarrow := (<_{\mathsf{proc}})_{\mathit{false}}$ and $\leftarrow := (>_{\mathsf{proc}})_{\mathit{false}}$ to express the process successor relation $\rightarrow$ and its converse.

Remark 5.9. The logic $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ can indeed be seen as a fragment of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$, since

$$\lhd_{p,q} \equiv \{p\}? \cdot \lhd \cdot \{q\}? \quad \text{and} \quad (<_{\mathsf{proc}})_\varphi \equiv <_{\mathsf{proc}} \cap (<_{\mathsf{proc}} \cdot \{\neg\varphi\}? \cdot <_{\mathsf{proc}})^{\mathsf{c}}\,,$$

where $<_{\mathsf{proc}} \equiv \leq_{\mathsf{proc}} \cap (\{\mathit{true}\}?)^{\mathsf{c}}$.

It is also closely related to $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$: the main difference between the two logics is that $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ uses formulas $\sqsubset_\varphi$ and $\sqsupset_\varphi$ defined in terms of the total order $\sqsubseteq$, whereas $(<_{\mathsf{proc}})_\varphi$ and $(>_{\mathsf{proc}})_\varphi$ are defined in terms of $\leq_{\mathsf{proc}}$. In a way, $(<_{\mathsf{proc}})_\varphi$ and $(>_{\mathsf{proc}})_\varphi$ can be seen as restrictions of $\sqsubset_\varphi$ and $\sqsupset_\varphi$:

$$(<_{\mathsf{proc}})_\varphi \equiv \sum_{p\in\mathsf{Procs}} \sqsubset_{\varphi\wedge p} \qquad \text{and} \qquad (>_{\mathsf{proc}})_\varphi \equiv \sum_{p\in\mathsf{Procs}} \sqsupset_{\varphi\wedge p}\,.$$

In addition, $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ also contains path formulas $\mathsf{inf}\ (\lambda \cdot \alpha)$ and $\mathsf{sup}\ (\lambda \cdot \alpha)$, where $\lambda \in \{\sqsubset, \sqsubseteq, \sqsupseteq, \sqsupset\}$ and $\alpha \in \{\lhd_{p,q}, \lhd^{-1}_{p,q} \mid (p,q) \in Ch\}$. However, over MSCs, these formulas are redundant.

Remark 5.10. Any $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ formula constructed without $\top$ is expressible in ICPDL[$\mathsf{Procs}, \mathsf{Prop}, \lhd, \rightarrow$]. Indeed, we have

$$(<_{\mathsf{proc}})_\varphi \ \equiv\ (\rightarrow \cdot \{\varphi\}?)^* \cdot \rightarrow \qquad \text{and} \qquad (>_{\mathsf{proc}})_\varphi \ \equiv\ (\rightarrow^{-1} \cdot \{\varphi\}?)^* \cdot \rightarrow^{-1}.$$

However, with the formula $\top$, it is possible to move from one connected component of the MSC to another, which is not possible with ICPDL[$\mathsf{Procs}, \mathsf{Prop}, \lhd, \rightarrow$] path formulas. On the other hand, at the level of sentences, one can also talk about several connected components of the MSC in ICPDL[$\mathsf{Procs}, \mathsf{Prop}, \lhd, \rightarrow$]. In fact, every sentence $\xi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ is equivalent to some sentence $\xi' \in$ ICPDL[$\mathsf{Procs}, \mathsf{Prop}, \lhd, \rightarrow$]. To eliminate every occurrence of $\top$ in ξ, we can first observe that

$$\langle \pi_1 \cdot \top \cdot \pi_2 \rangle\, \varphi \equiv \langle \pi_1 \rangle \wedge \langle \top \rangle\, \langle \pi_2 \rangle\, \varphi \qquad \text{and} \qquad \mathsf{Loop}(\pi_1 \cdot \top \cdot \pi_2) \equiv \langle \pi_1 \rangle \wedge \langle \pi_2^{-1} \rangle.$$

Therefore, at the level of event formulas, we only need $\top$ in subformulas of the form $\langle \top \rangle\, \varphi$. Moreover, the value of $\langle \top \rangle\, \varphi$ is the same at every event, and is the same as the value of the sentence $\mathsf{E}\, \varphi$. So, at the level of sentences, we can enumerate all possible values of subformulas $\langle \top \rangle\, \varphi$ and, in each case, replace them within event formulas by *true* or *false*.

Size of a formula. There are several ways to measure the size of a formula. Here, we choose not to count duplicates of sentence or event subformulas. This will give us more precise complexities in the logic-to-automata translations of the next chapter, since the important parameter for such constructions is the number of subformulas in the input formula, rather than the size of the formula seen as a character string.

set of subformulas $\mathcal{S}(\xi)$

The sets $\mathcal{S}(\xi)$, $\mathcal{S}(\varphi)$ or $\mathcal{S}(\pi)$ of (sentence or event) subformulas of a formula in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ is defined inductively as follows:

$$\begin{aligned}
\mathcal{S}(\mathsf{E}\, \varphi) &= \{\mathsf{E}\, \varphi\} \cup \mathcal{S}(\varphi) & \qquad \mathcal{S}(\xi_1 \vee \xi_2) &= \{\xi_1 \vee \xi_2\} \cup \mathcal{S}(\xi_1) \cup \mathcal{S}(\xi_2) \\
\mathcal{S}(\neg \xi) &= \{\neg \xi\} \cup \mathcal{S}(\xi) & & \\
\mathcal{S}(\mathit{true}) &= \{\mathit{true}\} & \qquad \mathcal{S}(\top) &= \emptyset \\
\mathcal{S}(p) &= \{p\} & \qquad \mathcal{S}(\lhd_{p,q}) &= \emptyset \\
\mathcal{S}(P) &= \{P\} & \qquad \mathcal{S}(\lhd^{-1}_{p,q}) &= \emptyset \\
\mathcal{S}(\varphi_1 \vee \varphi_2) &= \{\varphi_1 \vee \varphi_2\} \cup \mathcal{S}(\varphi_1) \cup \mathcal{S}(\varphi_2) & \qquad \mathcal{S}(\{\varphi\}?) &= \mathcal{S}(\varphi) \\
\mathcal{S}(\neg \varphi) &= \{\neg \varphi\} \cup \mathcal{S}(\varphi) & \qquad \mathcal{S}((<_{\mathsf{proc}})_\varphi) &= \mathcal{S}(\varphi) \\
\mathcal{S}(\langle \pi \rangle\, \varphi) &= \{\langle \pi \rangle\, \varphi\} \cup \mathcal{S}(\pi) \cup \mathcal{S}(\varphi) & \qquad \mathcal{S}((>_{\mathsf{proc}})_\varphi) &= \mathcal{S}(\varphi) \\
\mathcal{S}(\mathsf{Loop}(\pi)) &= \{\mathsf{Loop}(\pi)\} \cup \mathcal{S}(\pi) & \qquad \mathcal{S}(\pi_1 \cdot \pi_2) &= \mathcal{S}(\pi_1) \cup \mathcal{S}(\pi_2)
\end{aligned}$$

Another parameter to the size of a formula is the length of the path formulas occurring in it. The length $\mathit{length}(\pi)$ of a path formulas π is defined inductively as follows: $\mathit{length}(\pi)$

$$\begin{aligned}\mathit{length}(\top) = \mathit{length}(\lhd_{p,q}) = \mathit{length}(\lhd_{p,q}^{-1}) = \mathit{length}(\{\varphi\}?)\\ = \mathit{length}((<_{\mathsf{proc}})_\varphi) = \mathit{length}((>_{\mathsf{proc}})_\varphi) = 1\end{aligned}$$
$$\mathit{length}(\pi_1 \cdot \pi_2) = \mathit{length}(\pi_1) + \mathit{length}(\pi_2)\,.$$

The "size" $\|\xi\|$ of a formula ξ is then defined as follows.

Definition 5.11. Given a set $\mathcal{F} \subseteq \mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$ of sentences and event formulas, we write

$$\|\mathcal{F}\| = |\mathcal{F}| + \sum_{\substack{\langle\pi\rangle\,\varphi \in \mathcal{F} \text{ or} \\ \mathsf{Loop}(\pi) \in \mathcal{F}}} \mathit{length}(\pi)\,.$$

For a formula ξ, φ or π in $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$, let $\|\xi\|$, $\|\varphi\|$, $\|\pi\|$

$$\|\xi\| = \|\mathcal{S}(\xi)\| \qquad \|\varphi\| = \|\mathcal{S}(\varphi)\| \qquad \|\pi\| = \|\mathcal{S}(\pi)\| + \mathit{length}(\pi)\,.$$

Remark 5.12. We do *not* have $\|\varphi\| = \|\{\varphi\}\|$. For instance, if

$$\varphi = P \vee \langle (<_{\mathsf{proc}})_Q \rangle\, R\,,$$

We have $\mathcal{S}(\varphi) = \{\varphi, P, \langle (<_{\mathsf{proc}})_Q \rangle\, R, Q, R\}$ and $\|\varphi\| = |\mathcal{S}(\varphi)| + \mathit{length}((<_{\mathsf{proc}})_Q) = 6$, but $\|\{\varphi\}\| = 1$.

Converse of a formula. Note that while the converse operation is not explicitly in the syntax of $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$, it is easily expressible.

Lemma 5.13. *For all path formulas $\pi \in \mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$, there exists a path formula $\pi' \in \mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$ such that for all $M \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$,*

$$[\![\pi']\!]^M = \left([\![\pi]\!]^M\right)^{-1}\,.$$

Moreover, $\mathit{length}(\pi') = \mathit{length}(\pi)$ and $\mathcal{S}(\pi') = \mathcal{S}(\pi)$.

Proof. This follows from the fact that

$$\top^{-1} \equiv \top\,, \quad (<_{\mathsf{proc}})_\varphi^{-1} \equiv (>_{\mathsf{proc}})_\varphi \quad \text{and} \quad (\pi_1 \cdot \pi_2)^{-1} \equiv \pi_2^{-1} \cdot \pi_1^{-1}\,. \qquad \square$$

When it is clear from the context that we are considering $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$ formulas, we simply denote the formula given by Lemma 5.13 by π^{-1}.

Compatible pairs of processes. Every path formula $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ induces a binary relation on Procs, composed of pairs (p, q) such that in some MSC, there is a path from process p to process q matching π. We define below a simple syntactic over-approximation of this relation.

$\mathsf{Comp}(\pi)$

Definition 5.14. For all path formulas $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, the set $\mathsf{Comp}(\pi) \subseteq \mathsf{Procs} \times \mathsf{Procs}$ of *compatible pairs of* π is defined as follows:

$$\begin{aligned}
\mathsf{Comp}(\top) &= \mathsf{Procs} \times \mathsf{Procs} \\
\mathsf{Comp}(\lhd_{p,q}) &= \{(p, q)\} \\
\mathsf{Comp}(\lhd_{p,q}^{-1}) &= \{(q, p)\} \\
\mathsf{Comp}(\{\varphi\}?) &= \mathsf{Comp}((<_{\mathsf{proc}})_{\varphi}) = \mathsf{Comp}((>_{\mathsf{proc}})_{\varphi}) = \mathsf{Id}_{\mathsf{Procs}} \\
\mathsf{Comp}(\pi_1 \cdot \pi_2) &= \mathsf{Comp}(\pi_1) \cdot \mathsf{Comp}(\pi_2)\,,
\end{aligned}$$

where $\mathsf{Id}_{\mathsf{Procs}} = \{(p, p) \mid p \in \mathsf{Procs}\}$.

Example 5.15. Consider $\pi = <_{\mathsf{proc}} \cdot \lhd_{p,q} \cdot \leftarrow \cdot \lhd_{q,r}$. We have $\mathsf{Comp}(\pi) = \{(p, r)\}$.

As explained above, $\mathsf{Comp}(\pi)$ contains all pairs of processes (p, q) such that there might exist events e on process p and f on process q with $(e, f) \in [\![\pi]\!]$:

Lemma 5.16. *For all* $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ *and* $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$,

$$[\![\pi]\!]^M \subseteq \bigcup_{(p,q) \in \mathsf{Comp}(\pi)} E_p \times E_q\,.$$

Proof. This follows from a trivial induction on π. □

A useful observation is that, except for formulas using $\top$ as an atomic step, $\mathsf{Comp}(\pi)$ is either empty, a singleton, or $\mathsf{Id}_{\mathsf{Procs}}$. In particular, it is deterministic, in the sense that for all process p, there is at most one process q with $(p, q) \in \mathsf{Comp}(\pi)$. Note that this is true even if $\top$ occurs in some event subformulas of π; we only need to exclude formulas of the form $\pi_1 \cdot \top \cdot \pi_2$.

$\top$*-free formula*

Definition 5.17. A formula $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ is $\top$*-free* if any occurrence of $\top$ in π occurs within an event subformula φ of π.

Example 5.18. The formula

$$(<_{\mathsf{proc}})_{\langle \top \rangle\, P} \cdot \{\langle \top \rangle\, Q\}? \cdot (<_{\mathsf{proc}})_P$$

is $\top$-free, but $\{P\}? \cdot \top$ is not.

Lemma 5.19. *For all* $\top$*-free formulas* $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ *and* $p \in \mathsf{Procs}$,

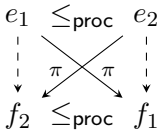
$$|\mathsf{Comp}(\pi)| \leq 1 \qquad \textit{or} \qquad \mathsf{Comp}(\pi) = \mathsf{Id}_{\mathsf{Procs}}\,.$$

Proof. We prove by induction on π that if π is $\top$-free, then $\mathsf{Comp}(\pi)$ is either empty, $\mathsf{Id}_{\mathsf{Procs}}$ or a singleton. This is true for all base cases. For concatenation, if $\mathsf{Comp}(\pi_1)$ or $\mathsf{Comp}(\pi_2)$ is empty, then so is $\mathsf{Comp}(\pi_1) \cdot \mathsf{Comp}(\pi_2)$. If $\mathsf{Comp}(\pi_1) = \mathsf{Id}_{\mathsf{Procs}}$, then $\mathsf{Comp}(\pi_1) \cdot \mathsf{Comp}(\pi_2) = \mathsf{Comp}(\pi_2)$, and similarly if $\mathsf{Comp}(\pi_2) = \mathsf{Id}_{\mathsf{Procs}}$. If $\mathsf{Comp}(\pi_1)$ and $\mathsf{Comp}(\pi_2)$ are singletons, then $\mathsf{Comp}(\pi_1) \cdot \mathsf{Comp}(\pi_2)$ is empty or a singleton. □

5.2.2 Monotonicity

MSCs enjoy stronger monotonicity properties than arbitrary interval-preserving structures. This allows us to derive stronger properties for $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ path formulas as well.

Lemma 5.20. *Let* $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$*, and* $(e_1, f_1), (e_2, f_2) \in [\![\pi]\!]$ *such that* $e_1 \leq_{\mathsf{proc}} e_2$ *and* $f_2 \leq_{\mathsf{proc}} f_1$*:*



Then $(e_1, f_2) \in [\![\pi]\!]$ *and* $(e_2, f_1) \in [\![\pi]\!]$*.*

Proof. We first consider the case where π is *not* $\top$-free, that is, $\pi \equiv \pi_1 \cdot \top \cdot \pi_2$ for some π_1, π_2. We then have $(e, f) \in [\![\pi]\!]$ if and only if $e \in [\![\langle \pi_1 \rangle]\!]$ and $f \in [\![\langle \pi_2^{-1} \rangle]\!]$. Therefore, $(e_1, f_2) \in [\![\pi]\!]$ and $(e_2, f_1) \in [\![\pi]\!]$.

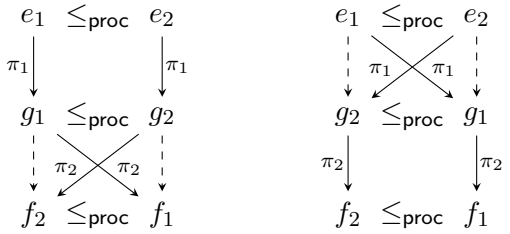
We now prove the result by induction for $\top$-free path formulas π. The cases $\pi = \lhd_{p,q}$ and $\pi = \lhd_{p,q}^{-1}$ are consequences of the FIFO assumption. If $\pi = \{\varphi\}?$, we have $e_1 \leq_{\mathsf{proc}} e_2 = f_2 \leq_{\mathsf{proc}} f_1 = e_1$, hence the four events are equal.

Assume $\pi = (<_{\mathsf{proc}})_\varphi$. We have

$$e_1 \leq_{\mathsf{proc}} e_2 <_{\mathsf{proc}} f_2 \leq_{\mathsf{proc}} f_1\,, \quad \text{and} \quad \forall e_1 <_{\mathsf{proc}} g <_{\mathsf{proc}} f_1\,, g \in [\![\varphi]\!]\,.$$

Therefore, $(e_1, f_2) \in \left[\!\!\left[(<_{\mathsf{proc}})_\varphi\right]\!\!\right]$ and $(e_2, f_1) \in \left[\!\!\left[(<_{\mathsf{proc}})_\varphi\right]\!\!\right]$. The case $\pi = (>_{\mathsf{proc}})_\varphi$ is symmetric.

Finally, assume $\pi = \pi_1 \cdot \pi_2$, where π_1 and π_2 are $\top$-free. There exist g_1 such that $(e_1, g_1) \in [\![\pi_1]\!]$ and $(g_1, f_1) \in [\![\pi_2]\!]$, and g_2 such that $(e_2, g_2) \in [\![\pi_1]\!]$ and $(g_2, f_2) \in [\![\pi_2]\!]$. Since $loc(e_1) = loc(e_2)$ and π_1 is $\top$-free, Lemma 5.19 implies $loc(g_1) = loc(g_2)$. Therefore, we have either $g_1 \leq_{\mathsf{proc}} g_2$ or $g_2 \leq_{\mathsf{proc}} g_1$:



In the first case, we apply the induction hypothesis for π_2: we have $g_1 \leq_{\mathsf{proc}} g_2$ and $f_2 \leq_{\mathsf{proc}} f_1$, hence $(g_1, f_2) \in [\![\pi_2]\!]$ and $(g_2, f_1) \in [\![\pi_2]\!]$. Therefore, $(e_1, f_2) \in [\![\pi_1 \cdot \pi_2]\!]$ and $(e_2, f_1) \in [\![\pi_1 \cdot \pi_2]\!]$. Similarly, in the case $g_2 \leq_{\mathsf{proc}} g_1$, we apply the induction hypothesis for π_1. □

Lemma 5.20 implies that every path formula is, in a sense, interval-preserving. In particular, $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ path formulas satisfy the following property.

Lemma 5.21. *Let* $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$.

1. *For all* $(e, f_1), (e, f_2) \in [\![\pi]\!]$ *and* $f \in [\![\langle \pi^{-1} \rangle]\!]$ *such that* $f_1 \leq_{\mathsf{proc}} f \leq_{\mathsf{proc}} f_2$, *we have* $(e, f) \in [\![\pi]\!]$.

2. *For all* $(e_1, f), (e_2, f) \in [\![\pi]\!]$ *and* $e \in [\![\langle \pi \rangle]\!]$ *such that* $e_1 \leq_{\mathsf{proc}} e \leq_{\mathsf{proc}} e_2$, *we have* $(e, f) \in [\![\pi]\!]$.

Proof. We prove the first part of the Lemma, the second one is symmetric. If π is not $\top$-free, that is, $\pi \equiv \pi_1 \cdot \top \cdot \pi_2$ for some π_1 and π_2, then we have $e \models \langle \pi_1 \rangle$ and $f \models \langle \pi_2^{-1} \rangle$, which implies $(e, f) \in [\![\pi]\!]$.

Assume π is $\top$-free. Since $f \in [\![\langle \pi^{-1} \rangle]\!]$, there exists e' such that $(e', f) \in [\![\pi]\!]$. By Lemma 5.19, we must have $loc(e) = loc(e')$, hence $e \leq_{\mathsf{proc}} e'$ or $e' \leq_{\mathsf{proc}} e$:

In the first case, we apply Lemma 5.20 to e, e', f, f_2, and in the second case, to e', e, f_1, f. In both cases, we obtain $(e, f) \in [\![\pi]\!]$. □

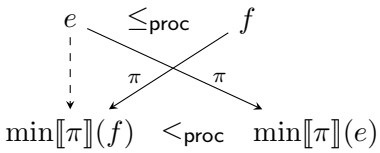
For a $\top$-free formula π, Lemma 5.19 implies that the image $[\![\pi]\!](e)$ of an event e is always included in some process p. In particular, $[\![\pi]\!](e)$ is completely ordered by $\leq_{\mathsf{proc}}$. If $[\![\pi]\!](e) \neq \emptyset$, we denote by $\min[\![\pi]\!](e)$ its minimal element with respect to $\leq_{\mathsf{proc}}$. The partial function mapping e to $\min[\![\pi]\!](e)$ is monotone:

Lemma 5.22. *For all* $\top$*-free path formulas* $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ *and events* e, f *such that* $[\![\pi]\!](e) \neq \emptyset$ *and* $[\![\pi]\!](f) \neq \emptyset$, *we have*

$$e \leq_{\mathsf{proc}} f \quad \Longrightarrow \quad \min[\![\pi]\!](e) \leq_{\mathsf{proc}} \min[\![\pi]\!](f)\,.$$

Proof. Since π is $\top$-free and $loc(e) = loc(f)$, $\min[\![\pi]\!](e)$ and $\min[\![\pi]\!](f)$ are well-defined and located on a same process. Therefore, we have $\min[\![\pi]\!](e) \leq_{\mathsf{proc}} \min[\![\pi]\!](f)$ or $\min[\![\pi]\!](f) <_{\mathsf{proc}} \min[\![\pi]\!](e)$.

Suppose towards a contradiction that $\min[\![\pi]\!](f) <_{\mathsf{proc}} \min[\![\pi]\!](e)$:



We have $(e, \min[\![\pi]\!](e)) \in [\![\pi]\!]$ and $(f, \min[\![\pi]\!](f)) \in [\![\pi]\!]$, and, by Lemma 5.20, we obtain $(e, \min[\![\pi]\!](f)) \in [\![\pi]\!]$. This contradicts the minimality of $\min[\![\pi]\!](e)$. □

In addition, this partial function mapping e to $\min[\![\pi]\!](e)$ can be defined in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$.

Lemma 5.23. *For all $\top$-free path formulas $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, one can construct a formula $\mathsf{min}\ \pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ such that*

$$[\![\mathsf{min}\ \pi]\!] = \{(e, f) \mid f = \min[\![\pi]\!](e)\}$$

and satisfying the following:

- *If π contains no $\mathsf{Loop}(\pi')$ subformulas, then neither does $\mathsf{min}\ \pi$.*
- *$\mathit{length}(\mathsf{min}\ \pi) = \mathcal{O}(\mathit{length}(\pi))$.*
- *$\|\mathcal{S}(\mathsf{min}\ \pi) \setminus \mathcal{S}(\pi)\| = \mathcal{O}(\mathit{length}(\pi))$.*

In particular, $\|\mathsf{min}\ \pi\| = \mathcal{O}(\|\pi\|)$.

Proof. A detailed proof of the lemma is given below. As a quick summary, the idea is to apply Lemma 5.22, and define $\mathsf{min}\ \pi$ inductively:

$$\begin{aligned}
\mathsf{min}\ (r \cdot \pi) &= r \cdot \mathsf{min}\ \pi \qquad \text{for } r = \lhd_{p,q}, r = \lhd^{-1}_{p,q} \text{ or } r = \{\varphi\}?\\
\mathsf{min}\ ((<_{\mathsf{proc}})_\varphi \cdot \pi) &= (<_{\mathsf{proc}})_{\varphi \wedge \neg\langle\pi\rangle} \cdot \mathsf{min}\ \pi\\
\mathsf{min}\ ((>_{\mathsf{proc}})_\varphi \cdot \pi) &= (>_{\mathsf{proc}})_\varphi \cdot \{\neg\varphi \vee \neg\langle(>_{\mathsf{proc}})_\varphi \cdot \pi\rangle\}? \cdot \mathsf{min}\ \pi\,.
\end{aligned}$$

Let us move on to the detailed proof. Since concatenation of binary relations is associative, we can assume that π is of the form $\pi = r_1 \cdot (r_2 \cdots (r_n) \cdots)$, where each r_i is of the form $\lhd_{p,q}$, $\lhd^{-1}_{p,q}$, $\{\varphi\}?$, $(<_{\mathsf{proc}})_\varphi$, or $(>_{\mathsf{proc}})_\varphi$. We take the convention of right associativity, and simply write this formula as $\pi = r_1 \cdots r_n$. Note that bringing π into this form has no impact on $\mathit{length}(\pi)$ or $\mathcal{S}(\pi)$, nor therefore $\|\pi\|$ (cf. Definition 5.11).

Without loss of generality, we assume that the last atomic step r_n of the path formula is $\{\mathit{true}\}?$. Again, this increases $\mathit{length}(\pi)$ and $|\mathcal{S}(\pi)|$ by at most one, and has no impact on the results concerning the size of $\mathsf{min}\ \pi$.

We define $\mathsf{min}\ \pi$ by induction on the length n of the sequence $\pi = r_1 \cdots r_n$. According to the previous assumption, the basis of the induction is $\pi = \{\mathit{true}\}?$, in

which case we let $\mathsf{min}\ \pi = \{true\}?$. For the inductive cases, we first define $\mathsf{min}\ \pi$, and then discuss its size.

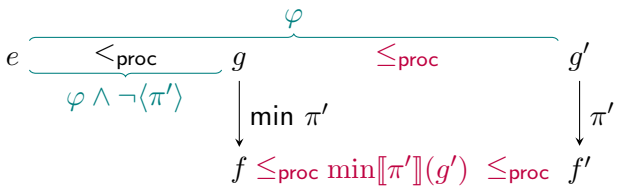
Case 1: $\pi = r \cdot \pi'$, with $r = \lhd_{p,q}$, $r = \lhd_{p,q}^{-1}$, or $r = \{\varphi\}?$. Notice that r is deterministic, in the sense that for all e, there is at most one f such that $(e, f) \in [\![r]\!]$. Therefore, we can define $\mathsf{min}\ \pi$ as

$$\mathsf{min}\ (r \cdot \pi') = r \cdot \mathsf{min}\ \pi' .$$

Case 2: $\pi = (<_{\mathsf{proc}})_\varphi \cdot \pi'$. We let

$$\mathsf{min}\ \left((<_{\mathsf{proc}})_\varphi \cdot \pi'\right) = (<_{\mathsf{proc}})_{\varphi \wedge \neg\langle\pi'\rangle} \cdot \mathsf{min}\ \pi' .$$

Let us show that $(e, f) \in [\![\mathsf{min}\ \pi]\!]$ if and only if $f = \min[\![\pi]\!](e)$. Let $(e, f) \in [\![\mathsf{min}\ \pi]\!]$. There exists g such that $(e, g) \in [\![(<_{\mathsf{proc}})_{\varphi \wedge \neg\langle\pi'\rangle}]\!]$ and $(g, f) \in [\![\mathsf{min}\ \pi']\!]$. Note that we also have $(e, g) \in [\![(<_{\mathsf{proc}})_\varphi]\!]$ and $(g, f) \in [\![\pi']\!]$, hence $f \in [\![\pi]\!](e)$. Let us show that f is minimal. Let $f' \in [\![\pi]\!](e)$. There exists g' such that $(e, g') \in [\![(<_{\mathsf{proc}})_\varphi]\!]$, and $(g', f') \in [\![\pi']\!]$.



Since $g' \models \langle\pi'\rangle$, we cannot have $e <_{\mathsf{proc}} g' <_{\mathsf{proc}} g$, which means that $g \leq_{\mathsf{proc}} g'$. By Lemma 5.22, we then have

$$f = \min [\![\pi']\!] (g) \leq_{\mathsf{proc}} \min [\![\pi']\!] (g') \leq_{\mathsf{proc}} f' ,$$

hence $f \leq_{\mathsf{proc}} f'$.

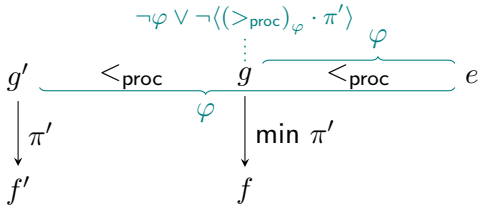
Conversely, suppose that $f = \min[\![\pi]\!](e)$. We have $[\![\pi]\!](e) \neq \emptyset$, which implies $[\![(<_{\mathsf{proc}})_\varphi \cdot \{\langle\pi'\rangle\}?]\!](e) \neq \emptyset$. Thus, $g = \min[\![(<_{\mathsf{proc}})_\varphi \cdot \{\langle\pi'\rangle\}?]\!](e)$ is well-defined. For all $e <_{\mathsf{proc}} h <_{\mathsf{proc}} g$, we have $h \in [\![\varphi]\!]$ since $(e, g) \in [\![(<_{\mathsf{proc}})_\varphi]\!]$, and $h \notin [\![\langle\pi'\rangle]\!]$ by minimality of g. Therefore, $(e, g) \in [\![(<_{\mathsf{proc}})_{\varphi \wedge \neg\langle\pi'\rangle}]\!]$. In addition, we have $[\![\pi']\!](g) \neq \emptyset$, hence there exists f' such that $(g, f') \in [\![\mathsf{min}\ \pi']\!]$. Then $(e, f') \in [\![\mathsf{min}\ \pi]\!]$, and by the previous implication, $f = f'$. Therefore, $(e, f) \in [\![\mathsf{min}\ \pi]\!]$.

Case 3: $\pi = (>_{\mathsf{proc}})_\varphi \cdot \pi'$. We let

$$\mathsf{min}\ \left((>_{\mathsf{proc}})_\varphi \cdot \pi'\right) = (>_{\mathsf{proc}})_\varphi \cdot \{\neg\varphi \vee \neg\langle(>_{\mathsf{proc}})_\varphi \cdot \pi'\rangle\}? \cdot \mathsf{min}\ \pi' .$$

We prove that $(e, f) \in [\![\mathsf{min}\ \pi]\!]$ if and only if $f = \min[\![\pi]\!](e)$ similarly to the previous case. Assume that $(e, f) \in [\![\mathsf{min}\ \pi]\!]$. There exists g such that $(e, g) \in$

$[\![(>_{\mathsf{proc}})_\varphi]\!]$, $g \in [\![\neg\varphi \vee \neg\langle(>_{\mathsf{proc}})_\varphi \cdot \pi'\rangle]\!]$, and $(g, f) \in [\![\mathsf{min}\ \pi']\!]$. In particular, we have $f \in [\![\pi]\!](e)$. Let us show that f is minimal. Let g', f' such that $(e, g') \in [\![(>_{\mathsf{proc}})_\varphi]\!]$ and $(g', f') \in [\![\pi']\!]$. Suppose towards a contradiction that $g' <_{\mathsf{proc}} g$:



We then have $g' <_{\mathsf{proc}} g <_{\mathsf{proc}} e$, which implies $g \in [\![\varphi]\!]$ and $(g, g') \in [\![(>_{\mathsf{proc}})_\varphi]\!]$. Then $(g, f') \in [\![(>_{\mathsf{proc}})_\varphi \cdot \pi']\!]$, which contradicts the fact that $g \in [\![\neg\varphi \vee \neg\langle(>_{\mathsf{proc}})_\varphi \cdot \pi'\rangle]\!]$. Therefore, $g \leq_{\mathsf{proc}} g'$. By Lemma 5.22,

$$f = \min[\![\pi']\!](g) \leq_{\mathsf{proc}} \min[\![\pi']\!](g') \leq_{\mathsf{proc}} f' \,,$$

which proves the minimality of f.

Conversely, suppose that $f = \min[\![\pi]\!](e)$. Then $g = \min[\![(>_{\mathsf{proc}})_\varphi \cdot \{\langle\pi'\rangle\}?]\!](e)$ is well-defined, and $(e, g) \in [\![(>_{\mathsf{proc}})_\varphi]\!]$. Let us show that $g \in [\![\neg\varphi \vee \neg\langle(>_{\mathsf{proc}})_\varphi \cdot \pi'\rangle]\!]$. Suppose towards a contradiction that this is not the case, i.e., $g \in [\![\varphi]\!]$ and there exists g' such that $(g, g') \in [\![(>_{\mathsf{proc}})_\varphi \cdot \{\langle\pi'\rangle\}?]\!]$. Then $g' <_{\mathsf{proc}} g <_{\mathsf{proc}} e$, and for all $g' <_{\mathsf{proc}} h <_{\mathsf{proc}} e$, $h \in [\![\varphi]\!]$, hence $(e, g') \in [\![(>_{\mathsf{proc}})_\varphi \cdot \{\langle\pi'\rangle\}?]\!]$, which contradicts the minimality of g. Hence

$$(e, g) \in [\![(>_{\mathsf{proc}})_\varphi \cdot \{\neg\varphi \vee \neg\langle(>_{\mathsf{proc}})_\varphi \cdot \pi'\rangle\}?]\!] \,.$$

In addition, $[\![\pi']\!](g) \neq \emptyset$, hence there exists f' such that $(g, f') \in [\![\mathsf{min}\ \pi']\!]$. Then $(e, f') \in [\![\mathsf{min}\ \pi]\!]$, and by the previous implication, $f = f'$.

Regarding the size of $\mathsf{min}\ \pi$, notice that in all cases

$$\mathit{length}(\mathsf{min}\ (r \cdot \pi')) \leq 2 + \mathit{length}(\mathsf{min}\ (\pi')) \,,$$

which proves that $\mathit{length}(\mathsf{min}\ \pi) \leq 2 \cdot \mathit{length}(\pi)$. Besides, at every step, we have

$$\mathcal{S}(\mathsf{min}\ (r_i \cdots r_n)) \subseteq \mathcal{S}(\pi) \cup \{\langle r_i \cdots r_n\rangle \mid 1 \leq i \leq n\} \cup \mathcal{S}(\mathsf{min}\ (r_{i+1} \cdots r_n)) \cup \mathcal{F}_i \,,$$

for some set of formulas $\mathcal{F}_i$ such that $\|\mathcal{F}_i\| \leq C$ for some constant C. For instance, if $r_i = (>_{\mathsf{proc}})_\varphi$,

$$\mathcal{F}_i = \{\neg\varphi, \neg\langle r_i \cdots r_n\rangle, \neg\varphi \vee \neg\langle r_i \cdots r_n\rangle\} \quad \text{and} \quad \|\mathcal{F}_i\| = 3 \,.$$

This proves that

$$\mathcal{S}(\mathsf{min}\ \pi) \setminus \mathcal{S}(\pi) \subseteq \{\langle r_i \cdots r_n\rangle \mid 1 \leq i \leq n\} \cup \mathcal{F} \,,$$

for some $\mathcal{F}$ such that $\|\mathcal{F}\| = \mathcal{O}(n)$. Furthermore, we can expand every formula $\langle r_i \cdots r_n \rangle$ in $\mathsf{min}\ \pi$ into $\langle r_i \rangle \langle r_{i+1} \rangle \cdots \langle r_n \rangle$, and

$$\|\{\langle r_i \rangle \cdots \langle r_n \rangle \mid 1 \leq i \leq n\}\| = \mathcal{O}(n)\,.$$

We then have

$$\|\mathcal{S}(\mathsf{min}\ \pi) \setminus \mathcal{S}(\pi)\| = \mathcal{O}(n) = \mathcal{O}(\mathit{length}(\pi))\,. \qquad \square$$

5.2.3 Expressive completeness

Let us prove that $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ is as expressive as $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ or $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$.

One possible way to prove this would be to use $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ as an intermediate language. Indeed, by Theorems 3.14 and 3.30, it is as expressive as $\mathrm{FO}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ (i.e., as expressive as $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$). It would then not be very difficult to prove that any $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}\big[\Sigma^{\mathrm{int}}_{\mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})}\big]$ can be expressed as a positive boolean combination of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ formulas; we would only need to translate formulas

$$\sqsubset_\varphi\,, \quad \sqsupset_\varphi\,, \quad \mathsf{inf}\ (\lambda \cdot \lhd_{p,q})\,, \quad \mathsf{inf}\ (\lambda \cdot \lhd^{-1}_{p,q})\,, \quad \mathsf{sup}\ (\lambda \cdot \lhd_{p,q})\,, \quad \mathsf{sup}\ (\lambda \cdot \lhd^{-1}_{p,q})\,,$$

where $\lambda \in \{\sqsubseteq, \sqsubset, \sqsupseteq, \sqsupset\}$.

However, the rather technical proof that $\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ is expressively complete over arbitrary complete interval-preserving Σ-structures becomes much simpler in the special case of MSCs. To illustrate this, we choose a different approach than the one described above, and give a direct translation from $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ to $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ which does not rely on Theorem 3.30. We proceed by induction, exactly as in the translation from $\mathrm{FO}[\Sigma]$ to $\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ over arbitrary Σ-structures (Theorem 3.14). All we need is to adapt Lemmas 3.17 and 3.19, which were used in the proof of Theorem 3.14 to deal with negation and existential quantification, respectively. We do so in the next two lemmas.

Lemma 5.24. *For all $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, π^{c} is equivalent, over MSCs, to a finite union of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ formulas.*

Proof. If π is not $\top$-free, i.e., is of the form $\pi \equiv \pi_1 \cdot \top \cdot \pi_2$, then as observed before,

$$[\![\pi]\!] = [\![\langle \pi_1 \rangle]\!] \times [\![\langle \pi_2^{-1} \rangle]\!]\ ,$$

therefore

$$[\![\pi^{\mathsf{c}}]\!] = ([\![\langle \pi_1 \rangle]\!]^{\mathsf{c}} \times E) \cup \Big(E \times [\![\langle \pi_2^{-1} \rangle]\!]^{\mathsf{c}}\Big)\ ,$$

that is,

$$\pi^{\mathsf{c}} \equiv \{\neg\langle \pi_1 \rangle\}? \cdot \top + \top \cdot \{\neg\langle \pi_2^{-1} \rangle\}?\,.$$

Assume now that π is $\top$-free. Let us show that π^{c} is equivalent to

$$\pi' = \{\neg\langle\pi\rangle\}? \cdot \top + \top \cdot \{\neg\langle\pi^{-1}\rangle\}? + \sum_{(p,q)\notin \mathsf{Comp}(\pi)} \{p\}? \cdot \top \cdot \{q\}? + (\mathsf{min}\ \pi) \cdot >_{\mathsf{proc}} + (\mathsf{min}\ (\pi^{-1}) \cdot >_{\mathsf{proc}})^{-1} .$$

Clearly,

$$[\![\{\neg\langle\pi\rangle\}? \cdot \top]\!] \subseteq [\![\pi^{\mathsf{c}}]\!] \quad \text{and} \quad [\![\top \cdot \{\neg\langle\pi^{-1}\rangle\}?]\!] \subseteq [\![\pi^{\mathsf{c}}]\!] .$$

According to Lemma 5.16, we also have, for all $(p, q) \notin \mathsf{Comp}(\pi)$,

$$[\![\{p\}? \cdot \top \cdot \{q\}?]\!] \subseteq [\![\pi^{\mathsf{c}}]\!] .$$

And by definition of $\mathsf{min}\ \pi$,

$$[\![(\mathsf{min}\ \pi) \cdot >_{\mathsf{proc}}]\!] \subseteq [\![\pi^{\mathsf{c}}]\!] \quad \text{and} \quad \left[\!\!\left[(\mathsf{min}\ (\pi^{-1}) \cdot >_{\mathsf{proc}})^{-1}\right]\!\!\right] \subseteq \left([\![\pi^{-1}]\!]^{\mathsf{c}}\right)^{-1} = [\![\pi^{\mathsf{c}}]\!] .$$

Therefore, $[\![\pi']\!] \subseteq [\![\pi^{\mathsf{c}}]\!]$.

Conversely, let $(e, f) \notin [\![\pi']\!]$. We have $e \in [\![\langle\pi\rangle]\!]$ and $f \in [\![\langle\pi^{-1}\rangle]\!]$, hence

$$f' = \min[\![\pi]\!](e) \qquad \text{and} \qquad e' = \min[\![\pi^{-1}]\!](f)$$

are well-defined. Moreover, $(e, f) \notin [\![\{p\}? \cdot \top \cdot \{q\}?]\!]$ for any $(p, q) \notin \mathsf{Comp}(\pi)$ means that $(loc(e), loc(f)) \in \mathsf{Comp}(\pi)$. By Lemma 5.19, we then have $loc(f') = loc(f)$ and $loc(e') = loc(e)$. Since $(e, f) \notin [\![(\mathsf{min}\ \pi) \cdot >_{\mathsf{proc}}]\!]$, i.e., $f \not<_{\mathsf{proc}} f'$, we must have $f' \leq_{\mathsf{proc}} f$. Similarly, since $(f, e) \notin [\![\mathsf{min}\ (\pi^{-1}) \cdot >_{\mathsf{proc}}]\!]$, we have $e' \leq_{\mathsf{proc}} e$:

$$\begin{array}{rcl} \min[\![\pi^{-1}]\!](f) = e' & \leq_{\mathsf{proc}} & e \\ & \overset{\pi}{\searrow}\!\!\!\!\!\!\underset{\pi}{\swarrow} & \vdots \\ \min[\![\pi]\!](e) = f' & \leq_{\mathsf{proc}} & f \end{array}$$

Then

$$e' \leq_{\mathsf{proc}} e\,, \quad f' \leq_{\mathsf{proc}} f\,, \quad (e', f) \in [\![\pi]\!]\,, \quad \text{and} \quad (e, f') \in [\![\pi]\!]\,.$$

Applying Lemma 5.20, we obtain $(e, f) \in [\![\pi]\!]$. Therefore,

$$\pi' \equiv \pi^{\mathsf{c}}\,.$$

□

For a path formula π or an event formula φ in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, we denote by $\widetilde{\varphi}(x)$ or $\widetilde{\pi}(x, y)$ the translation of π into $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$. The next lemma is similar to Lemma 3.19.

Lemma 5.25. *Let $n \geq 1$. For all path formulas $\pi_1, \ldots, \pi_n$ and all event formulas φ in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, the $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ formula*

$$\Phi(x_1, \ldots, x_n) = \exists x. \left(\widetilde{\varphi}(x) \wedge \bigwedge_{1 \leq i \leq n} \widetilde{\pi_i}(x_i, x) \right) \qquad (x_i \neq x \text{ for all } i)$$

is equivalent, over MSCs, to a finite positive boolean combination of formulas of the form $\widetilde{\pi}(x_j, x_k)$, with $1 \leq j, k \leq n$ and $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$.

Proof. The proof is very similar to the proof of Lemma 3.19. Let $\psi = \varphi \wedge \bigwedge_{1 \leq i \leq n} \langle \pi_i^{-1} \rangle$, and

$$\Psi(x_1, \ldots, x_n) = \bigwedge_{1 \leq i,j \leq n} (\pi_i \cdot \widetilde{\{\psi\}?} \cdot \pi_j^{-1})(x_i, x_j)\,.$$

Let us show that $\Phi(x_1, \ldots, x_n) = \Psi(x_1, \ldots, x_n)$.

Let $M \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$, and $\nu : \{x_1, \ldots, x_n\} \to E$. For all $1 \leq i \leq n$, let $I_i = [\![\pi_i]\!](\nu(x_i)) \cap [\![\psi]\!]$. Notice that

$$\begin{aligned} M, \nu \models \Phi(x_1, \ldots, x_n) \quad &\iff \quad \bigcap_{1 \leq i \leq n} I_i \neq \emptyset \\ M, \nu \models \Psi(x_1, \ldots, x_n) \quad &\iff \quad \forall 1 \leq i, j \leq n,\ I_i \cap I_j \neq \emptyset\,. \end{aligned}$$

Moreover, each I_i is:

- an interval of $([\![\psi]\!] \cap E_p, \leq_{\mathsf{proc}})$ for some process p, if π_i is $\top$-free. This follows from Lemmas 5.19 and 5.21.
- equal to $[\![\psi]\!]$, if π_i is not $\top$-free and $[\![\pi_i]\!](\nu(x_i)) \neq \emptyset$. Indeed, in that case, $[\![\pi_i]\!](\nu(x_i)) = [\![\langle \pi_i^{-1} \rangle]\!] \supseteq [\![\psi]\!]$.
- empty, if $[\![\pi_i]\!](\nu(x_i)) = \emptyset$.

By Lemma 3.18, it is not difficult to see that the intersection of these sets is non-empty if and only if all pairwise intersections are non-empty. Thus,

$$\begin{aligned} M, \nu \models \Phi(x_1, \ldots, x_n) \quad &\iff \quad \bigcap_{1 \leq i \leq n} I_i \neq \emptyset \\ &\iff \quad \forall 1 \leq i, j \leq n,\ I_i \cap I_j \neq \emptyset \\ &\iff \quad M, \nu \models \Psi(x_1, \ldots, x_n)\,. \end{aligned}$$

□

Theorem 5.26. *Every formula $\Phi \in \mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ with at least one free variable is equivalent, over MSCs, to a positive boolean combination of formulas of the form $\widetilde{\pi}(x, y)$, where $x, y \in \mathrm{Free}(\Phi)$ and $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$.*

Proof. Apart from atomic formulas, for which we have already given a translation, the proof is exactly the same as the proof of Theorem 3.14, applying Lemma 5.24 instead of Lemma 3.17, and Lemma 5.25 instead of Lemma 3.19. □

As in Corollary 3.15, this also implies that:

Theorem 5.27. $\mathcal{L}(\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]) = \mathcal{L}(\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}])$.

5.3 Fragment without Loop formulas

In this section, we show that any $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ formula is equivalent, modulo projection, to one which has no $\mathsf{Loop}(\pi)$ subformula. This is defined more precisely in the next subsection. By removing Loop operators, we obtain a logic which is closer to LTL, and which will be simpler to translate into CFMs (see Chapter 6).

5.3.1 Main result and sketch of proof

Let $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$, and $\mathsf{Prop}' \subseteq \mathsf{Prop}$. The *projection of M to* Prop' is the MSC $\mathrm{Pr}_{\mathsf{Prop}'}(M) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop}')$ defined as

$$\mathrm{Pr}_{\mathsf{Prop}'}(M) = (E, \rightarrow, \lhd, loc, \lambda')\,, \qquad \text{where} \qquad \lambda'(e) = \lambda(e) \cap \mathsf{Prop}'\,.$$

The projection of a language of MSCs $\mathbb{L} \subseteq \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ to Prop' is then

$$\mathrm{Pr}_{\mathsf{Prop}'}(\mathbb{L}) = \{\mathrm{Pr}_{\mathsf{Prop}'}(M) \mid M \in \mathbb{L}\}\,.$$

Conversely, given $M \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ and $\mathsf{Prop} \subseteq \mathsf{Prop}'$, an *extension of M to* Prop' is an MSC $M' \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop}')$ such that $M = \mathrm{Pr}_{\mathsf{Prop}}(M')$.

The aim of this section is to prove the following:

Elimination of Loop

Theorem 5.28. *For all $\xi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, there exist $\mathsf{Prop}' \supseteq \mathsf{Prop}$ and $\xi' \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}']$ such that ξ' contains no occurrence of* Loop*, and*

$$\mathbb{L}(\xi) = \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi'))\,.$$

Moreover, $|\mathsf{Prop}'| = \mathcal{O}(|\mathsf{Prop}| + \|\xi\|)$ and $\|\xi'\| = \mathcal{O}(\|\xi\|)$.

The removal of loop formulas as in Theorem 5.28 is done in several steps, which are explained informally below. This summary should contain all the main ideas of the proof. Further technical details for each step can be found in the next sections.

Step 1. First, one can remove some simple occurrences of the Loop operator: for instance, formulas which can never be satisfied, such as $\mathsf{Loop}(<_{\mathsf{proc}} \cdot \lhd_{p,q} \cdot <_{\mathsf{proc}} \cdot \lhd_{q,r})$ for $p \neq r$, or formulas such as $\mathsf{Loop}(\pi_1 \cdot \top \cdot \pi_2)$ which can be replaced with $\langle \pi_1 \rangle \wedge \langle \pi_2^{-1} \rangle$.

After these simplifications, described in Section 5.3.2, we obtain a formula where each occurrence of Loop is applied to a $\top$-free path formula π such that we always have $[\![\pi]\!](e) \subseteq E_{loc(e)}$, which slightly simplifies the next steps.

Step 2. The second step consists in putting all remaining $\mathsf{Loop}(\pi)$ formulas in a normal form, so that the Loop operator only occurs in subformulas of the form

$$\mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))\,,$$

where P is a new atomic proposition, and π is a $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ path formula without any occurrence of Loop.

This transformation relies on the following ideas:

- For each subformula $\mathsf{Loop}(\pi)$, we introduce a new atomic proposition P, which is meant to be interpreted in the same manner as $\mathsf{Loop}(\pi)$. To ensure that this is the case, we add to the formula the conditions

  $$\mathsf{A}(P \implies \mathsf{Loop}(\pi)) \qquad \text{and} \qquad \mathsf{A}(\neg P \implies \neg\mathsf{Loop}(\pi))\,.$$

  Every other occurrence of $\mathsf{Loop}(\pi)$ is then replaced with the proposition P.

- Conditions of the form $\mathsf{A}(\neg P \implies \neg\mathsf{Loop}(\pi))$ can be removed by adding further atomic propositions and conditions of the form $\mathsf{A}(P \implies \mathsf{Loop}(\pi))$. Indeed, negative occurrences of $\mathsf{Loop}(\pi)$ can be replaced using positive occurrences of formulas $\mathsf{Loop}(\pi')$. More precisely, we prove in Lemma 5.31 that

  $$\neg\mathsf{Loop}(\pi) \equiv \neg\langle\pi\rangle \vee \neg\langle\pi^{-1}\rangle \vee \mathsf{Loop}\left((\mathsf{min}\ \pi) \cdot >_{\mathsf{proc}}\right) \vee \mathsf{Loop}\left(\mathsf{min}\ \left(\pi^{-1}\right) \cdot >_{\mathsf{proc}}\right)\,.$$

  Intuitively, $\neg\mathsf{Loop}(\pi) \equiv \mathsf{Loop}(\pi^{\mathsf{c}})$, and this can be seen as a special case of Lemma 5.24.

  We are left with a formula where all occurrences of Loop are in subformulas of the form

  $$\mathsf{A}(P \implies \mathsf{Loop}(\pi))\,,$$

  where P is a new atomic proposition, and π is a $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ path formula without any occurrence of Loop.

- Finally, we prove that we can restrict the form of formulas π appearing in such formulas, so that they are all of the form $\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}$. This is done in Lemma 5.32, which shows that

  $$\mathsf{Loop}(\pi) \equiv \langle\pi\rangle \wedge \langle\pi^{-1}\rangle \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}) \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi^{-1} \cdot \leq_{\mathsf{proc}})\,.$$

Intuitively, this is similar to Lemma 3.11 (notice that $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi^{-1} \cdot \leq_{\mathsf{proc}}) \equiv \mathsf{Loop}(\geq_{\mathsf{proc}} \cdot \pi \cdot \geq_{\mathsf{proc}})$). We obtain a simpler formula because of the stronger monotonicity properties of MSCs.

The normal form obtained after Steps 1 and 2 is given by Lemma 5.33. Note that in the actual proof of that lemma, the three ideas above are combined rather than applied one after the other.

Step 3. To remove the last occurrences of Loop, all that remains to be done is to rewrite formulas of the form

$$\xi = \mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))\,,$$

where π is a $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ path formula without any occurrence of Loop, into equivalent formulas which do not use the operator Loop but may contain additional atomic propositions.

First, notice that an event satisfies the formula $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})$ if and only if it belongs to an interval of the form $[\mathsf{min}\ \pi(e), e]$ (Lemma 5.35). Therefore, formula ξ simply says that the set of all such intervals covers the set of all P-labeled events.

Now, there may be a lot of overlapping between intervals of the form $[\mathsf{min}\ \pi(e), e]$. However, it is always possible to identify two sequences of such intervals such that: (i) the union of all intervals in the two sequences covers all intervals of the form $[\mathsf{min}\ \pi(e), e]$; and (ii) within each sequence, the intervals are disjoint and not adjacent. Technically, one may need to add unbounded intervals $[e, +\infty)$ obtained as the limit of increasing intervals $[e, f_1], [e, f_2], \ldots$ such that $e = \min[\![\pi]\!](f_i)$ for all $i \in \mathbb{N}$.

This results in Lemma 5.36, which says that a necessary condition for the satisfaction of ξ is the existence of two sequences of intervals of the form $[\mathsf{min}\ \pi(e), e]$ (or limit intervals) which cover all P-labeled events.

This is refined in Lemma 5.37, which says that to ensure that ξ is satisfied, we only need to find two sequences of intervals such that for each interval $[e, f]$, there exists some interval $[e', f']$ (which may or may not be the same) such that $e = \min[\![\pi]\!](f')$. An induction proves that the interval $[e', f']$ has to be to the right of $[e, f]$, and therefore both are contained in $[e, f']$, which is of the expected form $[\min[\![\pi]\!](f'), f']$.

This gives us a characterization of when formula ξ holds which is easier to express in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ without using the Loop operator. The equivalent (up to projection) Loop-free formula is constructed as follows:

- We use two additional atomic propositions to identify two sequences of disjoint (and non adjacent) intervals.

- The formula states that for all leftmost event e of an interval, there exists an event f such that $(e, f) \in [\![\mathsf{min}\ \pi]\!]$ and f is the rightmost event of some interval.
- A similar condition has to be checked for unbounded (limit) intervals.

Step 3 results in Lemma 5.38, which says that every formula

$$\mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))$$

is equivalent, up to projection, to a formula without Loop.

We can apply Lemma 5.38 to remove all occurrences of Loop from the formula in normal form obtained with Lemma 5.33, thus proving Theorem 5.28.

5.3.2 Simple cases

Some occurrences of Loop can easily be removed. This is for instance the case for Loop formulas which are not $\top$-free:

Lemma 5.29. *For all path formulas* $\pi_1, \pi_2 \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$*,*

$$\mathsf{Loop}(\pi_1 \cdot \top \cdot \pi_2) \equiv \langle \pi_1 \rangle \wedge \langle \pi_2^{-1} \rangle \,.$$

Proof. We have

$$\begin{aligned} e \in [\![\mathsf{Loop}(\pi_1 \cdot \top \cdot \pi_2)]\!] \quad &\text{iff} \quad \exists f, g.\, (e, f) \in [\![\pi_1]\!] \wedge (f, g) \in [\![\top]\!] \wedge (g, e) \in [\![\pi_2]\!] \\ &\text{iff} \quad \exists f, g.\, (e, f) \in [\![\pi_1]\!] \wedge (e, g) \in [\![\pi_2^{-1}]\!] \\ &\text{iff} \quad e \in [\![\langle \pi_1 \rangle \wedge \langle \pi_2^{-1} \rangle]\!] \,. \end{aligned}$$

□

Another case where Loop can be easily removed are formulas such as

$$\pi = <_{\mathsf{proc}} \cdot \lhd_{p,q} \cdot <_{\mathsf{proc}} \cdot \lhd_{q,r} \qquad (p \neq r) \,,$$

where it can be seen syntactically that $[\![\mathsf{Loop}(\pi)]\!]$ is always empty: in this example, for all $(e, f) \in [\![\pi]\!]$, e must be on process p and f on process r, hence $e \neq f$. This can be formalized in terms of $\mathsf{Comp}(\pi)$.

Lemma 5.30. *For all* $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$*, if* $\mathsf{Comp}(\pi) \cap \mathsf{Id}_{\mathsf{Procs}} = \emptyset$ *then, over MSCs,*

$$\mathsf{Loop}(\pi) \equiv \mathit{false} \,.$$

Proof. If $\mathsf{Comp}(\pi) \cap \mathsf{Id}_{\mathsf{Procs}} = \emptyset$, then by Lemma 5.16,

$$\{(e, e) \mid e \in [\![\mathsf{Loop}(\pi)]\!]\} \subseteq [\![\pi]\!]^M \cap \{E_p \times E_p \mid p \in \mathsf{Procs}\} = \emptyset \,,$$

hence $[\![\mathsf{Loop}(\pi)]\!] = \emptyset$. □

By Lemma 5.19, in all remaining cases, that is, if π is $\top$-free and $\mathsf{Comp}(\pi) \cap \mathsf{Id}_{\mathsf{Procs}} \neq \emptyset$, then $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$, and $[\![\pi]\!](e)$ is always included in the process of e, which slightly simplifies the situation.

5.3.3 A normal form for Loop formulas

This subsection establishes a normal form for $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop}]$ sentences (Step 2), in which Loop operators occur only in subformulas of the form

$$\mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))\,,$$

where P is a new atomic proposition, and π is a $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop}]$ path formula without any occurrence of Loop.

Note that the formulas $\leq_{\mathsf{proc}}$ and therefore $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})$ are *not* in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop}]$. We could still expand $\mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))$ into a $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop}]$ formula, using the fact that

$$\begin{aligned}\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}) \equiv \mathsf{Loop}(\pi) \vee \mathsf{Loop}(<_{\mathsf{proc}} \cdot \pi \cdot <_{\mathsf{proc}}) \vee \\ \mathsf{Loop}(\pi \cdot <_{\mathsf{proc}}) \vee \mathsf{Loop}(<_{\mathsf{proc}} \cdot \pi)\,.\end{aligned}$$

However, this normal form will be used only as intermediate step, so this expansion is not necessary.

As explained in the outline of the proof in Section 5.3.1, the normal form relies on the two results below, showed in the next two lemmas:

- negative occurrences of Loop formulas can be replaced with positive ones.
- formulas of the form $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})$ suffice to express all other $\mathsf{Loop}(\pi')$ formulas.

Lemma 5.31. *Let $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop}]$ be a $\top$-free formula such that $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$. Over MSCs,*

$$\neg\mathsf{Loop}(\pi) \equiv \neg\langle\pi\rangle \vee \neg\langle\pi^{-1}\rangle \vee \mathsf{Loop}\left((\mathsf{min}\ \pi) \cdot >_{\mathsf{proc}}\right) \vee \mathsf{Loop}\left(\mathsf{min}\ \left(\pi^{-1}\right) \cdot >_{\mathsf{proc}}\right)\,.$$

Proof. We have

$$\neg\mathsf{Loop}(\pi) \equiv \mathsf{Loop}(\pi^{\mathsf{c}})\,.$$

In addition, we showed in the proof of Lemma 5.24 that

$$\begin{aligned}\pi^{\mathsf{c}} \equiv \{\neg\langle\pi\rangle\}? \cdot \top + \top \cdot \{\neg\langle\pi^{-1}\rangle\}? + \sum_{(p,q)\notin\mathsf{Comp}(\pi)} \{p\}? \cdot \top \cdot \{q\}? + \\ (\mathsf{min}\ \pi) \cdot >_{\mathsf{proc}} + (\mathsf{min}\ (\pi^{-1}) \cdot >_{\mathsf{proc}})^{-1}\,.\end{aligned}$$

We also have $[\![\mathsf{Loop}(\{p\}? \cdot \top \cdot \{q\}?)]\!] \subseteq [\![\neg\langle\pi\rangle]\!]$ for all $(p,q) \notin \mathsf{Comp}(\pi)$. Indeed, if $e \in [\![\mathsf{Loop}(\{p\}? \cdot \top \cdot \{q\}?)]\!]$, then $p = q = loc(e)$. And since $(p,p) \notin \mathsf{Comp}(\pi)$ and $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$, Lemma 5.19 implies that there is no q' such that $(p,q') \in \mathsf{Comp}(\pi)$. Therefore, $e \in [\![\neg\langle\pi\rangle]\!]$.

We then have

$$\begin{aligned}\neg\mathsf{Loop}(\pi) &\equiv \mathsf{Loop}\,(\pi^{\mathsf{c}})\\ &\equiv \mathsf{Loop}\,(\{\neg\langle\pi\rangle\}?\cdot\top)\vee\mathsf{Loop}\,\big(\top\cdot\{\neg\langle\pi^{-1}\rangle\}?\big)\vee\\ &\qquad\bigvee_{(p,q)\notin\mathsf{Comp}(\pi)}\mathsf{Loop}\,(\{p\}?\cdot\top\cdot\{q\}?)\vee\\ &\qquad\mathsf{Loop}\,((\mathsf{min}\ \pi)\cdot{>_{\mathsf{proc}}})\vee\mathsf{Loop}\left(\left(\mathsf{min}\ (\pi^{-1})\cdot{>_{\mathsf{proc}}}\right)^{-1}\right)\\ &\equiv \neg\langle\pi\rangle\vee\neg\langle\pi^{-1}\rangle\vee\mathsf{Loop}\,((\mathsf{min}\ \pi)\cdot{>_{\mathsf{proc}}})\vee\mathsf{Loop}\left(\mathsf{min}\ \left(\pi^{-1}\right)\cdot{>_{\mathsf{proc}}}\right)\,.\Box\end{aligned}$$

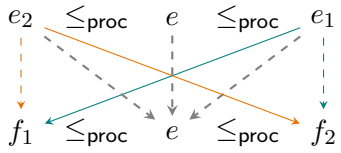
Lemma 5.32. *Let* $\pi\in\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop}]$. *Over MSCs,*

$$\mathsf{Loop}(\pi)\equiv\langle\pi\rangle\wedge\langle\pi^{-1}\rangle\wedge\mathsf{Loop}(\leq_{\mathsf{proc}}\cdot\pi\cdot\leq_{\mathsf{proc}})\wedge\mathsf{Loop}(\leq_{\mathsf{proc}}\cdot\pi^{-1}\cdot\leq_{\mathsf{proc}})\,.$$

Proof. The left-to-right implication is immediate. Conversely, let

$$e\in[\![\langle\pi\rangle\wedge\langle\pi^{-1}\rangle\wedge\mathsf{Loop}(\leq_{\mathsf{proc}}\cdot\pi\cdot\leq_{\mathsf{proc}})\wedge\mathsf{Loop}(\leq_{\mathsf{proc}}\cdot\pi^{-1}\cdot\leq_{\mathsf{proc}})]\!]\ .$$

Since $e\in[\![\mathsf{Loop}(\leq_{\mathsf{proc}}\cdot\pi\cdot\leq_{\mathsf{proc}})]\!]$, there exist $f_1\leq_{\mathsf{proc}}e\leq_{\mathsf{proc}}e_1$ such that $(e_1,f_1)\in[\![\pi]\!]$. And since $e\in[\![\mathsf{Loop}(\leq_{\mathsf{proc}}\cdot\pi^{-1}\cdot\leq_{\mathsf{proc}})]\!]$, there exist $e_2\leq_{\mathsf{proc}}e\leq_{\mathsf{proc}}f_2$ such that $(e_2,f_2)\in[\![\pi]\!]$:



We have

$$(e_1,f_1)\in[\![\pi]\!]\,,\quad(e_2,f_2)\in[\![\pi]\!]\,,\quad e_2\leq_{\mathsf{proc}}e_1\qquad\text{and}\qquad f_1\leq_{\mathsf{proc}}f_2\,,$$

thus, by Lemma 5.20,

$$(e_2,f_1)\in[\![\pi]\!]\quad\text{and}\quad(e_1,f_2)\in[\![\pi]\!]\,.$$

Then

$$(e_2,f_1)\in[\![\pi]\!]\,,\quad(e_2,f_2)\in[\![\pi]\!]\,,\quad e\in[\![\langle\pi^{-1}\rangle]\!]\quad\text{and}\quad f_1\leq e\leq f_2\,,$$

and by Lemma 5.21, we obtain $(e_2,e)\in[\![\pi]\!]$. Similarly, considering e_1,f_1,e,f_2, we obtain $(e_1,e)\in[\![\pi]\!]$. So we have

$$(e_2,e)\in[\![\pi]\!]\,,\quad(e_1,e)\in[\![\pi]\!]\,,\quad e\in[\![\langle\pi\rangle]\!]\quad\text{and}\quad e_2\leq e\leq e_1\,,$$

and applying again Lemma 5.21 yields $(e,e)\in[\![\pi]\!]$, that is, $e\in[\![\mathsf{Loop}(\pi)]\!]$. $\Box$

We are now ready to construct normal forms of formulas $\xi \in \mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$.

Lemma 5.33. *Let $\xi \in \mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$. There exist $n \geq 0$, $\mathsf{Prop}' = \mathsf{Prop} \uplus \{P_1, \ldots, P_n\}$, and formulas $\pi_1, \ldots, \pi_n$ and ξ' such that:*

- *ξ' is a sentence of $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}']$ with no occurrence of Loop.*
- *For all i, π_i is a $\top$-free path formula of $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}']$ with no occurrence of Loop, and such that $\mathsf{Comp}(\pi_i) \subseteq \mathsf{Id}_{\mathsf{Procs}}$.*
- *$\mathbb{L}(\xi) = \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi''))$, where*

$$\xi'' = \xi' \wedge \bigwedge_{1 \leq i \leq n} \mathsf{A}\,(P_i \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}}))\ .$$

- *$\|\xi''\| = \mathcal{O}(\|\xi\|)$. In particular, $n = \mathcal{O}(\|\xi\|)$.*

Proof. First, we apply Lemma 5.30 to replace every subformula $\mathsf{Loop}(\pi) \in \mathcal{S}(\xi)$ such that $\mathsf{Comp}(\pi) \cap \mathsf{Id}_{\mathsf{Procs}} = \emptyset$ with the equivalent formula *false*. We then apply Lemma 5.29 to replace every remaining subformula $\mathsf{Loop}(\pi)$ such that π is not $\top$-free, that is, $\pi = \pi_1 \cdot \top \cdot \pi_2$ for some π_1, π_2, with the equivalent formula $\langle \pi_1 \rangle \wedge \langle \pi_2^{-1} \rangle$. Each such substitution introduces a constant number of new subformulas, and does not increase the sum of the lengths of the path formulas occurring in ξ. So, in total, this increases $\|\xi\|$ by a linear factor at most. For any remaining subformula $\mathsf{Loop}(\pi)$, π is $\top$-free and $\mathsf{Comp}(\pi) \cap \mathsf{Id}_{\mathsf{Procs}} \neq \emptyset$, which implies, by Lemma 5.19, that $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$.

So in the remainder of the proof, we can assume that for all $\mathsf{Loop}(\pi) \in \mathcal{S}(\xi)$, π is $\top$-free and $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$. We proceed by induction on the number of formulas $\mathsf{Loop}(\pi) \in \mathcal{S}(\xi)$. If there are none, we take $n = 0$ and $\xi' = \xi$. Otherwise, let $\mathsf{Loop}(\pi) \in \mathcal{S}(\xi)$ such that π contains no occurrence of Loop.

Based on Lemma 5.31 and 5.32, we first define formulas $\pi_1, \ldots, \pi_6$ such that $\mathsf{Loop}(\pi)$ and $\neg\mathsf{Loop}(\pi)$ can be expressed as positive boolean combinations of formulas $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})$ and formulas without Loop. We let

$$\begin{aligned} \pi_1 &= \pi & \pi_2 &= \pi^{-1} \\ \pi_3 &= \mathsf{min}\ (\pi) \cdot >_{\mathsf{proc}} & \pi_4 &= \pi_3^{-1} \\ \pi_5 &= \mathsf{min}\ (\pi^{-1}) \cdot >_{\mathsf{proc}} & \pi_6 &= \pi_5^{-1}\ . \end{aligned}$$

Note that all the π_i are Loop free. By Lemma 5.32, we have

$$\mathsf{Loop}(\pi) \equiv \langle \pi_1 \rangle \wedge \langle \pi_2 \rangle \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_1 \cdot \leq_{\mathsf{proc}}) \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_2 \cdot \leq_{\mathsf{proc}}) \quad (5.1)$$

$$\mathsf{Loop}(\pi_3) \equiv \langle \pi_3 \rangle \wedge \langle \pi_4 \rangle \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_3 \cdot \leq_{\mathsf{proc}}) \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_4 \cdot \leq_{\mathsf{proc}}) \quad (5.2)$$

$$\mathsf{Loop}(\pi_5) \equiv \langle \pi_5 \rangle \wedge \langle \pi_6 \rangle \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_5 \cdot \leq_{\mathsf{proc}}) \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_6 \cdot \leq_{\mathsf{proc}}) \quad (5.3)$$

and by Lemma 5.31,

$$\neg\mathsf{Loop}(\pi) \equiv \neg\langle\pi_1\rangle \vee \neg\langle\pi_2\rangle \vee \mathsf{Loop}(\pi_3) \vee \mathsf{Loop}(\pi_5) \tag{5.4}$$

We now introduce, for each π_i, a new atomic proposition P_i which, intuitively, is meant to be interpreted in the same way as $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})$. In fact, though, we will ensure that $[\![P_i]\!] \subseteq [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})]\!]$, but not necessarily the converse inclusion. Accordingly, we let

$$\gamma = \bigwedge_{1\leq i\leq 6} \mathsf{A}\left(P_i \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})\right) .$$

Notice that γ respects the desired normal form.

Let $\tilde{\xi}$ be the formula obtained by replacing every occurrence of $\mathsf{Loop}(\pi)$ in ξ with the formula

$$\varphi = \langle\pi_1\rangle \wedge \langle\pi_2\rangle \wedge P_1 \wedge P_2 .$$

Note that in the special case where $[\![P_i]\!] = [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})]\!]$, φ is equivalent to $\mathsf{Loop}(\pi)$ and therefore, $\tilde{\xi}$ is equivalent to ξ. On the other hand, if we only assume that $[\![P_i]\!] \subseteq [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})]\!]$, we only have $[\![\varphi]\!] \subseteq [\![\mathsf{Loop}(\pi)]\!]$. We show below that we can recover the second inclusion by adding the condition

$$\delta = \mathsf{A}\left(\neg\varphi \implies \left(\begin{array}{c} \neg\langle\pi_1\rangle \vee \neg\langle\pi_2\rangle \\ \vee \left(\langle\pi_3\rangle \wedge \langle\pi_4\rangle \wedge P_3 \wedge P_4\right) \\ \vee \left(\langle\pi_5\rangle \wedge \langle\pi_6\rangle \wedge P_5 \wedge P_6\right) \end{array}\right)\right) .$$

Intuitively, the right-hand side of δ is obtained by unfolding (5.2) and (5.3) in (5.4), and replacing $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})$ with P_i.

Note that for all $i \in \{1, \ldots, 6\}$, by Lemma 5.13 and 5.23, $\|\mathcal{S}(\pi_i) \setminus \mathcal{S}(\xi)\| = \mathcal{O}(\mathit{length}(\pi))$. Therefore,

$$\|\tilde{\xi} \wedge \delta \wedge \gamma\| = \|\xi\| + \mathcal{O}(\mathit{length}(\pi)) \tag{5.5}$$

Claim 5.34. $\mathbb{L}(\xi) = \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\tilde{\xi} \wedge \delta \wedge \gamma))$.

Proof. Let $M = (E, \rightarrow, \lhd, \mathit{loc}, \lambda) \in \mathbb{L}(\xi)$. Let $M' = (E, \rightarrow, \lhd, \mathit{loc}, \lambda')$ be the extension of M to Prop' defined by $P_i \in \lambda'(e)$ if and only if $e \in [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})]\!]^M$. By definition, $M' \models \gamma$. Moreover, applying the definitions of $[\![P_1]\!]^{M'}$ and $[\![P_2]\!]^{M'}$ and (5.1),

$$\begin{aligned} [\![\varphi]\!]^{M'} &= [\![\langle\pi_1\rangle \wedge \langle\pi_2\rangle \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_1 \cdot \leq_{\mathsf{proc}}) \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_2 \cdot \leq_{\mathsf{proc}})]\!]^{M'} \\ &= [\![\mathsf{Loop}(\pi)]\!]^{M'} = [\![\mathsf{Loop}(\pi)]\!]^{M} . \end{aligned}$$

In particular, since $M \models \xi$, and since $\tilde{\xi}$ is obtained by replacing $\mathsf{Loop}(\pi)$ with φ in ξ, we have $M' \models \tilde{\xi}$. Using (5.2) and (5.3), we also have

$$\begin{aligned}\llbracket \mathsf{Loop}(\pi_3) \rrbracket^{M'} &= \llbracket \langle \pi_3 \rangle \wedge \langle \pi_4 \rangle \wedge P_3 \wedge P_4 \rrbracket^{M'} \\ \llbracket \mathsf{Loop}(\pi_5) \rrbracket^{M'} &= \llbracket \langle \pi_5 \rangle \wedge \langle \pi_6 \rangle \wedge P_5 \wedge P_6 \rrbracket^{M'} .\end{aligned}$$

Then, $\llbracket \varphi \rrbracket^{M'} = \llbracket \mathsf{Loop}(\pi) \rrbracket^{M'}$ also implies, with (5.4),

$$\begin{aligned}\llbracket \neg\varphi \rrbracket^{M'} &= \llbracket \neg \mathsf{Loop}(\pi) \rrbracket^{M'} \\ &= \llbracket \neg\langle \pi_1 \rangle \vee \neg\langle \pi_2 \rangle \vee \mathsf{Loop}(\pi_3) \vee \mathsf{Loop}(\pi_5) \rrbracket^{M'} \\ &= \llbracket \neg\langle \pi_1 \rangle \vee \neg\langle \pi_2 \rangle \vee (\langle \pi_3 \rangle \wedge \langle \pi_4 \rangle \wedge P_3 \wedge P_4) \vee (\langle \pi_5 \rangle \wedge \langle \pi_6 \rangle \wedge P_5 \wedge P_6) \rrbracket^{M'} .\end{aligned}$$

Therefore, $M' \models \delta$ and $M' \models \tilde{\xi} \wedge \delta \wedge \gamma$.

Conversely, let $M' = (E, \to, \lhd, loc, \lambda') \in \mathbb{L}(\tilde{\xi} \wedge \delta \wedge \gamma)$, and denote its projection on Prop by $M = (E, \to, \lhd, loc, \lambda)$. Since $M' \models \tilde{\xi}$, a sufficient condition for $M \models \xi$ is again

$$\llbracket \mathsf{Loop}(\pi) \rrbracket^{M} = \llbracket \varphi \rrbracket^{M'} .$$

Let us prove that this equality holds. Since $M' \models \gamma$, for all $i \in \{1, \ldots, 6\}$, we have $\llbracket P_i \rrbracket^{M'} \subseteq \llbracket \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}}) \rrbracket^{M'}$. In particular, using this and (5.1),

$$\begin{aligned}\llbracket \varphi \rrbracket^{M'} &= \llbracket \langle \pi_1 \rangle \wedge \langle \pi_2 \rangle \wedge P_1 \wedge P_2 \rrbracket^{M'} \\ &\subseteq \llbracket \langle \pi_1 \rangle \wedge \langle \pi_2 \rangle \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_1 \cdot \leq_{\mathsf{proc}}) \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_2 \cdot \leq_{\mathsf{proc}}) \rrbracket^{M'} \\ &\subseteq \llbracket \mathsf{Loop}(\pi) \rrbracket^{M'} = \llbracket \mathsf{Loop}(\pi) \rrbracket^{M} .\end{aligned}$$

Conversely,

$$\begin{aligned}\llbracket \neg\varphi \rrbracket^{M'} &\subseteq \llbracket \neg\langle \pi_1 \rangle \vee \neg\langle \pi_2 \rangle \vee (\langle \pi_3 \rangle \wedge \langle \pi_4 \rangle \wedge P_3 \wedge P_4) \vee (\langle \pi_5 \rangle \wedge \langle \pi_6 \rangle \wedge P_5 \wedge P_6) \rrbracket^{M'} \\ &\qquad\qquad \text{(since } M' \models \delta\text{)} \\ &\subseteq \left\llbracket \begin{array}{l} \neg\langle \pi_1 \rangle \vee \neg\langle \pi_2 \rangle \\ \vee (\langle \pi_3 \rangle \wedge \langle \pi_4 \rangle \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_3 \cdot \leq_{\mathsf{proc}}) \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_4 \cdot \leq_{\mathsf{proc}})) \\ \vee (\langle \pi_5 \rangle \wedge \langle \pi_6 \rangle \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_5 \cdot \leq_{\mathsf{proc}}) \wedge \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_6 \cdot \leq_{\mathsf{proc}})) \end{array} \right\rrbracket^{M'} \\ &\qquad\qquad \text{(since } M' \models \gamma\text{)} \\ &\subseteq \llbracket \neg \mathsf{Loop}(\pi) \rrbracket^{M'} \qquad \text{(using (5.2), (5.3) and (5.4))}\end{aligned}$$

Hence

$$\llbracket \varphi \rrbracket^{M'} = \llbracket \mathsf{Loop}(\pi) \rrbracket^{M} \quad \text{and} \quad M \models \xi . \qquad \square$$

Note that $\mathcal{S}(\tilde{\xi})$ contains one less Loop formula than $\mathcal{S}(\xi)$. To conclude the proof of the lemma, we apply the induction hypothesis to $\tilde{\xi}$. We obtain a formula ξ_1 over $\mathsf{Prop} \uplus \{P_1, \ldots, P_6\} \uplus \{P_7, \ldots, P_n\}$ such that

$$\mathbb{L}\left(\tilde{\xi}\right) = \mathrm{Pr}_{\mathsf{Prop} \cup \{P_1, \ldots, P_6\}}\left(\mathbb{L}\left(\xi_1 \wedge \gamma_1\right)\right) ,$$

where

$$\gamma_1 = \bigwedge_{7 \leq i \leq n} \mathsf{A}\left(P_i \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})\right) .$$

We can then define ξ' as

$$\xi' = \xi_1 \wedge \delta ,$$

and

$$\xi'' = \xi' \wedge (\gamma \wedge \gamma_1) = \xi_1 \wedge \delta \wedge \gamma \wedge \gamma_1 .$$

We then have

$$\begin{aligned} \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi'')) &= \mathrm{Pr}_{\mathsf{Prop}}(\mathrm{Pr}_{\mathsf{Prop} \cup \{P_1, \ldots, P_6\}}(\mathbb{L}(\xi_1 \wedge \gamma_1 \wedge \delta \wedge \gamma))) \\ &= \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\tilde{\xi} \wedge \delta \wedge \gamma)) , \end{aligned}$$

since the evaluation of δ and γ does not depend on the interpretation of atomic propositions outside of $\mathsf{Prop} \cup \{P_1, \ldots, P_6\}$. Then, by Claim 5.34,

$$\mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi'')) = \mathbb{L}(\xi) .$$

Regarding the size of ξ'', we obtain $\|\xi''\| = \mathcal{O}(\|\xi\|)$ from Equation (5.5). □

5.3.4 The case of Loop formulas in normal form

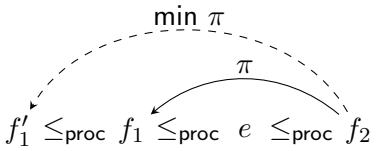
We now move on to Step 3: our goal is to prove that any formula of the form $\mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))$ can be translated into an equivalent formula (up to projection) without Loop.

A first observation is that the formula $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})$ can be replaced with $\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \mathsf{min}\ \pi \cdot \leq_{\mathsf{proc}})$:

Lemma 5.35. *Let $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ be a $\top$-free path formula. Over $\mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$,*

$$\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}) \equiv \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \mathsf{min}\ \pi \cdot \leq_{\mathsf{proc}}) .$$

Proof. The right-to-left implication is immediate, as $[\![\mathsf{min}\ \pi]\!] \subseteq [\![\pi]\!]$. Conversely, suppose that $e \in [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})]\!]$. There exist $f_1 \leq_{\mathsf{proc}} e \leq_{\mathsf{proc}} f_2$ such that $(f_2, f_1) \in [\![\pi]\!]$. Then $f_1' := \min[\![\pi]\!](f_2)$ is well-defined, and $f_1' \leq_{\mathsf{proc}} f_1$:



We then have $e \in [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \mathsf{min}\ \pi \cdot \leq_{\mathsf{proc}})]\!]$, since $f_1' \leq_{\mathsf{proc}} e \leq_{\mathsf{proc}} f_2$ and $(f_2, f_1') \in [\![\mathsf{min}\ \pi]\!]$. □

We say that a set I of events is a $\leq_{\mathsf{proc}}$-*interval* (or *interval* when there is no ambiguity) if it is an interval of $(E_p, \leq_{\mathsf{proc}})$ for some process p. Given an event e, we denote by $[e, +\infty)$ the interval $\{f \mid e \leq_{\mathsf{proc}} f\}$.

Lemma 5.35 implies that the formula $\mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))$ is satisfied if and only if every P-labeled event belongs to an interval of the form $[\min[\![\pi]\!](e), e]$. The next two lemmas give necessary and sufficient conditions for this to hold, which are easier to express in $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$ without using Loop.

First, from the set of all intervals $[\min[\![\pi]\!](e), e]$, we can extract two sequences of disjoint (and non-adjacent) intervals which cover all intervals $[\min[\![\pi]\!](e), e]$. In the case where infinitely many intervals $[\min[\![\pi]\!](e), e]$ overlap in a same event, then all but a finite number of them must share a same left endpoint f, and in that case, we replace those with the limit interval $[f, +\infty)$.

Having non-adjacent intervals will later allow us to identify them by labeling them with two extra atomic propositions (one for each sequence), and describe more easily the properties they should satisfy in $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$.

Lemma 5.36. *Let $\pi \in \mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$ be a $\top$-free path formula such that $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$, and $\xi = \mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))$. For all MSCs $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{L}(\xi)$, there exist two sets $\mathcal{I}$ and $\mathcal{J}$ of $\leq_{\mathsf{proc}}$-intervals such that:*

1. *The union of all intervals in $\mathcal{I}$ and $\mathcal{J}$ covers all P-labeled events, that is, $[\![P]\!]^M \subseteq \bigcup_{I \in \mathcal{I}} I \cup \bigcup_{J \in \mathcal{J}} J$.*
2. *For all $I, J \in \mathcal{I}$ or $I, J \in \mathcal{J}$ such that $I \neq J$, the intervals I and J are disjoint and non adjacent, that is, $I \cup J$ is not a $\leq_{\mathsf{proc}}$-interval.*
3. *Every bounded interval $I \in \mathcal{I}$ or $I \in \mathcal{J}$ is of the form $[\min[\![\pi]\!](e), e]$.*
4. *Every unbounded interval $I \in \mathcal{I}$ or $I \in \mathcal{J}$ is of the form $[e, +\infty)$ with $e = \min[\![\pi]\!](f)$ for infinitely many f.*

Proof. Let us prove that for all processes p, we can define two (finite or infinite) sequences $I_1, I_2, \ldots$ and $J_1, J_2, \ldots$ of intervals of $(E_p, \leq_{\mathsf{proc}})$ such that $\mathcal{I}_p = \{I_1, I_2, \ldots\}$ and $\mathcal{J}_p = \{J_1, J_2, \ldots\}$ satisfy conditions 2, 3 and 4, and such that

$$[\![P]\!]^M \cap E_p \subseteq \bigcup_{I \in \mathcal{I}_p} I \cup \bigcup_{J \in \mathcal{J}_p} J \tag{5.6}$$

We can then simply take $\mathcal{I} = \bigcup_{p \in \mathsf{Procs}} \mathcal{I}_p$ and $\mathcal{J} = \bigcup_{p \in \mathsf{Procs}} \mathcal{J}_p$ to conclude.

Fix $p \in \mathsf{Procs}$. Let $\mathcal{H}$ denote the set of intervals

$$\mathcal{H} = \{[\min[\![\pi]\!](e), e] \mid e \in E_p \wedge \min[\![\pi]\!](e) \leq e\}\,.$$

By Lemma 5.35, for every $f \in E_p$, we have $f \in [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot\, \pi \cdot \leq_{\mathsf{proc}})]\!]$ if and only if $f \in [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \min\, \pi \cdot \leq_{\mathsf{proc}})]\!]$, that is, if and only if there exist e, g such that $e \leq_{\mathsf{proc}} f \leq_{\mathsf{proc}} g$ and $e = \min[\![\pi]\!](g)$, i.e., $f \in \bigcup_{I \in \mathcal{H}} I$. Since $M \models \xi$, we have

$$[\![P]\!] \cap E_p \subseteq [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot\, \pi \cdot \leq_{\mathsf{proc}})]\!] \cap E_p = \bigcup_{I \in \mathcal{H}} I \tag{5.7}$$

So we only need to define two sequences of intervals satisfying conditions 2, 3 and 4 and covering exactly $\bigcup_{I \in \mathcal{H}} I$. To do so, we simply define alternately intervals $I_1, J_1, I_2, J_2, \ldots$ from left to right, choosing at each step the largest interval $I \in \mathcal{H}$ covering the next event $e \in \bigcup_{I \in \mathcal{H}} I$ which is not already covered. If there is no such "largest" interval, we add an unbounded interval instead.

Let us define this more formally. Given two intervals I and J, we write $I \preceq J$ if the right endpoint of J is larger or equal to the right endpoint of I, that is, $J = [e, +\infty)$, or $I = [e, f]$ and $J = [e', f']$ for some $f \leq_{\mathsf{proc}} f'$. Assume that intervals $I_1, J_1, \ldots, I_i, J_i$ are defined. If $\bigcup_{1 \leq j \leq i} I_j \cup \bigcup_{1 \leq j \leq i} J_j = \bigcup_{I \in \mathcal{H}} I$, we stop both sequences. Otherwise, let

$$e = \min\left(\bigcup_{I \in \mathcal{H}} I\right) \setminus \left(\bigcup_{1 \leq j \leq i} I_j \cup \bigcup_{1 \leq j \leq i} J_j\right)\,.$$

We define I_{i+1} as follows:

- If there are finitely many intervals $I \in \mathcal{H}$ such that $e \in I$, we define I_{i+1} as their maximum with respect to $\preceq$.
- If there are infinitely many intervals $I \in \mathcal{H}$ such that $e \in I$, then infinitely many of them share a same left endpoint, i.e., are of the form $[f, e_1], [f, e_2], \ldots$, with $f = \min[\![\pi]\!](e_i)$ for all $i \in \mathbb{N}$. We then let $I_{i+1} = [f, +\infty)$.

If $I_1, J_1, \ldots, I_i$ are defined, J_i is defined exactly as above.

Note that $I_i \subseteq \bigcup_{I \in \mathcal{H}}$ and $J_i \subseteq \bigcup_{I \in \mathcal{H}} I$ for all i, and that the set of events covered after each step is strictly increasing:

$$I_1 \subsetneq I_1 \cup J_1 \subsetneq I_1 \cup J_1 \cup I_2 \subsetneq \cdots$$

Moreover, by minimality of the event e considered at each step, these sets are all intervals of $\left(\bigcup_{I \in \mathcal{H}} I, \leq_{\mathsf{proc}}\right)$. Therefore, for the complete sequences,

$$\bigcup_i I_i \cup \bigcup_i J_i = \bigcup_{I \in \mathcal{H}} I\,.$$

By (5.7), this means that (5.6) is satisfied.

Condition 2 follows from the maximality of I_{i+1} and J_{i+1} at each step. Suppose towards a contradiction that there exist $i < j$ such that $I_i \cup I_j$ is an interval. Let

$$e = \min\left(\bigcup_{I\in\mathcal{H}} I\right) \setminus \left(\bigcup_{1\le k\le i} I_k \cup \bigcup_{1\le k<i} J_k\right)$$

be the event considered for the definition of J_i. We have

$$I_i <_{\mathsf{proc}} e \le_{\mathsf{proc}} \sup J_i <_{\mathsf{proc}} \sup I_j$$

(with the convention $f <_{\mathsf{proc}} +\infty$ for all events f), and since $I_i \cup I_j$ is an interval, $e \in I_j$. This contradicts the maximality of J_i.

Finally, conditions 3 and 4 follow from the definition of the intervals I_i and J_i. □

Condition 3 in Lemma 5.36 is difficult to express in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop}]$ without using the Loop operator. The next lemma provides an alternative condition, which will be easier to formalize in the logic, and which suffices to ensure that the formula $\mathsf{A}(P \implies \mathsf{Loop}(\le_{\mathsf{proc}} \cdot \pi \cdot \le_{\mathsf{proc}}))$ holds. Namely, instead of requiring that for all intervals $[e,f]$, we have $e = \min[\![\pi]\!](f)$, we only require that $e = \min[\![\pi]\!](f')$ for some (possibly other) interval $[e',f']$.

Lemma 5.37. *Let $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop}]$ be a $\top$-free path formula such that $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$ and $M = (E,\to,\lhd,loc,\lambda) \in \mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})$. Assume that there exist two sets $\mathcal{I}$ and $\mathcal{J}$ of $\le_{\mathsf{proc}}$-intervals such that:*

1. *$[\![P]\!]^M \subseteq \bigcup_{I\in\mathcal{I}} I \cup \bigcup_{J\in\mathcal{J}} J$.*
2. *For all $I,J \in \mathcal{I}$ or $I,J \in \mathcal{J}$ such that $I \neq J$, $I \cup J$ is not a $\le_{\mathsf{proc}}$-interval.*
3. *For every bounded interval $I = [e,f]$ in $\mathcal{I}$ (resp. in $\mathcal{J}$), there exists $J = [e',f']$ in $\mathcal{I}$ (resp. in $\mathcal{J}$) such that $e = \min[\![\pi]\!](f')$.*
4. *Every unbounded interval $I \in \mathcal{I}$ or $I \in \mathcal{J}$ is of the form $[e,+\infty)$ with $e = \min[\![\pi]\!](f)$ for infinitely many f.*

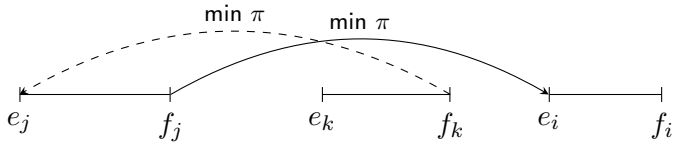
Then $M \models \mathsf{A}(P \implies \mathsf{Loop}(\le_{\mathsf{proc}} \cdot \pi \cdot \le_{\mathsf{proc}}))$.

Proof. We first prove that we can strengthen condition 3, namely, that for every bounded interval $I = [e,f] \in \mathcal{I}$, there exists $J = [e',f'] \in \mathcal{I}$ such that $e = \min[\![\pi]\!](f')$ and $I = J$ or $I <_{\mathsf{proc}} J$ (by symmetry, the same also holds for $\mathcal{J}$).

According to condition 2, for each process p, the set of intervals in $\mathcal{I}$ on process p forms a (finite or infinite) sequence $I_1 <_{\mathsf{proc}} I_2 <_{\mathsf{proc}} \cdots$. Since $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$, condition 3 implies that for all $I_i = [e_i,f_i]$, there exists $I_j = [e_j,f_j]$ such that

$e_i = \min[\![\pi]\!](f_j)$. Let us prove by induction on i that we can require that in addition, $j \geq i$ (i.e., $I = J$ or $I_i <_{\mathsf{proc}} I_j$).

For $i = 1$, this is immediate since $j \geq i$ for all j. Suppose that $I_i = [e_i, f_i]$, and that the result holds for all $i' < i$. There exists j such that $I_j = [e_j, f_j]$ and $e_i = \min[\![\pi]\!](f_j)$. Suppose towards a contradiction that $j < i$. Then, by induction, there exists $I_k = [e_k, f_k]$ such that $j \leq k$ and $e_j = \min[\![\pi]\!](f_k)$:



Since $I_j <_{\mathsf{proc}} I_i$ and ($j = k$ or $I_j <_{\mathsf{proc}} I_k$), we have

$$f_j \leq_{\mathsf{proc}} f_k \qquad \text{and} \qquad \min[\![\pi]\!](f_k) = e_j <_{\mathsf{proc}} e_i = \min[\![\pi]\!](f_j)\,,$$

a contradiction with Lemma 5.22.

We are now ready to prove that $M \models \mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))$. By condition 1, we only need to show that all intervals $I \in \mathcal{I} \cup \mathcal{J}$ are included in $[\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})]\!]$. We assume without loss of generality that $I \in \mathcal{I}$.

If I is bounded, i.e., $I = [e, f]$ for some e, f, then (as proved above) there exists an interval $J = [e', f'] \in \mathcal{I}$ such that $e = \min[\![\pi]\!](f')$ and $I = J$ or $I <_{\mathsf{proc}} J$. For all $g \in I$, we then have $e \leq_{\mathsf{proc}} g \leq_{\mathsf{proc}} f'$. Moreover, $(f', e) \in [\![\pi]\!]$, hence $g \in [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})]\!]$.

If I is unbounded, then by condition 4, for all $f \in I = [e, +\infty)$, there exists $g \geq_{\mathsf{proc}} f$ such that $e = \min[\![\pi]\!](g)$. Then $e \leq_{\mathsf{proc}} f \leq_{\mathsf{proc}} g$ and $(g, e) \in [\![\pi]\!]$, hence $f \in [\![\mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})]\!]$. □

The next lemma shows how to transform the $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ formula $\mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))$ into one without any occurrence of Loop. As explained above, the idea is to use two additional atomic propositions to identify sets of intervals satisfying the conditions of Lemma 5.37.

Lemma 5.38. *Let $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ be a $\top$-free path formula such that $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$ and π has no occurrence of Loop. Let $P \in \mathsf{Prop}$ and*

$$\xi = \mathsf{A}(P \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}))\,.$$

There exists a formula $\xi' \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop} \uplus \{I, J\}]$ with no occurrence of Loop such that

$$\mathbb{L}(\xi) = \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi'))\,, \quad \textit{and} \quad \|\mathcal{S}(\xi') \setminus \mathcal{S}(\pi)\| = \mathcal{O}(\textit{length}(\pi))\,.$$

Proof. We define event formulas I_{left}, I_{right}, J_{left} and J_{right} to describe leftmost and rightmost events of maximal intervals (with respect to $\leq_{\mathsf{proc}}$) of I-labeled or J-labeled events:

$$I_{left} := I \wedge \neg \langle \leftarrow \rangle I \qquad J_{left} := J \wedge \neg \langle \leftarrow \rangle J$$
$$I_{right} := I \wedge \neg \langle \rightarrow \rangle I \qquad J_{right} := J \wedge \neg \langle \rightarrow \rangle J\,.$$

To distinguish between leftmost events of bounded and unbounded intervals, we also introduce

$$I^{fin}_{left} = I_{left} \wedge (I_{right} \vee \langle <_{\mathsf{proc}} \rangle I_{right}) \qquad J^{fin}_{left} = J_{left} \wedge (J_{right} \vee \langle <_{\mathsf{proc}} \rangle J_{right})$$
$$I^{inf}_{left} = I_{left} \wedge \neg(I_{right} \vee \langle <_{\mathsf{proc}} \rangle I_{right}) \qquad J^{inf}_{left} = J_{left} \wedge \neg(J_{right} \vee \langle <_{\mathsf{proc}} \rangle J_{right})\,.$$

We also let

$$\langle (\mathsf{min}\ \pi)^{-1} \rangle_\infty := \langle (\mathsf{min}\ \pi)^{-1} \rangle \wedge \neg \langle <_{\mathsf{proc}} \cdot (\mathsf{min}\ \pi)^{-1} \rangle \wedge [<_{\mathsf{proc}}] \langle <_{\mathsf{proc}} \cdot \pi \rangle\,,$$

where, as defined before, $[\pi]\,\varphi = \neg \langle \pi \rangle \neg \varphi$.

Recall that

$$\begin{aligned} \|\mathcal{S}((\mathsf{min}\ \pi)^{-1}) \setminus \mathcal{S}(\pi)\| &= \|\mathcal{S}(\mathsf{min}\ \pi) \setminus \mathcal{S}(\pi)\| && \text{(Lemma 5.13)} \\ &= \mathcal{O}(\mathit{length}(\pi)) && \text{(Lemma 5.23)} \end{aligned}$$

and

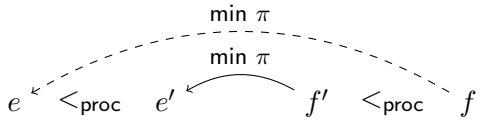
$$\mathit{length}((\mathsf{min}\ \pi)^{-1}) = \mathit{length}(\mathsf{min}\ \pi) = \mathcal{O}(\mathit{length}(\pi))\,.$$

Therefore,

$$\|\mathcal{S}(\langle (\mathsf{min}\ \pi)^{-1} \rangle_\infty) \setminus \mathcal{S}(\pi)\| = \mathcal{O}(\mathit{length}(\pi))\,.$$

Claim 5.39. We have $e \in [\![\langle (\mathsf{min}\ \pi)^{-1} \rangle_\infty]\!]$ if and only if $e = \min[\![\pi]\!](f)$ for infinitely many f.

Proof. Assume $e = \min [\![\pi]\!] (f)$ for infinitely many f. Note that since $\mathsf{Comp}(\pi) \subseteq \mathsf{Id}_{\mathsf{Procs}}$, all such f are located on process $\mathit{loc}(e)$. So we have $e \in [\![\langle (\mathsf{min}\ \pi)^{-1} \rangle]\!]$, and for all $f >_{\mathsf{proc}} e$, there exists $g >_{\mathsf{proc}} f$ such that $e = \min[\![\pi]\!](g)$. In particular, $f \in [\![\langle <_{\mathsf{proc}} \cdot \pi \rangle]\!]$. Hence $e \in [\![[<_{\mathsf{proc}}] \langle <_{\mathsf{proc}} \cdot \pi \rangle]\!]$. Finally, suppose towards a contradiction that $e \notin [\![\neg \langle <_{\mathsf{proc}} \cdot (\mathsf{min}\ \pi)^{-1} \rangle]\!]$, i.e., there exist $e' >_{\mathsf{proc}} e$ and f' such that $e' = \min[\![\pi]\!](f')$. Since $e = \min[\![\pi]\!](f)$ for infinitely many f, we can find one such f with $f >_{\mathsf{proc}} f'$:



We then have

$$f' <_{\mathsf{proc}} f \quad \text{and} \quad \min[\![\pi]\!](f) = e <_{\mathsf{proc}} e' = \min[\![\pi]\!](f')\,,$$

which contradicts Lemma 5.22.

Conversely, assume that $e \in [\![\langle(\mathsf{min}\ \pi)^{-1}\rangle_\infty]\!]$. Since $e \in [\![\langle(\mathsf{min}\ \pi)^{-1}\rangle]\!]$, there exists f such that $e = \min[\![\pi]\!](f)$. Since $e \in [\![[<_{\mathsf{proc}}]\,\langle <_{\mathsf{proc}} \cdot \pi\rangle]\!]$, there exists an infinite sequence $f <_{\mathsf{proc}} f_1 <_{\mathsf{proc}} f_2 <_{\mathsf{proc}} \cdots$ such that $f_i \in [\![\langle\pi\rangle]\!]$ for all $i \in \mathbb{N}$, that is, $e_i = \min[\![\pi]\!](f_i)$ is well-defined. Moreover, by Lemma 5.22, $f \leq_{\mathsf{proc}} f_i$ implies $e \leq_{\mathsf{proc}} e_i$. Since $e \in [\![\neg\langle <_{\mathsf{proc}} \cdot (\mathsf{min}\ \pi)^{-1}\rangle]\!]$, we cannot have $e <_{\mathsf{proc}} e_i$. Therefore, $e = e_i$ for all i. That is, $e = \min[\![\pi]\!](f_i)$ for infinitely many f_i. □

Let

$$\begin{aligned}
\xi' := \mathsf{A}(\quad P &\implies I \vee J &)\wedge\\
\mathsf{A}(\quad I^{fin}_{left} &\implies \langle(\mathsf{min}\ \pi)^{-1}\rangle\, I_{right} &)\wedge\\
\mathsf{A}(\quad I^{inf}_{left} &\implies \langle(\mathsf{min}\ \pi)^{-1}\rangle_\infty &)\wedge\\
\mathsf{A}(\quad J^{fin}_{left} &\implies \langle(\mathsf{min}\ \pi)^{-1}\rangle\, J_{right} &)\wedge\\
\mathsf{A}(\quad J^{inf}_{left} &\implies \langle(\mathsf{min}\ \pi)^{-1}\rangle_\infty &)\,.
\end{aligned}$$

As explained above, $\|\mathcal{S}(\langle(\mathsf{min}\ \pi)^{-1}\rangle_\infty) \setminus \mathcal{S}(\pi)\| = \mathcal{O}(\mathit{length}(\pi))$. Similarly, we also have $\|\mathcal{S}(\langle(\mathsf{min}\ \pi)^{-1}\rangle\, I_{right}) \setminus \mathcal{S}(\pi)\| = \mathcal{O}(\mathit{length}(\pi))$. Therefore,

$$\|\mathcal{S}(\xi') \setminus \mathcal{S}(\pi)\| = \mathcal{O}(\mathit{length}(\pi))\,.$$

The two claims below prove that

$$\mathbb{L}(\xi) = \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi'))\,.$$

Claim 5.40. For all $M \in \mathbb{L}(\xi)$, there exists $M' \in \mathbb{L}(\xi')$ such that $M = \mathrm{Pr}_{\mathsf{Prop}}(M')$.

Proof. Let $\mathcal{I}$ and $\mathcal{J}$ be the two sets of intervals given by Lemma 5.36. We define an extension M' of M to $\mathsf{Prop} \uplus \{I, J\}$ by setting

- $e \in [\![I]\!]^{M'}$ if and only if e is part of some interval in $\mathcal{I}$, and
- $e \in [\![J]\!]^{M'}$ if and only if e is part of some interval in $\mathcal{J}$.

By condition 1 of Lemma 5.36, we have

$$M' \models \mathsf{A}(P \implies I \vee J)\,.$$

Moreover, by condition 2 of Lemma 5.36, the intervals of $\mathcal{I}$ are exactly the maximal intervals of I-labeled events in M', and similarly for J. Then by condition 3,

$$\begin{aligned}
M' &\models \mathsf{A}(I^{fin}_{left} \implies \langle(\mathsf{min}\ \pi)^{-1}\rangle\, I_{right})\\
M' &\models \mathsf{A}(J^{fin}_{left} \implies \langle(\mathsf{min}\ \pi)^{-1}\rangle\, J_{right})\,,
\end{aligned}$$

and by condition 4,

$$M' \models \mathsf{A}(I^{inf}_{left} \implies \langle(\mathsf{min}\ \pi)^{-1}\rangle_\infty)$$
$$M' \models \mathsf{A}(J^{inf}_{left} \implies \langle(\mathsf{min}\ \pi)^{-1}\rangle_\infty)\,.$$

Hence, $M' \models \xi'$. □

Claim 5.41. For all $M' \in \mathbb{L}(\xi')$, $\mathrm{Pr}_{\mathsf{Prop}}(M') \in \mathbb{L}(\xi)$.

Proof. Let $M = \mathrm{Pr}_{\mathsf{Prop}}(M')$. Let $\mathcal{I}$ and $\mathcal{J}$ be the sets of maximal $\leq_{\mathsf{proc}}$-intervals of I- and J-labeled events. Let us show that these two sets satisfy the conditions of Lemma 5.37, which implies that $M \models \xi$.

Condition 1 follows from the fact that

$$M' \models \mathsf{A}(P \implies I \vee J)\,.$$

Condition 2 comes from the definition of the two sequences. Condition 3 is a consequence of

$$M' \models \mathsf{A}(I^{fin}_{left} \implies \langle(\mathsf{min}\ \pi)^{-1}\rangle\, I_{right}) \wedge \mathsf{A}(J^{fin}_{left} \implies \langle(\mathsf{min}\ \pi)^{-1}\rangle\, J_{right})\,,$$

and condition 4 of

$$M' \models \mathsf{A}(I^{inf}_{left} \implies \langle(\mathsf{min}\ \pi)^{-1}\rangle_\infty) \wedge \mathsf{A}(J^{inf}_{left} \implies \langle(\mathsf{min}\ \pi)^{-1}\rangle_\infty)\,.$$ □

□

5.3.5 Proof of Theorem 5.28

We can now combine results from the previous subsections to prove Theorem 5.28.

Theorem 5.28. *For all* $\xi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$*, there exist* $\mathsf{Prop}' \supseteq \mathsf{Prop}$ *and* $\xi' \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}']$ *such that* ξ' *contains no occurrence of* Loop*, and*

$$\mathbb{L}(\xi) = \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi'))\,.$$

Moreover, $|\mathsf{Prop}'| = \mathcal{O}(|\mathsf{Prop}| + \|\xi\|)$ *and* $\|\xi'\| = \mathcal{O}(\|\xi\|)$.

Proof of Theorem 5.28. Let $\xi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$.

We first apply Lemma 5.33. We obtain a formula

$$\xi'' = \xi' \wedge \bigwedge_{1 \leq i \leq n} \mathsf{A}\,(P_i \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot\, \pi_i \cdot \leq_{\mathsf{proc}}))$$

in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}'']$, where $\mathsf{Prop}'' = \mathsf{Prop} \uplus \{P_i \mid 1 \leq i \leq n\}$, such that

$$\|\xi''\| = \mathcal{O}(\|\xi\|) \quad \text{and} \quad \mathbb{L}(\xi) = \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi'')) \,.$$

Moreover, ξ' and each of the π_i contains no occurrence of Loop, and for all i, π_i is $\top$-free and $\mathsf{Comp}(\pi_i) \subseteq \mathsf{Id}_{\mathsf{Procs}}$.

Next, we apply Lemma 5.38 to each subformula

$$\xi_i = \mathsf{A}(P_i \implies \mathsf{Loop}(\leq_{\mathsf{proc}} \cdot \pi_i \cdot \leq_{\mathsf{proc}})) \,.$$

Let

$$\mathsf{Prop}_i = \mathsf{Prop}'' \uplus \{I_i, J_i\} = \mathsf{Prop} \uplus \{P_j \mid 1 \leq j \leq n\} \uplus \{I_i, J_i\} \,.$$

By Lemma 5.38, for all i, there exists a formula $\xi'_i \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}_i]$ with no occurrence of Loop such that $\|\mathcal{S}(\xi'_i) \setminus \mathcal{S}(\pi_i)\| = \mathcal{O}(\mathit{length}(\pi_i))$ and

$$\mathbb{L}(\xi_i) = \mathrm{Pr}_{\mathsf{Prop}''}(\mathbb{L}(\xi'_i)) \,.$$

Finally, let $\mathsf{Prop}' = \mathsf{Prop} \uplus \{P_i, I_i, J_i \mid 1 \leq i \leq n\}$, and $\hat{\xi} \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ be the formula

$$\hat{\xi} = \xi' \wedge \xi'_1 \wedge \cdots \wedge \xi'_n \,.$$

When evaluating $\hat{\xi}$ in a given MSC, the interpretation of each pair of atomic propositions (I_i, J_i) affects only the evaluation of ξ'_i, and not other conjuncts. Therefore,

$$\mathrm{Pr}_{\mathsf{Prop}''}\left(\mathbb{L}\left(\hat{\xi}\right)\right) = \mathbb{L}\left(\xi' \wedge \xi_1 \wedge \cdots \wedge \xi_n\right) = \mathbb{L}\left(\xi''\right) \,,$$

and

$$\mathrm{Pr}_{\mathsf{Prop}}\left(\mathbb{L}\left(\hat{\xi}\right)\right) = \mathrm{Pr}_{\mathsf{Prop}}\left(\mathbb{L}\left(\xi''\right)\right) = \mathbb{L}(\xi) \,.$$

Regarding the size of $\hat{\xi}$, we have

$$\|\mathcal{S}(\hat{\xi}) \setminus \mathcal{S}(\xi'')\| = \mathcal{O}(\mathit{length}(\pi_1) + \cdots + \mathit{length}(\pi_n)) = \mathcal{O}(\|\xi''\|) \,,$$

hence

$$\|\hat{\xi}\| = \mathcal{O}(\|\xi''\|) = \mathcal{O}(\|\xi\|) \,. \qquad \square$$

5.4 Temporal logics

Event formulas of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ constructed without the Loop operator constitute a simple temporal logic extending LTL from words to MSCs: any such formula is equivalent to one over the syntax

$$\varphi ::= \mathit{true} \mid p \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle \lhd_{p,q} \rangle\, \varphi \mid \langle \lhd^{-1}_{p,q} \rangle\, \varphi \mid \langle (<_{\mathsf{proc}})_{\varphi} \rangle\, \varphi \mid \langle (>_{\mathsf{proc}})_{\varphi} \rangle\, \varphi \mid \langle \top \rangle\, \varphi \,,$$

where $P \in \mathsf{Prop}$ and $p \in \mathsf{Procs}$. In the case of words, that is, for $|\mathsf{Procs}| = 1$, there are no formulas $\langle \lhd_{p,q} \rangle \varphi$ or $\langle \lhd_{p,q}^{-1} \rangle \varphi$, and the formula $\langle \top \rangle \varphi$ becomes equivalent to $\varphi \vee \langle <_{\mathsf{proc}} \rangle \varphi \vee \langle >_{\mathsf{proc}} \rangle \varphi$. Moreover, $\langle (<_{\mathsf{proc}})_\varphi \rangle \psi$ is the usual "φ strict until ψ" from LTL, while $\langle (>_{\mathsf{proc}})_\varphi \rangle \psi$ corresponds to the strict since modality. Therefore, for $|\mathsf{Procs}| = 1$, this logic corresponds exactly to LTL over words.

While this logic is quite expressive, as shown in the previous section, it has one major drawback: it does not allow to easily express properties of the happened-before relation $\leq$. Consider for instance the mutual exclusion property, which could be expressed in FO$[\mathsf{Procs}, \mathsf{Prop}, \leq]$ with the formula

$$\neg(\exists x. \exists y.\, x \parallel y \wedge CS(x) \wedge CS(y)), \quad \text{where} \quad x \parallel y := \neg(x \leq y) \wedge \neg(y \leq x),$$

assuming the predicate CS denotes the fact that the current process is in the critical section. It is not clear whether it can be defined in the temporal logic described above ($\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs}, \mathsf{Prop}]$ without Loop). Moreover, while Theorems 5.26 and 5.28 tell us that it becomes expressible when we add either the Loop operator, or existential unary predicate quantification, the resulting formula is far from simple.

We define below another temporal logic for MSCs, based on the happened-before relation $\leq$ rather than on the process order $\leq_{\mathsf{proc}}$. It is adapted from classical logics for partial orders which have been studied for Mazurkiewicz traces [85, 23, 33]. It uses universal versions of the (strict) until and (strict) since modalities, as well as a Co modality which jumps to a concurrent event: we say that two events e and f are *concurrent*, written $e \parallel f$, if $e \not\leq f$ and $f \not\leq e$.

Temporal logic for MSCs

Definition 5.42. Let TL$[\mathsf{Procs}, \mathsf{Prop}, \langle \lhd \rangle, \langle \lhd^{-1} \rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ be the set of formulas

$$\varphi ::= \mathit{true} \mid p \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle \lhd \rangle \varphi \mid \langle \lhd^{-1} \rangle \varphi \mid \mathsf{Co}\,\varphi \mid \varphi\,\mathsf{SU}\,\varphi \mid \varphi\,\mathsf{SS}\,\varphi$$

where $p \in \mathsf{Procs}$ and $P \in \mathsf{Prop}$.

A formula $\varphi \in$ TL$[\mathsf{Procs}, \mathsf{Prop}, \langle \lhd \rangle, \langle \lhd^{-1} \rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ is interpreted over events of MSCs. Given $M = (E, \rightarrow, \lhd, \mathit{loc}, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ and $e \in E$,

$M, e \models \mathit{true}$		
$M, e \models p$	if	$\mathit{loc}(e) = p$
$M, e \models P$	if	$P \in \lambda(e)$
$M, e \models \varphi \vee \psi$	if	$M, e \models \varphi$ or $M, e \models \psi$
$M, e \models \neg\varphi$	if	$M, e \not\models \varphi$
$M, e \models \langle \lhd \rangle \varphi$	if	there exists $f \in E$ such that $(e, f) \in \lhd$ and $M, f \models \varphi$
$M, e \models \langle \lhd^{-1} \rangle \varphi$	if	there exists $f \in E$ such that $(f, e) \in \lhd$ and $M, f \models \varphi$
$M, e \models \mathsf{Co}\,\varphi$	if	there exists $f \in E$ such that $e \parallel f$ and $M, f \models \varphi$
$M, e \models \varphi\,\mathsf{SU}\,\psi$	if	there exists $f \in E$ such that $e < f$ and $M, f \models \psi$ and, for all $e < g < f$, $M, g \models \varphi$
$M, e \models \varphi\,\mathsf{SS}\,\psi$	if	there exists $f \in E$ such that $f < e$ and $M, f \models \psi$ and, for all $f < g < e$, $M, g \models \varphi$.

With the modalities SU and SS, we can define various derived modalities, such as the non-strict until U and since S, or process-based modalities X_p, Y_p and U_p, with the following meaning: X_p moves to the first event on process p that is in the strict future of the current event, while Y_p moves to the last event on process p that is in the strict past of the current event; finally, U_p is the usual LTL (non-strict) until for a single process p, evaluated at the current event if it is on process p, or the first event of its future that is on process p otherwise:

$$\begin{aligned} \varphi \,\mathsf{U}\, \psi &:= \psi \vee (\varphi \wedge (\varphi \,\mathsf{SU}\, \psi)) & \varphi \,\mathsf{S}\, \psi &:= \psi \vee (\varphi \wedge (\varphi \,\mathsf{SS}\, \psi)) \\ \mathsf{X}_p\, \varphi &:= \neg p \,\mathsf{SU}\, (p \wedge \varphi) & \mathsf{Y}_p\, \varphi &:= \neg p \,\mathsf{SS}\, (p \wedge \varphi) \\ \varphi_1 \,\mathsf{U}_p\, \varphi_2 &:= (\neg p \vee \varphi_1) \,\mathsf{U}\, (p \wedge \varphi_2)\,. \end{aligned}$$

Temporal logics over traces based on the modalities SU, U, X_p and U_p and expressively complete for first-order logic were defined in [23]. The modality Co has been used e.g. in [2, 95]. The logic introduced by Thiagarajan in [85] uses an until modality similar to U_p, except that if the current event is not on process p, the evaluation starts at the latest event on process p in the past of the current event (or the first event of process p if none exists), instead of the first in its future. The second temporal modality of this logic is a unary modality $\mathcal{O}_p$ interpreted as follows: $\mathcal{O}_p\, \varphi$ holds at e if the first event on process p that is not in the past of e satisfies φ. Both can similarly be expressed in $\mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle \lhd \rangle, \langle \lhd^{-1} \rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$. An overview of temporal logics over Mazurkiewicz traces based on the modalities presented here and others is given in [33].

Example 5.43. The mutual exclusion property as defined above holds when all events satisfy the formula

$$CS \implies \neg\, \mathsf{Co}\, CS\,.$$

As another example, the formula

$$(p \wedge \langle \lhd \rangle\, q) \implies \mathsf{X}_q \left(\langle \lhd^{-1} \rangle\, p \right)$$

says that if the current event e is a write to channel (p, q), then the first event on process q in the future of e is the matching receive event. In other terms, messages sent through channel (p, q) travel faster than information sent indirectly through other processes.

Complexity. All $\mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle \lhd \rangle, \langle \lhd^{-1} \rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ formulas can be translated in a straightforward way into $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$, and then, by Theorem 5.26, into event formulas of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$. However, the translation from first-order logic to $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ is non-elementary (each negation causing an exponential blow-up), so this approach is not efficient. Instead, we give a direct translation from $\mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle \lhd \rangle, \langle \lhd^{-1} \rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ to $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$. This will be

particularly useful when combined with the results from the next chapter, which gives a translation from $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs},\mathsf{Prop}]$ to CFMs. While we need to be careful to obtain a better complexity, this direct translation still uses the same ideas as in the translation from first-order logic to star-free PDL.

As expected, we say that a formula $\varphi \in \mathrm{TL}[\mathsf{Procs},\mathsf{Prop},\langle\lhd\rangle,\langle\lhd^{-1}\rangle,\mathsf{Co},\mathsf{SU},\mathsf{SS}]$ is *equivalent* to an event formula $\psi \in \mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs},\mathsf{Prop}]$, or, more generally, $\psi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Procs},\mathsf{Prop},\leq_{\mathsf{proc}},\lhd]$, if for all MSCs $M \in \mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})$ and events e in M, we have $M,e \models \varphi$ if and only if $M,e \models \psi$. This is written $\varphi \equiv \psi$.

Let us explain how the modalities $\mathsf{Co}\,\varphi$, $\varphi\mathsf{SU}\psi$ and $\varphi\mathsf{SS}\psi$ based on the happened-before relation $\leq$ can be defined in $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs},\mathsf{Prop}]$. First, $\leq$ can be expressed in $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs},\mathsf{Prop}]$ in a similar way as in $\mathrm{FO}[\mathsf{Procs},\mathsf{Prop},\lhd,\leq_{\mathsf{proc}}]$.

Definition 5.44. Let Π be the set of path formulas $\pi \in \mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs},\mathsf{Prop}]$ of the form

$$\pi = \pi_1 \cdot \lhd_{p_1,p_2} \cdot <_{\mathsf{proc}} \cdot \lhd_{p_2,p_3} \cdots <_{\mathsf{proc}} \cdot \lhd_{p_{n-1},p_n} \cdot \pi_2\,,$$

where $\pi_1,\pi_2 \in \{\{\mathit{true}\}?,<_{\mathsf{proc}}\}$, $1 \leq n \leq |\mathsf{Procs}|$, $p_1,\ldots,p_n \in \mathsf{Procs}$ and $p_i \neq p_j$ for $i \neq j$.

We also denote by Π^+ the set of formulas in Π which additionally satisfy $n > 1$ or $\pi_1 = <_{\mathsf{proc}}$ or $\pi_2 = <_{\mathsf{proc}}$ (i.e., such that $\pi \not\equiv \{\mathit{true}\}?$).

Remark 5.45. All path formulas $\pi \in \Pi$ are $\top$-free (cf. Definition 5.17).

Formulas in Π describe all possible direct paths (not going twice through a same process) from one process to another. Formulas in Π^+ additionally require the path to be nonempty. This leads to the following lemma:

Lemma 5.46. *Let $M = (E,\to,\lhd,\mathit{loc},\lambda) \in \mathbb{MSC}(\mathsf{Procs},\mathsf{Prop})$, and $e,f \in E$. The following are equivalent:*

1. *$e \leq f$ (resp. $e < f$).*
2. *There exists $\pi \in \Pi$ (resp. $\pi \in \Pi^+$) such that $(e,f) \in [\![\pi]\!]$.*
3. *There exists $\pi \in \Pi$ (resp. $\pi \in \Pi^+$) such that $(e,f) \in [\![\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}]\!]$.*
4. *There exists $\pi \in \Pi$ (resp. $\pi \in \Pi^+$) such that $(e,f) \in [\![\leq_{\mathsf{proc}} \cdot \pi]\!]$.*
5. *There exists $\pi \in \Pi$ (resp. $\pi \in \Pi^+$) such that $(e,f) \in [\![\pi \cdot \leq_{\mathsf{proc}}]\!]$.*

As a first step towards expressing $\mathsf{Co}\,\varphi$, $\varphi\mathsf{SU}\psi$, or $\varphi\mathsf{SS}\psi$ in $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}[\mathsf{Procs},\mathsf{Prop}]$, we can start with formulas in $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Prop},\mathsf{Prop},\leq_{\mathsf{proc}},\lhd]$. We could define $\mathsf{Co}\,\varphi$ as

$$\begin{aligned}\left\langle (\leq + \leq^{-1})^{\mathsf{c}} \right\rangle \varphi &\equiv \left\langle \left(\textstyle\sum_{\pi\in\Pi} \leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}} + \sum_{\pi\in\Pi} \geq_{\mathsf{proc}} \cdot \pi^{-1} \cdot \geq_{\mathsf{proc}}\right)^{\mathsf{c}} \right\rangle \varphi \\ &\equiv \left\langle \textstyle\bigcap_{\pi\in\Pi} (\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})^{\mathsf{c}} \cap (\geq_{\mathsf{proc}} \cdot \pi^{-1} \cdot \geq_{\mathsf{proc}})^{\mathsf{c}} \right\rangle \varphi\,.\end{aligned}$$

Similarly, we could define $\varphi \,\mathsf{SU}\, \psi$ as

$$\begin{aligned}
&\langle < \cap (< \cdot \{\neg\varphi\}? \cdot <)^{\mathsf{c}} \rangle \, \psi \\
&\quad \equiv \left\langle \begin{array}{c} \left(\sum_{\pi\in\Pi^+} \leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}\right) \\ \cap \left(\left(\sum_{\pi\in\Pi^+} \leq_{\mathsf{proc}} \cdot \pi\right) \cdot \{\neg\varphi\}? \cdot \left(\sum_{\pi\in\Pi^+} \pi \cdot \leq_{\mathsf{proc}}\right)\right)^{\mathsf{c}} \end{array} \right\rangle \psi \\
&\quad \equiv \bigvee_{\pi\in\Pi^+} \left\langle \begin{array}{c} \left(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}\right) \\ \cap \bigcap_{\pi',\pi''\in\Pi^+} \left(\leq_{\mathsf{proc}} \cdot \pi' \cdot \{\neg\varphi\}? \cdot \pi'' \cdot \leq_{\mathsf{proc}}\right)^{\mathsf{c}} \end{array} \right\rangle \psi \,,
\end{aligned}$$

and $\varphi \,\mathsf{SS}\, \psi$ as

$$\begin{aligned}
&\langle > \cap (> \cdot \{\neg\varphi\}? \cdot >)^{\mathsf{c}} \rangle \, \psi \\
&\quad \equiv \left\langle \begin{array}{c} \left(\sum_{\pi\in\Pi^+} \geq_{\mathsf{proc}} \cdot \pi^{-1} \cdot \geq_{\mathsf{proc}}\right) \\ \cap \left(\left(\sum_{\pi\in\Pi^+} \geq_{\mathsf{proc}} \cdot \pi^{-1}\right) \cdot \{\neg\varphi\}? \cdot \left(\sum_{\pi\in\Pi^+} \pi^{-1} \cdot \geq_{\mathsf{proc}}\right)\right)^{\mathsf{c}} \end{array} \right\rangle \psi \\
&\quad \equiv \bigvee_{\pi\in\Pi^+} \left\langle \begin{array}{c} \left(\geq_{\mathsf{proc}} \cdot \pi^{-1} \cdot \geq_{\mathsf{proc}}\right) \\ \cap \bigcap_{\pi',\pi''\in\Pi^+} \left(\geq_{\mathsf{proc}} \cdot (\pi')^{-1} \cdot \{\neg\varphi\}? \cdot (\pi'')^{-1} \cdot \geq_{\mathsf{proc}}\right)^{\mathsf{c}} \end{array} \right\rangle \psi \,,
\end{aligned}$$

The motivation for using extra occurrences of $\leq_{\mathsf{proc}}$ in these formulas is to obtain simpler complements at no cost, and be able to apply the next lemma.

Lemma 5.47. *Let φ be an event formula of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, and $\pi_1, \ldots, \pi_n$ path formulas of the form*

$$\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}, \quad \geq_{\mathsf{proc}} \cdot \pi \cdot \geq_{\mathsf{proc}}, \quad (\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})^{\mathsf{c}} \quad \text{or} \quad (\geq_{\mathsf{proc}} \cdot \pi \cdot \geq_{\mathsf{proc}})^{\mathsf{c}},$$

where $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$. Then

$$\langle \pi_1 \cap \cdots \cap \pi_n \rangle \, \varphi \equiv \bigvee_{p\in\mathsf{Procs}} \bigwedge_{1\leq i,j\leq n} \mathsf{Loop}\left(\pi_i \cdot \{\varphi \wedge p\}? \cdot \pi_j^{-1}\right).$$

Proof. Note that

$$\langle \pi_1 \cap \cdots \cap \pi_n \rangle \, \varphi \equiv \bigvee_{p\in\mathsf{Procs}} \langle \pi_1 \cap \cdots \cap \pi_n \rangle \, (\varphi \wedge p)$$

and

$$\mathsf{Loop}\left(\pi_i \cdot \{\varphi \wedge p\}? \cdot \pi_j^{-1}\right) \equiv \langle \pi_i \cap \pi_j \rangle \, (\varphi \wedge p).$$

Thus, the left-to-right implication holds.

The proof for the right-to-left implication is extremely similar to the proof of Lemma 3.19, and relies on the following observation:

Claim 5.48. For all $1 \leq i \leq n$, for all $M \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ and all events e in M, for all processes $p \in \mathsf{Procs}$, the set $E_p \cap [\![\pi_i]\!](e)$ is an interval of $(E_p, \leq_{\mathsf{proc}})$.

Proof. For all path formulas π, $E_p \cap [\![\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}]\!](e)$ is an upward-closed interval of $(E_p, \leq_{\mathsf{proc}})$, and therefore

$$E_p \cap [\![(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})^{\mathsf{c}}]\!](e) = E_p \setminus (E_p \cap [\![\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}]\!](e))$$

is a (downward-closed) interval of $(E_p, \leq_{\mathsf{proc}})$. The case of $\geq_{\mathsf{proc}} \cdot \pi \cdot \geq_{\mathsf{proc}}$ and $(\geq_{\mathsf{proc}} \cdot \pi \cdot \geq_{\mathsf{proc}})^{\mathsf{c}}$ are similar, switching upward-closed and downward-closed. □

Suppose that

$$M, e \models \bigwedge_{1 \leq i,j \leq n} \mathsf{Loop}\left(\pi_i \cdot \{\varphi \wedge p\}? \cdot \pi_j^{-1}\right),$$

and let us show that

$$M, e \models \langle \pi_1 \cap \cdots \cap \pi_n \rangle (\varphi \wedge p).$$

For all $p \in \mathsf{Procs}$ and $1 \leq i \leq n$, we let $I_i = [\![\pi_i]\!](e) \cap [\![\varphi \wedge p]\!]$. By Claim 5.48, I_i is an interval of $([\![\varphi \wedge p]\!], \leq_{\mathsf{proc}})$. Besides, for all i, j,

$$M, e \models \mathsf{Loop}\left(\pi_i \cdot \{\varphi \wedge p\}? \cdot \pi_j^{-1}\right)$$

implies $I_i \cap I_j \neq \emptyset$. By Lemma 3.18, we obtain

$$I_1 \cap \cdots \cap I_n \neq \emptyset,$$

that is,

$$M, e \models \langle \pi_1 \cap \cdots \cap \pi_n \rangle (\varphi \wedge p). \qquad \square$$

All that remains to be done for the translation to $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ is to express formulas $\mathsf{Loop}(\pi_i \cdot \{\varphi \wedge p\}? \cdot \pi_j)$, for path formulas π_i and π_j as in Lemma 5.47, in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$. To do so, we first express all formulas π_i as unions of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ formulas. Note that we cannot apply directly Lemma 5.24 to formulas containing $\leq_{\mathsf{proc}}$, which is not in the syntax of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$. However, the next lemma is similar.

Lemma 5.49. *For all $\top$-free path formula $\pi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, over MSCs,*

$$(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})^{\mathsf{c}} \equiv \{\neg\langle\pi\rangle \wedge \neg\langle <_{\mathsf{proc}} \cdot \pi\rangle\}? \cdot \top + \sum_{(p,q) \notin \mathsf{Comp}(\pi)} \{p\}? \cdot \top \cdot \{q\}? + (\mathsf{min}\ (\pi) \cdot >_{\mathsf{proc}}) + (\{\neg\langle\pi\rangle\}? \cdot \mathsf{min}\ (<_{\mathsf{proc}} \cdot \pi) \cdot >_{\mathsf{proc}}).$$

Proof. For e such that $[\![\leq_{\mathsf{proc}} \cdot \pi]\!](e) = \emptyset$, i.e., $e \in [\![\neg\langle\pi\rangle \wedge \neg\langle <_{\mathsf{proc}} \cdot \pi\rangle]\!]$, we have

$$[\![(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})^{\mathsf{c}}]\!](e) = E = [\![\{\neg\langle\pi\rangle \wedge \neg\langle <_{\mathsf{proc}} \cdot \pi\rangle\}? \cdot \top]\!](e).$$

Now, let e such that $[\![\leq_{\mathsf{proc}} \cdot \pi]\!](e) \neq \emptyset$. Since π is $\top$-free, $f = \min[\![\leq_{\mathsf{proc}} \cdot \pi]\!](e)$ is well-defined. We then have

$$[\![(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})^{\mathsf{c}}]\!](e) = \{g \in E \mid g <_{\mathsf{proc}} f\} \cup \bigcup_{(loc(e),q) \notin \mathsf{Comp}(\pi)} E_q$$

Finally, notice that by monotonicity of $[\![\mathsf{min}\ \pi]\!]$, for all e', f', we have

$$f' = \min[\![\leq_{\mathsf{proc}} \cdot \pi]\!](e') \iff (e', f') \in [\![\mathsf{min}\ \pi + \{\neg\langle\pi\rangle\}? \cdot \mathsf{min}\ (<_{\mathsf{proc}} \cdot \pi)]\!] .$$

Thus,

$$\{g \in E \mid g <_{\mathsf{proc}} f\} = [\![\mathsf{min}\ (\pi) \cdot >_{\mathsf{proc}} + \{\neg\langle\pi\rangle\}? \cdot \mathsf{min}\ (<_{\mathsf{proc}} \cdot \pi) \cdot >_{\mathsf{proc}}]\!](e) ,$$

and

$$[\![(\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}})^{\mathsf{c}}]\!](e) = \left[\!\!\left[\begin{array}{l} \mathsf{min}\ (\pi) \cdot >_{\mathsf{proc}} + \\ \{\neg\langle\pi\rangle\}? \cdot \mathsf{min}\ (<_{\mathsf{proc}} \cdot \pi) \cdot >_{\mathsf{proc}} + \\ \sum_{(p,q) \notin \mathsf{Comp}(\pi)} \{p\}? \cdot \top \cdot \{q\}? \end{array} \right]\!\!\right](e) .$$

□

For $\varphi\, \mathsf{SU}\, \psi$, we obtain:

Lemma 5.50. *For all $\varphi, \psi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, there exists an event formula in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, denoted $\varphi\, \mathsf{SU}\, \psi$, such that for all $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$,*

$$[\![\varphi\, \mathsf{SU}\, \psi]\!] = \{e \in E \mid \exists f \in [\![\psi]\!], e < f \wedge \forall g \in E, e < g < f \implies g \in [\![\varphi]\!]\} ,$$

and

$$\|\mathcal{S}(\varphi\, \mathsf{SU}\, \psi) \setminus (\mathcal{S}(\varphi) \cup \mathcal{S}(\psi))\| = 2^{\mathcal{O}(|\mathsf{Procs}| \cdot \log |\mathsf{Procs}|)} .$$

Proof. Denote by $\pi_1, \ldots, \pi_N$ all path formulas of the form

$$(\leq_{\mathsf{proc}} \cdot \pi \cdot \{\neg\varphi\}? \cdot \pi' \cdot \leq_{\mathsf{proc}})^{\mathsf{c}} ,$$

where $\pi, \pi' \in \Pi^+$. Note that $N = |\Pi^+|^2 = 2^{\mathcal{O}(|\mathsf{Procs}| \cdot \log |\mathsf{Procs}|)}$.

As explained page 112, the formula $\varphi\, \mathsf{SU}\, \psi$ should be equivalent to

$$\varphi_1 = \bigvee_{\pi_0 \in \{\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}} \mid \pi \in \Pi^+\}} \langle \pi_0 \cap \pi_1 \cap \cdots \cap \pi_N \rangle\, \psi .$$

By Lemma 5.47, φ_1 is equivalent to

$$\varphi_2 = \bigvee_{\pi_0 \in \{\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}} \mid \pi \in \Pi^+\}} \bigvee_{p \in \mathsf{Procs}} \bigwedge_{0 \leq i,j \leq N} \mathsf{Loop}\left(\pi_i \cdot \{\psi \wedge p\}? \cdot \pi_j^{-1}\right) .$$

Moreover, for all $0 \leq i \leq N$, we can construct path formulas $\pi_{i,1}, \ldots, \pi_{i,n_i} \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ such that

$$\begin{aligned} \pi_i &\equiv \pi_{i,1} + \cdots + \pi_{i,n_i} \\ n_i &= \mathcal{O}(|\mathsf{Procs}|^2) \\ \mathit{length}(\pi_{i,j}) &= \mathcal{O}(|\mathsf{Procs}|) \\ \|\mathcal{S}(\pi_{i,j}) \setminus \mathcal{S}(\varphi)\| &= \mathcal{O}(|\mathsf{Procs}|)\,. \end{aligned}$$

Indeed, for $i = 0$ and $\pi_0 = \leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}}$, we can simply take

$$\pi_i \equiv \pi + (<_{\mathsf{proc}} \cdot \pi) + (\pi \cdot <_{\mathsf{proc}}) + (<_{\mathsf{proc}} \cdot \pi \cdot <_{\mathsf{proc}})\,.$$

For $i \geq 1$ and $\pi_i = (\leq_{\mathsf{proc}} \cdot \pi \cdot \{\neg\varphi\}? \cdot \pi' \cdot \leq_{\mathsf{proc}})^{\mathsf{c}}$, we apply Lemma 5.49 to the formula $\pi \cdot \{\neg\varphi\}? \cdot \pi'$. Recall from Lemma 5.23 that, in general,

$$\mathit{length}(\mathsf{min}\ \pi) = \mathcal{O}(\mathit{length}(\pi)) \qquad \text{and} \qquad \|\mathcal{S}(\mathsf{min}\ \pi) \setminus \mathcal{S}(\pi)\| = \mathcal{O}(\mathit{length}(\pi))\,.$$

So we obtain $\mathit{length}(\pi_{i,j}) = \mathcal{O}(|\mathsf{Procs}|)$ and $\|\mathcal{S}(\pi_{i,j}) \setminus \mathcal{S}(\varphi)\| = \mathcal{O}(|\mathsf{Procs}|)$ from the fact that $\mathit{length}(\pi \cdot \{\neg\varphi\}? \cdot \pi') = \mathcal{O}(|\mathsf{Procs}|)$ and $\mathcal{S}(\pi \cdot \{\neg\varphi\}? \cdot \pi') = \mathcal{S}(\neg\varphi)$.

We then let

$$\begin{aligned} \varphi\ \mathsf{SU}\ \psi &= \bigvee_{\pi_0 \in \{\leq_{\mathsf{proc}} \cdot \pi \cdot \leq_{\mathsf{proc}} \mid \pi \in \Pi^+\}} \bigvee_{p \in \mathsf{Procs}} \bigwedge_{0 \leq i,j \leq N} \bigvee_{\substack{1 \leq k \leq n_i \\ 1 \leq \ell \leq n_j}} \mathsf{Loop}\left(\pi_{i,k} \cdot \{\psi \wedge p\}? \cdot \pi_{j,\ell}^{-1}\right) \\ &\equiv \varphi_2 \equiv \varphi_1\,. \end{aligned}$$

We then have

$$\begin{aligned} \|\mathcal{S}(\varphi\ \mathsf{SU}\ \psi) \setminus (\mathcal{S}(\varphi) \cup \mathcal{S}(\psi))\| &= \mathcal{O}(|\Pi^+| \times |\mathsf{Procs}| \times |\Pi^+|^4 \times |\mathsf{Procs}|^4 \times |\mathsf{Procs}|) \\ &= 2^{\mathcal{O}(|\mathsf{Procs}| \cdot \log |\mathsf{Procs}|)}\,. \end{aligned}$$

□

Symmetrically, we also have:

Lemma 5.51. *For all $\varphi, \psi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, there exists an event formula in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, denoted $\varphi\ \mathsf{SS}\ \psi$, such that for all $M = (E, \rightarrow, \lhd, \mathit{loc}, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$,*

$$[\![\varphi\ \mathsf{SS}\ \psi]\!] = \{e \in E \mid \exists f \in [\![\psi]\!], f < e \wedge \forall g \in E, f < g < e \implies g \in [\![\varphi]\!]\}\,,$$

and

$$\|\mathcal{S}(\varphi\ \mathsf{SS}\ \psi) \setminus (\mathcal{S}(\varphi) \cup \mathcal{S}(\psi))\| = 2^{\mathcal{O}(|\mathsf{Procs}| \cdot \log |\mathsf{Procs}|)}\,.$$

Proof. The proof is the same as the proof of Lemma 5.50, except we consider the converses of the formulas π_i and $\pi_{i,j}$ defined for SU. □

Similarly, for $\mathsf{Co}\,\varphi$, we obtain:

Lemma 5.52. *For all $\varphi \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, there exists an event formula in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, denoted $\mathsf{Co}\,\varphi$, such that for all $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$,*

$$[\![\mathsf{Co}\,\varphi]\!] = \{e \in E \mid \exists f \in [\![\varphi]\!], e \parallel f\}\,,$$

and

$$\|\mathcal{S}(\mathsf{Co}\,\varphi) \setminus \mathcal{S}(\varphi)\| = 2^{\mathcal{O}(|\mathsf{Procs}| \cdot \log |\mathsf{Procs}|)}\,.$$

Proof. The proof is essentially the same as the proof for $\varphi\,\mathsf{SU}\,\psi$ (Lemma 5.50). This time, $\pi_1, \ldots, \pi_N$ denote the set of all path formulas of the form $(\leq_{\mathsf{proc}} \cdot\, \pi \cdot \leq_{\mathsf{proc}})^{\mathsf{c}}$ or $(\geq_{\mathsf{proc}} \cdot\, \pi^{-1} \cdot \geq_{\mathsf{proc}})^{\mathsf{c}}$, for $\pi \in \Pi$. We apply Lemmas 5.47 and 5.49 to obtain a $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ formula $\mathsf{Co}\,\varphi$ equivalent to

$$\langle \pi_1 \cap \cdots \cap \pi_N \rangle\, \varphi$$

and such that $\|\mathcal{S}(\mathsf{Co}\,\varphi) \setminus \mathcal{S}(\varphi)\| = 2^{\mathcal{O}(|\mathsf{Procs}| \cdot \log |\mathsf{Procs}|)}$. □

Lemmas 5.50, 5.51 and 5.52 lead to an efficient translation from the temporal logic $\mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle \lhd \rangle, \langle \lhd^{-1} \rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ to $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$. Given a formula $\varphi \in \mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle \lhd \rangle, \langle \lhd^{-1} \rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ we denote by $\mathcal{S}(\varphi)$ the set of subformulas of φ, and we let $\|\varphi\| = |\mathcal{S}(\varphi)|$.

Theorem 5.53. *For every formula $\varphi \in \mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle \lhd \rangle, \langle \lhd^{-1} \rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$, there exists an equivalent formula $\varphi' \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ such that $\|\varphi'\| = 2^{\mathcal{O}(|\mathsf{Procs}| \log |\mathsf{Procs}|)} \|\varphi\|$.*

Proof. We construct φ' by induction on φ, in such a way that

$$\left\| \mathcal{S}(\varphi') \setminus \textstyle\bigcup_{\psi \in \mathcal{S}(\varphi) \setminus \{\varphi\}} \mathcal{S}(\psi') \right\| = 2^{\mathcal{O}(|\mathsf{Procs}| \cdot \log |\mathsf{Procs}|)}\,,$$

which implies $\|\varphi'\| = 2^{\mathcal{O}(|\mathsf{Procs}| \log |\mathsf{Procs}|)} \|\varphi\|$.

The cases $\varphi = \mathit{true}$, $\varphi = p$, $\varphi = P$, $\varphi = \varphi_1 \vee \varphi_2$ and $\varphi = \neg\varphi$ are straightforward. If $\varphi = \langle \lhd \rangle\, \psi$, we let

$$\varphi' = \bigvee_{(p,q) \in Ch} \langle \lhd_{p,q} \rangle\, \psi'\,,$$

and similarly for $\varphi = \langle \lhd^{-1} \rangle\, \psi$. If $\varphi = \varphi_1\, \mathsf{SU}\, \varphi_2$, we apply Lemma 5.50. If $\varphi = \varphi_1\, \mathsf{SS}\, \varphi_2$, we apply Lemma 5.51. If $\varphi = \mathsf{Co}\,\psi$, we apply Lemma 5.52. □

Chapter 6

From logic to CFMs

This chapter presents one of the main result of the thesis: namely, that every FO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$] formula can be translated into an equivalent CFM. This proves that CFMs are exactly as expressive as EMSO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$]. As expected, this translation uses $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ as an intermediate step. It is worth noting that, while the translation from first-order logic to automata is non-elementary, it is only exponential in the size of the formula for $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ or for temporal logics.

In contrast with some of the results cited in Section 4.3, our translation from FO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$] to CFMs applies even in the case of unbounded channels. However, it also has interesting consequences for the case of bounded MSCs. In Section 6.4, we apply it to obtain a new proof of the equivalence, for languages of existentially B-bounded MSCs, of the notions of recognizability by a CFM, definability in MSO, and regularity of the corresponding (word) language of linearizations. This was initially proven in [34] for finite MSCs. We also extends the result to infinite MSCs.

6.1 Star-free PDL

Let us first prove that every $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ formula can be translated into an equivalent CFM. Using the Loop elimination procedure from Chapter 5, this comes down to a problem very similar to the translation of LTL into automata.

Theorem 6.1. *For every* $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ *sentence* ξ*, one can construct a CFM* $\mathcal{A}_\xi$ *with* $2^{\mathcal{O}(\|\xi\|)}$ *states per process such that* $\mathbb{L}(\mathcal{A}_\xi) = \mathbb{L}(\xi)$*.*

Proof. By Theorem 5.28, it is sufficient to consider the case where ξ does not contain any occurrence of Loop. Indeed, for all other formulas ξ, we can first apply Theorem 5.28 to obtain $\mathsf{Prop}' \supseteq \mathsf{Prop}$ and a formula $\xi' \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}']$

without occurrences of Loop such that

$$\mathbb{L}(\xi) = \mathrm{Pr}_{\mathsf{Prop}}(\mathbb{L}(\xi')) \qquad \text{and} \qquad \|\xi'\| = \mathcal{O}(\|\xi\|)\,.$$

The CFM $\mathcal{A}_\xi$ is then obtained by projection from the CFM $\mathcal{A}_{\xi'}$: $\mathcal{A}_\xi$ guesses an interpretation of $\mathsf{Prop}' \setminus \mathsf{Prop}$, and simulates $\mathcal{A}_{\xi'}$.

So, suppose that ξ contains no occurrence of Loop. We further assume that every model of ξ is nonempty. If that is not the case, we obtain the CFM $\mathcal{A}_\xi$ by a simple modification of $\mathcal{A}_{\xi \wedge \mathsf{E}\, true}$: assuming the initial states have no incoming transitions, we simply change the accepting condition to accept runs where all processes end in their initial states, that is, the unique run over the empty MSC.

Over nonempty MSCs, we have

$$\mathsf{E}\,\varphi \equiv \mathsf{A}(\langle\top\rangle\,\varphi)\,, \qquad \mathsf{A}\,\varphi \vee \mathsf{A}\,\psi \equiv \mathsf{A}([\top]\,\varphi \vee [\top]\,\psi)\,, \qquad \neg\,\mathsf{A}\,\varphi \equiv \mathsf{A}(\langle\top\rangle\,\neg\varphi)\,,$$

so we can rewrite ξ into a formula of the form $\mathsf{A}\,\varphi$ and of linear size. Using the equivalences $\langle\pi_1 \cdot \pi_2\rangle\,\varphi \equiv \langle\pi_1\rangle\,(\langle\pi_2\rangle\,\varphi)$ and $\langle\{\varphi\}?\rangle\,\psi \equiv \varphi \wedge \psi$, we can further assume that φ follows the syntax

$$\begin{aligned}\varphi ::= {} & \mathit{true} \mid p \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \\ & \langle \lhd_{p,q}\rangle\,\varphi \mid \langle \lhd_{p,q}^{-1}\rangle\,\varphi \mid \langle (<_{\mathsf{proc}})_\varphi\rangle\,\varphi \mid \langle (>_{\mathsf{proc}})_\varphi\rangle\,\varphi \mid \langle\top\rangle\,\varphi\,,\end{aligned}$$

where $p \in \mathsf{Procs}$, $P \in \mathsf{Prop}$, $(p,q) \in \mathit{Ch}$. Note that after these rewritings, we still have $\|\,\mathsf{A}\,\varphi\| = \mathcal{O}(\|\xi\|)$.

We construct $\mathcal{A}_\xi = \mathcal{A}_{\mathsf{A}\,\varphi} = ((\mathcal{A}_p)_{p\in\mathsf{Procs}}, \mathit{Msg}, \mathit{Acc})$ similarly to classic translations from LTL over words to automata. Intuitively, the states of each process contain subsets of $\mathcal{S}(\varphi)$, and the state of the CFM after reaching event e is the set s of subformulas satisfied by e. In addition, each process also remembers, for every subformula $\langle\top\rangle\,\psi \in \mathcal{S}(\varphi)$, whether the formula ψ has already been satisfied on that process, that is, the value of $\langle >_{\mathsf{proc}}\rangle\,\psi$. Finally, in order for the CFM to be able to check the value of subformulas $\langle \lhd_{p,q}\rangle\,\varphi$ or $\langle \lhd_{p,q}^{-1}\rangle\,\varphi$, at each send event, the process sends its current state to the receiving process. A process receiving a message from channel (p,q) then sets the value for formulas ψ such that $\langle \lhd_{p,q}\rangle\,\psi \in \mathcal{S}(\varphi)$ and for formulas $\langle \lhd_{p,q}^{-1}\rangle\,\psi'$ according to whether $\langle \lhd_{p,q}\rangle\,\psi$ or ψ' are contained in the message received.

For all $p \in \mathsf{Procs}$, let S_p be the set consisting of an initial state ι_p, and all sets of subformulas

$$s \subseteq \mathcal{S}(\varphi) \cup \{\langle >_{\mathsf{proc}}\rangle\,\psi \mid \langle\top\rangle\,\psi \in \mathcal{S}(\varphi)\}$$

satisfying the following conditions:

1. $\varphi \in s$.
2. If $\mathit{true} \in \mathcal{S}(\varphi)$, then $\mathit{true} \in s$.

3. For all $q \in \mathcal{S}(\varphi) \cap \mathsf{Procs}$, $q \in s$ if and only if $q = p$.
4. For all $\varphi_1 \vee \varphi_2 \in \mathcal{S}(\varphi)$, $\varphi_1 \vee \varphi_2 \in s$ if and only if $\varphi_1 \in s$ or $\varphi_2 \in s$.
5. For all $\neg\psi \in \mathcal{S}(\varphi)$, $\neg\psi \in s$ if and only if $\psi \notin s$.
6. For all $\langle\top\rangle\,\psi \in \mathcal{S}(\varphi)$, $\psi \in s$ or $\langle >_{\mathsf{proc}}\rangle\,\psi \in s$ implies $\langle\top\rangle\,\psi \in s$.
7. For all $\langle\lhd_{q,r}\rangle\,\psi \in \mathcal{S}(\varphi)$ such that $q \neq p$, $\langle\lhd_{q,r}\rangle\,\psi \notin s$.
8. For all $\langle\lhd_{q,r}^{-1}\rangle\,\psi \in \mathcal{S}(\varphi)$ such that $r \neq p$, $\langle\lhd_{q,r}^{-1}\rangle\,\psi \notin s$.

Let also

$$Msg = \bigcup_{p \in \mathsf{Procs}} S_p\,.$$

For each process p, we let $\mathcal{A}_p = (S_p, \iota_p, \Delta_p)$, where Δ_p is the set of all $t \in S_p \times Act_p(Msg) \times S_p$ such that:

9. $target(t) \neq \iota_p$.
10. For all $P \in \mathcal{S}(\varphi) \cap \mathsf{Prop}$, $P \in target(t)$ if and only if $P \in label(t)$.
11. For all $\langle\lhd_{p,q}\rangle\,\psi \in \mathcal{S}(\varphi)$, if t is an internal transition, a receive transition, or a send transition such that $receiver(t) \neq q$, then $\langle\lhd_{p,q}\rangle\,\psi \notin target(t)$.
12. For all $\langle\lhd_{q,p}^{-1}\rangle\,\psi \in \mathcal{S}(\varphi)$, if t is an internal transition, a send transition, or a receive transition such that $sender(t) \neq q$, then $\langle\lhd_{q,p}^{-1}\rangle\,\psi \notin target(t)$.
13. If t is a send transition, then $msg(t) = target(t)$.
14. If t is a receive transition and $sender(t) = q$, then for all $\langle\lhd_{q,p}\rangle\,\psi \in \mathcal{S}(\varphi)$, $\psi \in target(t)$ if and only if $\langle\lhd_{q,p}\rangle\,\psi \in msg(t)$.
15. If t is a receive transition and $sender(t) = q$, then for all $\langle\lhd_{q,p}^{-1}\rangle\,\psi \in \mathcal{S}(\varphi)$, $\langle\lhd_{q,p}^{-1}\rangle\,\psi \in target(t)$ if and only if $\psi \in msg(t)$.
16. If $source(t) \neq \iota_p$, then for all $\langle\top\rangle\,\psi \in \mathcal{S}(\varphi)$, $\langle\top\rangle\,\psi \in target(t)$ if and only if $\langle\top\rangle\,\psi \in source(t)$.
17. If $source(t) \neq \iota_p$, then for all $\langle(<_{\mathsf{proc}})_\psi\rangle\,\psi' \in \mathcal{S}(\varphi)$, $\langle(<_{\mathsf{proc}})_\psi\rangle\,\psi' \in source(t)$ if and only if $\psi' \in target(t)$ or $\psi \in target(t)$ and $\langle(<_{\mathsf{proc}})_\psi\rangle\,\psi' \in target(t)$.
18. For all $\langle(>_{\mathsf{proc}})_\psi\rangle\,\psi' \in \mathcal{S}(\varphi)$, $\langle(>_{\mathsf{proc}})_\psi\rangle\,\psi' \in target(t)$ if and only if $\psi' \in source(t)$ or $\psi \in source(t)$ and $\langle(>_{\mathsf{proc}})_\psi\rangle\,\psi' \in source(t)$. In particular, if $source(t) = \iota_p$, then $\langle(>_{\mathsf{proc}})_\psi\rangle\,\psi' \notin target(t)$.

Finally, Acc is defined so that a run is accepting if and only if:

19. For all $\langle\top\rangle\,\psi \in \mathcal{S}(\varphi)$, either (i) all processes go infinitely often through or end in a state containing $\langle\top\rangle\,\psi$, or end in their initial state, and at least one process goes infinitely often or ends in a state containing ψ or $\langle{>_{\mathsf{proc}}}\rangle\,\psi$, or (ii) all processes go infinitely often through or end in a state *not* containing $\langle\top\rangle\,\psi$.

20. For all $\langle({<_{\mathsf{proc}}})_{\psi}\rangle\,\psi' \in \mathcal{S}(\varphi)$, all processes either go infinitely often through or end in a state containing ψ', or go infinitely often through or end in a state *not* containing $\langle({<_{\mathsf{proc}}})_{\psi}\rangle\,\psi' \in \mathcal{S}(\varphi)$. In addition, no process ends in a state containing $\langle({<_{\mathsf{proc}}})_{\psi}\rangle\,\psi'$.

It is easy to check that for all $M \in \mathbb{L}(\mathsf{A}\,\varphi)$, there is an accepting run ρ of $\mathcal{A}_{\mathsf{A}\,\varphi}$ on M, defined by setting $\mathit{target}(\rho(e))$ to be the set of formulas satisfied by e, for all events e.

Conversely, for all runs ρ of $\mathcal{A}_{\mathsf{A}\,\varphi}$ on an MSC M, we can prove by induction on ψ that for all $\psi \in \mathcal{S}(\varphi) \cup \{\langle{>_{\mathsf{proc}}}\rangle\,\psi \mid \langle\top\rangle\,\psi \in \mathcal{S}(\varphi)\}$, for all events e, we have $\psi \in \mathit{target}(\rho(e))$ if and only if $M, e \models \psi$.

For $\psi = \mathit{true}$, it follows from condition 2. For $\psi = p$, it follows from condition 3, and for $\psi = P$, from condition 10. For $\psi = \psi_1 \vee \psi_2$, it follows from the induction hypothesis and condition 4, and for $\psi = \neg\psi'$, condition 5.

For $\psi = \langle{\lhd_{p,q}}\rangle\,\psi'$, if e is not a send on channel (p, q), then $M, e \not\models \psi$ and by conditions 7 and 11, ψ is not in $\mathit{target}(\rho(e))$. Otherwise, by conditions 13 and 14, we have $\psi \in \mathit{target}(\rho(e))$ if and only if $\psi' \in \mathit{target}(\rho(f))$ for the matching receive event f, and we apply the induction hypothesis to ψ'.

Similarly, for $\psi = \langle{\lhd^{-1}_{p,q}}\rangle\,\psi'$, the result follows from conditions 8 and 12 if e is not a receive from channel (p, q), and from conditions 13 and 15 otherwise.

For $\psi = \langle\top\rangle\,\psi'$, by condition 16, the value guessed by the CFM for $\langle\top\rangle\,\psi'$ is the same for all events along a given process. By condition 19, it is also the same on all processes. If $\langle\top\rangle\,\psi'$ holds, then by condition 6, we have $\psi \in \mathit{target}(\rho(e))$ for all events e. Conversely, if $\psi \in \mathit{target}(\rho(e))$ for all events e, then by condition 19, at least one process must contain an event satisfying ψ'.

For $\psi = \langle({>_{\mathsf{proc}}})_{\psi'}\rangle\,\psi''$, the result follows from condition 18 and the fact that

$$\langle({>_{\mathsf{proc}}})_{\psi'}\rangle\,\psi'' \equiv \langle{\leftarrow}\rangle\,\psi'' \vee \langle{\leftarrow}\rangle\left(\psi' \wedge \langle({>_{\mathsf{proc}}})_{\psi'}\rangle\,\psi''\right).$$

For $\psi = \langle({<_{\mathsf{proc}}})_{\psi'}\rangle\,\psi''$, the result follows from conditions 17 and 20 and the fact that

$$\langle({<_{\mathsf{proc}}})_{\psi'}\rangle\,\psi'' \equiv \langle{\rightarrow}\rangle\,\psi'' \vee \langle{\rightarrow}\rangle\left(\psi' \wedge \langle({<_{\mathsf{proc}}})_{\psi'}\rangle\,\psi''\right).$$

Since $\psi \in \mathit{target}(\rho(e))$ if and only if $M, e \models \psi$, then in particular, for $\psi = \varphi$, condition 1 implies that for all events, $M, e \models \varphi$. Thus, $M \models \mathsf{A}\,\varphi$. □

6.2 First-order logic and EMSO

As an immediate consequence of Theorems 5.27 and 6.1, we obtain:

Proposition 6.2. $\mathcal{L}(\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]) \subseteq \mathcal{L}(\mathrm{CFM}[\mathsf{Procs}, \mathsf{Prop}])$.

Since $\mathcal{L}(\mathrm{CFM}[\mathsf{Procs}, \mathsf{Prop}])$ is closed under projection, Proposition 6.2 implies that $\mathcal{L}(\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]) \subseteq \mathcal{L}(\mathrm{CFM}[\mathsf{Procs}, \mathsf{Prop}])$. Conversely, CFMs can be translated into $\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ in a standard way [11].

Theorem 6.3. $\mathcal{L}(\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]) = \mathcal{L}(\mathrm{CFM}[\mathsf{Procs}, \mathsf{Prop}])$.

Remark 6.4. Our translation from $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ to $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$, and therefore from $\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ to CFMs, is non-elementary. This is unavoidable, already when $|\mathsf{Procs}| = 1$ [80].

6.3 Temporal logics

Combining Proposition 5.53 and Theorem 6.1, we also obtain a translation from $\mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle\lhd\rangle, \langle\lhd^{-1}\rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ to CFMs. Contrarily to the translation from first-order logic to CFMs, the size of the resulting CFM is only exponential in the size of the formula, and doubly exponential in the number of processes. In particular, for $|\mathsf{Procs}| = 1$, it corresponds to the complexity of the translation from LTL to automata.

More precisely, given a formula $\varphi \in \mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle\lhd\rangle, \langle\lhd^{-1}\rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$, the language we want to recognize is the set of MSCs extended with an extra labeling indicating the value of φ at each event. For all $M \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$, we denote by $M_\varphi \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop} \uplus \{P_\varphi\})$ the extension of M to $\mathsf{Prop} \uplus \{P_\varphi\}$ defined by

$$e \in [\![P_\varphi]\!]^{M_\varphi} \quad \text{if} \quad M, e \models \varphi\,.$$

Theorem 6.5. *For every $\varphi \in \mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle\lhd\rangle, \langle\lhd^{-1}\rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$, one can construct a CFM $\mathcal{A}_\varphi$ with $2^{\|\varphi\| \cdot 2^{\mathcal{O}(|\mathsf{Procs}| \log |\mathsf{Procs}|)}}$ states per process such that*

$$\mathbb{L}(\mathcal{A}_\varphi) = \{M_\varphi \mid M \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})\}\,.$$

Proof. Let $\varphi' \in \mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ be the formula given by Proposition 5.53, and $\xi = \mathsf{A}(P_\varphi \iff \varphi')$. We have $\|\xi\| = \|\varphi\| \cdot 2^{\mathcal{O}(|\mathsf{Procs}| \log |\mathsf{Procs}|)}$, and by Theorem 6.1, we can construct a CFM accepting $\mathbb{L}(\xi) = \{M_\varphi \mid M \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})\}$, with $2^{\|\varphi\| \cdot 2^{\mathcal{O}(|\mathsf{Procs}| \log |\mathsf{Procs}|)}}$ states per process. □

6.4 Existentially-bounded MSCs

As mentioned in Section 4.3, many results from the literature have been concerned with the case of bounded channels. In particular, under that assumption, a translation from logical formulas to CFMs is possible not only for EMSO, but for the full MSO logic.

In this section, we show that some difficult results for bounded CFMs can be obtained as corollaries of our translation from first-order logic to CFMs. We also extend them to infinite MSCs.

Recall that the basic definitions and notations related to B-bounded linearizations and existentially B-bounded MSCs are given in Chapter 4, Section 4.1.3.

6.4.1 Known results

For existentially bounded sets of MSCs, the notions of recognizability by a CFM, definability in MSO, and regularity of the language of linearizations coincide [34]. This is stated more formally below. The original proof of this result [34] relies on the theory of Mazurkiewicz traces, and the construction of a CFM recognizing the set $\mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$ of $\exists B$-bounded MSCs.

Linearizations as words $M_{\preceq}$

Definition 6.6. With every MSC $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ and linearization $e_1 \prec e_2 \prec e_3 \prec \cdots$ of M, we associate a word as follows. Given $e \in E$, we write $type(e) = p$ if e is an internal event on process p, $type(e) = p!q$ if e is a write on channel (p, q), and $type(e) = q?p$ if e is a read from channel (p, q). We denote by $M_{\preceq}$ the (finite or infinite, depending on M) word

$$M_{\preceq} = (\lambda \times type)(e_1) \cdot (\lambda \times type)(e_2) \cdot (\lambda \times type)(e_3) \cdots \in A_{lin}^* \cup A_{lin}^\omega$$

where $A_{lin} = 2^{\mathsf{Prop}} \times (\mathsf{Procs} \cup \{p?q, p!q \mid (p, q) \in Ch\})$. We also let

$$Lin_B(M) = \{M_{\preceq} \mid \preceq \text{ is a } B\text{-bounded linearization of } M\}\,.$$

Remark 6.7. M can be retrieved from any of its linearizations $M_{\preceq}$.

The result cited above can be stated formally as follows:

Theorem 6.8 ([34, Theorem 4.1])**.** *Let $B \in \mathbb{N}$ and $L \subseteq \mathbb{MSC}^{\text{fin}}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$. The following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$.*
2. *$L = \mathbb{L}(\Phi)$ for some $\text{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ formula Φ.*
3. *$Lin_B(L)$ is a regular language (of finite words).*

In [34], Theorem 6.8 was proved for finite MSCs. In the remainder of the section, we give a new, direct (not relying on reductions to Mazurkiewicz traces) proof of the difficult direction, which is (3) $\implies$ (1), and extend the result to infinite MSCs.

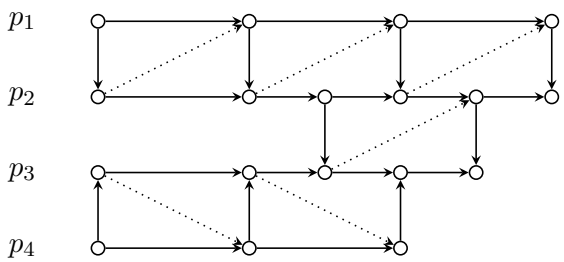


Figure 6.1: The relation rev_B for $B = 1$.

6.4.2 A CFM for existentially-bounded MSCs

An important part of the original proof of Theorem 6.8 [34] is the construction of a CFM recognizing the set $\mathbb{MSC}^{\text{fin}}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$ of finite $\exists B$-bounded MSCs. Here, we show that such a CFM can be obtained very easily as an application of Proposition 6.2, both for finite or infinite MSCs. Indeed, the set of $\exists B$-bounded MSCs is FO$[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$-definable. This follows from a characterization of $\exists B$-bounded MSCs dating back from [61], explained below.

Let $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$. We denote by $rev_B \subseteq E \times E$ the relation consisting of the set of pairs (f, g) such that f is a receive event from some channel (p, q) with corresponding send event $e \lhd f$, and g is the B-th send on channel (p, q) after event e. The relation rev_B is illustrated in Figure 6.1 (represented by the dotted edges) for $B = 1$ and an $\exists 1$-bounded MSC. It can be defined in FO$[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$ as follows: rev_B

$$\begin{aligned} rev_B(x, y) := \exists x_0, x_1, \ldots, x_B, y_0, y_1, \ldots, y_B.\ & \\ x_0 = x \wedge y_B = y \wedge \bigwedge_{0 \leq i \leq B} y_i \lhd x_i \wedge \bigwedge_{0 \leq i < B} & \big(x_i <_{\mathsf{proc}} x_{i+1} \wedge z_i <_{\mathsf{proc}} z_{i+1} \\ \wedge \neg(\exists x', y'.\ x_i <_{\mathsf{proc}} x' <_{\mathsf{proc}} x_{i+1} \wedge y_i <_{\mathsf{proc}} y' <_{\mathsf{proc}} y_{i+1} \wedge y' \lhd x')\big)\,. & \end{aligned}$$

Lemma 6.9 ([61, Lemma 3.1])**.** *A linearization $\preceq$ of an MSC M is B-bounded if and only if it contains rev_B. In particular, M is $\exists B$-bounded if and only if the relation $(< \cup rev_B)$ is acyclic.*

The following proposition generalizes [34, Proposition 5.14], stating that the set of finite $\exists B$-bounded MSCs is recognizable by a CFM, to arbitrary (finite or infinite) MSCs.

Proposition 6.10. $\mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop}) \in \mathcal{L}(\text{CFM}[\mathsf{Procs}, \mathsf{Prop}])$.

Proof. If $(< \cup rev_B)$ contains a cycle, then it contains one of length at most $2|\mathsf{Procs}|$ (entering and leaving each process at most once). Therefore, by Lemma 6.9, M is

$\exists B$-bounded if and only if it satisfies the following formula:

$$\Phi_{\exists B} = \bigwedge_{1 \leq n \leq 2|\mathsf{Procs}|} \neg\Big(\exists x_0, \ldots, x_n.\ x_0 = x_n \wedge \bigwedge_{0 \leq i < n} x_i < x_{i+1} \vee rev_B(x_i, x_{i+1})\Big)\,.$$

Applying Theorem 6.3, we can construct a CFM $\mathcal{A}_{\exists B}$ such that

$$\mathbb{L}(\mathcal{A}_{\exists B}) = \mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})\,. \qquad \square$$

Remark 6.11. We could also express rev_B and the acyclicity of $(< \cup rev_B)$ directly in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ instead of going through first-order logic: the relation rev_B can be expressed by the path formula

$$\mathsf{rev}_B = \sum_{(p,q) \in Ch} \lhd_{p,q}^{-1} \cdot \Big((<_{\mathsf{proc}})_{\neg\langle \lhd_{p,q} \rangle} \cdot \{\langle \lhd_{p,q} \rangle\}? \Big)^B$$

and the set of $\exists B$-bounded MSCs, that is, MSCs such that $(< \cup rev_B)$ is acyclic, by the sentence

$$\bigwedge_{1 \leq n \leq |\mathsf{Procs}|} \neg\, \mathsf{E}\, \mathsf{Loop}\, (((\lhd + \mathsf{rev}_B) \cdot <)^n)\ ,$$

where $\lhd := \sum_{(p,q) \in Ch} \lhd_{p,q}$. Note that, strictly speaking, these formulas are not in $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ since $+$ is not allowed in the syntax. However, the latter Loop formula can easily be expanded into a formula of $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$.

6.4.3 FO-definable linearizations for existentially-bounded MSCs

The second ingredient to the new proof of Theorem 6.8 is the definition of a canonical B-bounded linearization for finite $\exists B$-bounded MSCs, definable in first-order logic. We choose one adapted from [86, Definition 13], where the definition was given for traces. It is based on the following lemma. Though it is stated for a special case in [86], the proof can be taken almost verbatim. We only provide a proof for completeness.

Lemma 6.12 ([86, Lemma 14])**.** *Let $(E, \leq)$ be a partially ordered set, and $\sqsubseteq \subseteq E \times E$ a well-founded linear order. For $e, f \in E$, we write $e \parallel f$ when $e \not\leq f$ and $f \not\leq e$, and we let $\uparrow e = \{f \in E \mid e \leq f\}$. Then the relation $\preceq \subseteq E \times E$ defined by*

$$e \preceq f \iff \begin{pmatrix} & e \leq f & & \\ \vee & e \parallel f & \wedge & \min_{\sqsubseteq}(\uparrow e \setminus \uparrow f) \sqsubset \min_{\sqsubseteq}(\uparrow f \setminus \uparrow e) \end{pmatrix}$$

is a linear order extending $\leq$.

Proof. The relation $\preceq$ is clearly reflexive, since $\leq$ is. Notice also that, for all $e \neq f$, we have either $e \prec f$ or $f \prec e$, but not both. Indeed, we have $\uparrow e \setminus \uparrow f \cap \uparrow f \setminus \uparrow e = \emptyset$, and if $e \parallel f$, then the two sets are nonempty as $e \in \uparrow e \setminus \uparrow f$ and $f \in \uparrow f \setminus \uparrow e$.

It remains to show that $\prec$ is transitive. Let $e_1, e_2, e_3 \in E$ such that $e_1 \prec e_2 \prec e_3$. Note that e_1, e_2, e_3 are pairwise distinct. For distinct $i, j \in \{1, 2, 3\}$, if $\uparrow e_i \setminus \uparrow e_j \neq \emptyset$, we let $e_{ij} = \min_{\sqsubseteq} \uparrow e_i \setminus \uparrow e_j$.

To prove $e_1 \prec e_3$, we distinguish several cases.

Case $e_1 < e_2 < e_3$: As $<$ is transitive, we get $e_1 < e_3$.

Case $e_1 < e_2 \parallel e_3$: This implies $e_3 \not\leq e_1$. If $e_1 < e_3$, we are done. So suppose $e_1 \parallel e_3$. Since $e_2 \prec e_3$, we have $e_{23} \sqsubset e_{32}$. From $\uparrow e_2 \subseteq \uparrow e_1$, we deduce $\uparrow e_2 \setminus \uparrow e_3 \subseteq \uparrow e_1 \setminus \uparrow e_3$. Thus, $e_{13} \sqsubseteq e_{23}$. Similarly, $e_{32} \sqsubseteq e_{31}$. We obtain $e_{13} \sqsubseteq e_{23} \sqsubset e_{32} \sqsubseteq e_{31}$.

Case $e_1 \parallel e_2 < e_3$: This case is very similar to the previous one.

Case $e_1 \parallel e_2 \parallel e_3$: Since $e_1 \prec e_2 \prec e_3$, we have $e_{12} \sqsubset e_{21}$ and $e_{23} \sqsubset e_{32}$. Suppose $e_1 \not\leq e_3$ (otherwise, we are done). We have $e_3 \not\leq e_1$, since $e_3 < e_1$ implies $e_{32} \sqsubseteq e_{12} \sqsubset e_{21} \sqsubseteq e_{23}$, a contradiction.

So we can assume $e_1 \parallel e_3$. It remains to show $e_{13} \sqsubset e_{31}$. First, one shows that

$$e_{13} \sqsubseteq e_{12} \,. \tag{6.1}$$

If $e_{12} \notin \uparrow e_3$, then (6.1) is immediate. So suppose $e_{12} \in \uparrow e_3$, i.e., $e_{12} \in \uparrow e_3 \setminus \uparrow e_2$. Then,

$$e_{23} \sqsubset e_{32} \sqsubseteq e_{12} \,. \tag{6.2}$$

Let us consider two cases. If $e_{23} \notin \uparrow e_1$ then $e_{21} \sqsubseteq e_{23}$. By (6.2), we obtain $e_{21} \sqsubset e_{12}$, which contradicts $e_1 \prec e_2$. Hence $e_{23} \in \uparrow e_1$ and we get $e_{13} \sqsubseteq e_{23}$. We deduce that (6.1) holds.

To conclude the proof, we distinguish once more two cases:

Case $e_{31} \in \uparrow e_2$: Then, $e_{12} \sqsubset e_{21} \sqsubseteq e_{31}$. Applying (6.1), we obtain $e_{13} \sqsubset e_{31}$.

Case $e_{31} \notin \uparrow e_2$: Then, $e_{23} \sqsubset e_{32} \sqsubseteq e_{31}$. If $e_{23} \in \uparrow e_1$, then $e_{13} \sqsubseteq e_{23} \sqsubset e_{31}$ and we are done. If $e_{23} \notin \uparrow e_1$, then $e_{21} \sqsubseteq e_{23}$, which implies $e_{12} \sqsubset e_{31}$. By (6.1), we obtain $e_{13} \sqsubset e_{31}$.

This concludes the proof of Lemma 6.12. □

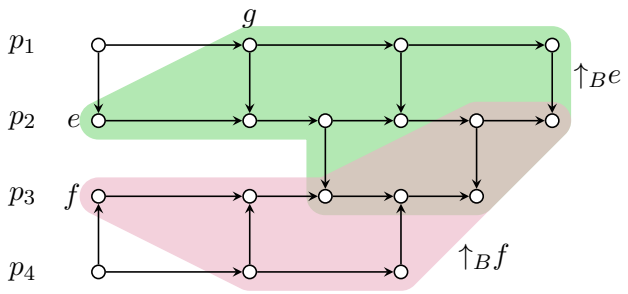


Figure 6.2: The sets $\uparrow_B e$ and $\uparrow_B f$

We can now associate with each $\exists B$-bounded MSC $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop})$ a canonical linear order $\preceq_B$ extending $\leq$. We will see that if M is finite, $\preceq_B$ is a B-bounded linearization of M, but if M is infinite, $\preceq_B$ might be of order type greater than ω, and therefore not a linearization. The total order $\preceq_B$ is defined by applying Lemma 6.12 with a linear extension $\sqsubseteq$ of $\leq_{\mathsf{proc}}$ defined as in Chapter 5: We fix some linear order $\sqsubseteq_{\mathsf{Procs}}$ on Procs, and let

$$e \sqsubseteq f \qquad \text{if} \qquad e \leq_{\mathsf{proc}} f \;\text{ or }\; loc(e) \sqsubset_{\mathsf{Procs}} loc(f)\,.$$

$\leq_B$ Define also

$$\leq_B = (< \cup\, rev_B)^*\,.$$

Recall that, by Lemma 6.9, if M is $\exists B$-bounded, then $\leq_B$ is a partial order. Therefore, the following is well-defined:

$\preceq_B$ **Definition 6.13.** Given $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$, we denote by $\preceq_B$ the linear order obtained by applying Lemma 6.12 to $(E, \leq_B)$ and $\sqsubseteq$.

Example 6.14. Consider the MSC M in Figure 6.2 and suppose $p_1 \sqsubset_{\mathsf{Procs}} p_2 \sqsubset_{\mathsf{Procs}} p_3 \sqsubset_{\mathsf{Procs}} p_4$. We have $\min_{\sqsubset}(\uparrow e \setminus \uparrow f) = g$ and $\min_{\sqsubset}(\uparrow f \setminus \uparrow e) = f$. Since $g \sqsubset f$, we obtain $e \prec_B f$.

In general, $\preceq_B$ need not be of order type at most ω, i.e., it is not necessarily a linearization. For instance, with two processes $p \sqsubset_{\mathsf{Procs}} q$, infinitely many local events on both p and q, and no communication between p and q, the order type of $\preceq_B$ is $\omega + \omega$.

On the other hand, when $\preceq_B$ *is* a linearization (i.e., is of order type at most ω), then it is a B-bounded one since it contains $\leq_B$. This applies in particular to all finite MSCs.

Lemma 6.15. *For all $M \in \mathbb{MSC}^{\mathrm{fin}}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$, $\preceq_B$ is a B-bounded linearization of M.*

An important observation is that the canonical linear order $\preceq_B$ associated with an $\exists B$-bounded MSC is definable in FO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$] (independently of whether the MSC is finite or infinite).

Lemma 6.16. *There exists a formula $x \preceq_B y \in \mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ such that for all $\exists B$-bounded MSC $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$ and events $e, f \in E$, we have*

$$M, [x \mapsto e, y \mapsto f] \models x \preceq_B y \qquad \textit{if and only if} \qquad e \preceq_B f\,.$$

Proof. Recall that if there is a path from event e to event f using relations $<$ and rev_B, there is one of length at most $2|\mathsf{Procs}|$. So we can define the relation $\leq_B$ in FO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$] as follows:

$$x \leq_B y := \bigvee_{0 \leq n \leq 2|\mathsf{Procs}|} \exists x_0, \ldots, x_n.\, x_0 = x \wedge x_n = y \wedge \bigwedge_{0 \leq i < n} x_i < x_{i+1} \vee rev_B(x_i, x_{i+1})\,.$$

We also let

$$x \sqsubset y := x <_{\mathsf{proc}} y \vee \bigvee_{\substack{p,q \in \mathsf{Procs} \\ p \sqsubset_{\mathsf{Procs}} q}} p(x) \wedge q(y)\,.$$

The condition $\min_{\sqsubseteq}(\uparrow x \backslash \uparrow y) \sqsubset \min_{\sqsubseteq}(\uparrow y \backslash \uparrow x)$ can then be expressed with the formula

$$\Phi(x, y) := \exists z. x \leq_B z \wedge \neg(y \leq_B z) \wedge \forall z'. (y \leq_B z' \wedge \neg(x \leq_B z')) \implies z \sqsubset z'\,.$$

Finally, following the definition of $\preceq_B$, we let

$$x \preceq_B y = x \leq_B y \vee (\neg(x \leq_B y) \wedge \neg(y \leq_B x) \wedge \Phi(x, y))\,. \qquad \square$$

6.4.4 Logic for linearizations

Theorem 6.8 establishes a relation between (E)MSO definability of languages of linearizations (i.e., words over A_{lin}), and of languages of MSCs. Before moving on to the proof of Theorem 6.8, let us fix some notations regarding MSO for linearizations.

Let Prop_{type} denote the extension of Prop with the set of possible event types:

$$\mathsf{Prop}_{type} = \mathsf{Prop} \cup \mathsf{Procs} \cup \{p!q \mid (p, q) \in Ch\} \cup \{p?q \mid (p, q) \in Ch\}\,.$$

We assume that $\mathsf{Prop} \cap (\mathsf{Procs} \cup \{p!q \mid (p, q) \in Ch\} \cup \{p?q \mid (p, q) \in Ch\}) = \emptyset$.

We can see words $w \in A^*_{lin} \cup A^\omega_{lin}$ as $\big(\mathsf{Prop}_{type}, \{\preceq\}\big)$-structures in which $\preceq$ is interpreted as a linear order of order type at most ω, and every element is in the interpretation of exactly one of the propositions of the form p, $p!q$ or $p?q$. Recall that the standard MSO logic MSO[$\mathsf{Prop}_{type}, \preceq$] over this signature has the following syntax:

MSO[$\mathsf{Prop}_{type}, \preceq$]

$$\Phi ::= P(x) \mid p(x) \mid (p!q)(x) \mid (p?q)(x) \mid x \preceq y \mid$$
$$x = y \mid x \in X \mid \Phi \vee \Phi \mid \neg\Phi \mid \exists x.\Phi \mid \exists X.\Phi\,,$$

where $P \in \mathsf{Prop}$, $p \in \mathsf{Procs}$, and $(p,q) \in Ch$.

For a sentence $\Phi \in \mathrm{MSO}[\mathsf{Prop}_{type}, \preceq]$ and $B \in \mathbb{N}$, we denote by $\mathbb{L}_B(\Phi)$ the set of all words $w \in A_{lin}^* \cup A_{lin}^\omega$ that correspond to a B-bounded linearization of some MSC, and satisfy Φ:

$$\mathbb{L}_B(\Phi) = \{w \in Lin_B(\mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})) \mid w \models \Phi\}\,.$$

We also write

$$\mathbb{L}_B^{\mathrm{fin}}(\Phi) = \{w \in Lin_B(\mathbb{MSC}_{\exists B}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop})) \mid w \models \Phi\} = \mathbb{L}_B(\Phi) \cap A_{lin}^*$$
$$\mathbb{L}_B^{\omega}(\Phi) = \{w \in Lin_B(\mathbb{MSC}_{\exists B}^{\omega}(\mathsf{Procs}, \mathsf{Prop})) \mid w \models \Phi\} = \mathbb{L}_B(\Phi) \cap A_{lin}^\omega\,.$$

Let us recall two known results:

Proposition 6.17. *A language $L \subseteq Lin_B(\mathbb{MSC}_{\exists B}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop}))$ is regular if and only if there exists $\Phi \in \mathrm{MSO}[\mathsf{Prop}_{type}, \preceq]$ such that $L = \mathbb{L}_B^{\mathrm{fin}}(\Phi)$.*

A language $L \subseteq Lin_B(\mathbb{MSC}_{\exists B}^{\omega}(\mathsf{Procs}, \mathsf{Prop}))$ is ω-regular if and only if there exists $\Phi \in \mathrm{MSO}[\mathsf{Prop}_{type}, \preceq]$ such that $L = \mathbb{L}_B^{\omega}(\Phi)$.

Proof. Suppose $L \subseteq Lin_B(\mathbb{MSC}_{\exists B}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop}))$ (the infinite case is similar). If L is regular, then according to Büchi-Elgot-Trakhtenbrot theorem, there is a sentence $\Phi \in \mathrm{MSO}[\mathsf{Prop}_{type}, \preceq]$ such that $L = \{w \in A_{lin}^* \mid w \models \Phi\}$. Since L contains only words in $Lin_B(\mathbb{MSC}_{\exists B}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop}))$, we also have $\mathbb{L}_B^{\mathrm{fin}}(\Phi) = L$.

Conversely, if $L = \mathbb{L}_B^{\mathrm{fin}}(\Phi)$ for some $\Phi \in \mathrm{MSO}[\mathsf{Prop}_{type}, \preceq]$, then it is the intersection of $\{w \in A_{lin}^* \mid w \models \Phi\}$, which is regular, with $Lin_B(\mathbb{MSC}_{\exists B}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop}))$. The set $Lin_B(\mathbb{MSC}_{\exists B}^{\mathrm{fin}}(\mathsf{Procs}, \mathsf{Prop}))$ is also regular: it can be recognized by an automaton that remembers the number of pending messages (between 0 and B) in every channel, and checks that every read and write can be performed. Therefore, L is regular. □

Proposition 6.18. *Let $\Phi \in \mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ be a formula over MSCs. There exists a formula $\widetilde{\Phi} \in \mathrm{MSO}[\mathsf{Prop}_{type}, \preceq]$, with $\mathrm{Free}(\widetilde{\Phi}) = \mathrm{Free}(\Phi)$, such that for all $M \in \mathbb{MSC}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$ and all interpretations ν of the free variables of Φ in M, for all $w \in Lin_B(M)$, we have*

$$M, \nu \models \Phi \quad \textit{if and only if} \quad w, \widetilde{\nu} \models \widetilde{\Phi}\,,$$

where $\widetilde{\nu}$ is the interpretation that maps each free variable x (resp. X) of $\widetilde{\Phi}$ to the position (resp. set of positions) in w corresponding to the event $\nu(x)$ (resp. set of events $\nu(X)$) in M.

Proof. We prove this for $\Phi \in \mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd]$, the predicates $\leq_{\mathsf{proc}}$ and $\leq$ being inter-definable. We construct $\widetilde{\Phi}$ by induction on Φ. The inductive cases are straightforward. We let

$$\widetilde{P(x)} = P(x) \quad \text{and} \quad \widetilde{p(x)} = p(x) \vee \bigvee_{(p,q)\in Ch} (p!q)(x) \vee (p?q)(x)\,.$$

We can then define the translation of $x \leq_{\mathsf{proc}} y$ as

$$x \preceq y \wedge \bigvee_{p\in\mathsf{Procs}} \widetilde{p(x)} \wedge \widetilde{p(y)}\,.$$

Finally, the translation of $x \lhd y$ states that, for some channel $(p,q) \in Ch$, (i) x is a write on channel (p,q), (ii) y is a receive from channel (p,q), (iii) between x and y, there are less than B messages sent, and read, on channel (p,q), (iv) the count modulo B of messages sent on channel (p,q) up until x, and of messages read from channel (p,q) up until y, are equal. □

6.4.5 A new proof of Theorem 6.8

The FO definitions of finite $\exists B$-bounded MSCs and their canonical linearizations are all we need to prove direction (3) $\implies$ (1) of Theorem 6.8, which is the difficult part of the proof. We recall here the theorem:

Theorem 6.8 ([34, Theorem 4.1])**.** *Let $B \in \mathbb{N}$ and $L \subseteq \mathbb{MSC}^{\mathrm{fin}}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$. The following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$.*
2. *$L = \mathbb{L}(\Phi)$ for some $\mathrm{MSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ formula Φ.*
3. *$Lin_B(L)$ is a regular language (of finite words).*

Proof. The translation (1) $\implies$ (2) is standard [13]: the formula guesses an assignment of transitions to events in terms of existentially quantified second-order variables (one for each transition) and then checks, in its first-order kernel, that the assignment is indeed an accepting run.

Implication (2) $\implies$ (3) follows from Proposition 6.18: if $L = \mathbb{L}(\Phi)$, then $Lin_B(L) = \mathbb{L}^{\mathrm{fin}}_B(\widetilde{\Phi})$. By Proposition 6.17, $Lin_B(L)$ is regular.

Let us now prove that (3) $\implies$ (1). Let L be a set of finite $\exists B$-bounded MSCs such that $Lin_B(L)$ is regular. There exists an $\mathrm{EMSO}[\mathsf{Prop}_{type}, \preceq]$ sentence Φ_{lin} such that $Lin_B(L) = \mathbb{L}^{\mathrm{fin}}_B(\Phi_{lin})$. Using the formula from Lemma 6.16, we can translate Φ_{lin} into an $\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ formula Φ such that, for all finite $\exists B$-bounded MSCs M, we have $M \models \Phi$ if and only if $M_{\preceq_B} \models \Phi_{lin}$. Recall also that there exists

an FO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$] formula $\Phi^{\text{fin}}_{\exists B}$ characterizing the set of finite $\exists B$-bounded MSCs. Applying Proposition 6.2, we obtain a CFM $\mathcal{A}$ such that $\mathbb{L}(\mathcal{A}) = \mathbb{L}(\Phi \wedge \Phi^{\text{fin}}_{\exists B})$. Then, for all $M \in L$, M is $\exists B$-bounded and $M_{\preceq_B} \models \Phi_{lin}$, hence $M \models \Phi \wedge \Phi^{\text{fin}}_{\exists B}$, i.e., $M \in \mathbb{L}(\mathcal{A})$. Conversely, if $M \in \mathbb{L}(\mathcal{A})$, then M is $\exists B$-bounded and $\preceq_B$ is a linearization of M. Moreover, $M_{\preceq_B} \in Lin_B(L)$, hence $M \in L$. □

6.4.6 Extension to infinite MSCs

Our proof of Theorem 6.8 can be adapted to infinite MSCs. However, there is one major difference: the canonical linear order $\preceq_B$ associated with an infinite $\exists B$-bounded MSC is not necessarily of order type ω, which means that for $M \in L$, it does not directly define a linearization $M_{\preceq_B} \in Lin_B(L)$, contrarily to the finite case. In fact, as observed in [56], there are no canonical FO-definable linearizations for infinite MSCs. In other settings (Mazurkiewicz traces in [86], universally bounded MSCs in [56]), this issue has been resolved by decomposing infinite behaviors into one finite part, and boundedly many infinite disconnected parts whose shape guarantees that the FO-definable linear order $\preceq_B$ is of order type at most ω in each part. We follow the same approach, our aim being to prove the following extension of Theorem 6.8 to infinite $\exists B$-bounded MSCs:

Theorem 6.19. *Let $B \in \mathbb{N}$ and $L \subseteq \mathbb{MSC}^{\omega}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$. The following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$.*
2. *$L = \mathbb{L}(\Phi)$ for some* MSO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$] *formula Φ.*
3. *$Lin_B(L)$ is ω-regular.*

The proofs of directions (1) $\implies$ (2) and (2) $\implies$ (3) are as in the finite case (Theorem 6.8), so we focus on (3) $\implies$ (1). The outline of the proof is as follows. We first define a decomposition of $\exists B$-bounded MSCs into one finite part M^{fin}, which contains all types of actions that occur only finitely many times, and boundedly many infinite parts $M^1, \ldots, M^n$, defined as the connected components of the remainder of the MSC. We show that in each M^j, $\preceq_B$ is of order type ω, i.e., we can define canonical linearizations for each part of the decomposition. The decomposition of an MSC M into $M^{\text{fin}}, M^1, \ldots, M^n$ induces a decomposition of any of its linearizations $w \in Lin_B(M)$ into subwords w^{fin}, $w^1, \ldots, w^n$. We prove that if $Lin_B(L)$ is ω-regular, we can find a finite family of tuples of regular languages $(K, L^1, \ldots, L^n)$ such that $w \in Lin_B(L)$ if and only if $w^{\text{fin}} \in K, w^1 \in L^1, \ldots, w^n \in L^n$ for one of the tuples $(K, L^1, \ldots, L^n)$. Each language in these tuples can be defined by an EMSO[$\mathsf{Prop}_{type}, \preceq$] formula over linearizations. Since each language talks only about one of the components $M^{\text{fin}}, M^1, \ldots, M^n$, and since we have FO-definable canonical linearizations for all these components, we can translate the

EMSO[$\mathsf{Prop}_{type}, \preceq$] formulas over words into formulas over MSCs. Finally, we apply again Theorem 6.3 to obtain a CFM for L.

Notations. We use standard notations from formal language theory. The empty word is denoted by ε. Given $L \subseteq A_{lin}^{\omega}$, $u \in A_{lin}^{*}$, and $K \subseteq A_{lin}^{*}$, we write $u^{-1}L = \{v \in A_{lin}^{\omega} \mid uv \in L\}$ and $K^{-1}L = \{v \in L \mid \exists u \in K, uv \in L\}$. For words $u, v \in A_{lin}^{*} \cup A_{lin}^{\omega}$, we denote by $u \sqcup\!\sqcup v$ the *shuffle* of u and v, i.e., the set of words $w \in A_{lin}^{*} \cup A_{lin}^{\omega}$ that are possible interleavings of u and v. This extends to languages $K, L \subseteq A_{lin}^{*} \cup A_{lin}^{\omega}$ with $K \sqcup\!\sqcup L = \bigcup_{u \in K, v \in L} u \sqcup\!\sqcup v$.

MSC decompositions. The first step of the decomposition consists in dividing MSCs $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}_{\exists B}^{\omega}(\mathsf{Procs}, \mathsf{Prop})$ into one downward-closed finite part M^{fin} containing all the action types that occur only finitely many times in M, and one infinite part M^{inf} in which all actions are repeated infinitely often. We let

$$\begin{aligned} \mathit{Types}^{\text{fin}}(M) &= \{\mathit{type}(e) \mid e \in E \wedge \{f \in E \mid \mathit{type}(e) = \mathit{type}(f)\} \text{ is finite}\} \\ \mathit{Types}^{\text{inf}}(M) &= \{\mathit{type}(e) \mid e \in E \wedge \{f \in E \mid \mathit{type}(e) = \mathit{type}(f)\} \text{ is infinite}\}\,. \end{aligned}$$

We define E^{fin} as the downward-closure, with respect to $\leq_B$, of all events whose type is in $\mathit{Types}^{\text{fin}}(M)$, and E^{inf} as the set of remaining events:

$$\begin{aligned} E^{\text{fin}} &= \{e \in E \mid \exists f \in E, \mathit{type}(f) \in \mathit{Types}^{\text{fin}}(M) \wedge e \leq_B f\} \\ E^{\text{inf}} &= E \setminus E^{\text{fin}}\,. \end{aligned}$$

The MSC M can then be split into one MSC M^{fin}, and one MSC M^{inf}, defined respectively as the restriction of M to events in E^{fin} or E^{inf}. Note that E^{fin} and E^{inf} are not necessarily $\lhd$-closed: there may be some send events in E^{fin} whose matching receive events are in E^{inf} (the converse is not possible, since E^{fin} is downward closed). So some message edges are present neither in M^{fin} nor in M^{inf}, and the corresponding events become internal events rather than send or receive events. In order to not lose any information in the decomposition of M, we add to the labeling of M^{fin} and M^{inf} the (former) type of every event. More formally, we define the MSCs $M^{\text{fin}}, M^{\text{inf}} \in \mathbb{MSC}(\mathsf{Procs}, \mathsf{Prop}_{type})$ as follows:

$$\begin{aligned} M^{\text{fin}} &= \left(E^{\text{fin}}, \rightarrow \cap\, E^{\text{fin}} \times E^{\text{fin}}, \lhd \cap\, E^{\text{fin}} \times E^{\text{fin}}, loc|_{E^{\text{fin}}}, (\lambda \cup \mathit{type})|_{E^{\text{fin}}}\right) \\ M^{\text{inf}} &= \left(E^{\text{inf}}, \rightarrow \cap\, E^{\text{inf}} \times E^{\text{inf}}, \lhd \cap\, E^{\text{inf}} \times E^{\text{inf}}, loc|_{E^{\text{inf}}}, (\lambda \cup \mathit{type})|_{E^{\text{inf}}}\right)\,, \end{aligned}$$

where $\lambda \cup \mathit{type}$ denotes the function which maps e to $\lambda(e) \cup \{\mathit{type}(e)\}$.

In a second step, we further decompose M^{inf} according to its (weakly) connected components. Consider the undirected communication graph at infinity, where vertices are the processes p such that E_p is infinite, and two processes p, q are connected

by an edge if they communicate infinitely often, that is, $\mathit{Types}^{\mathrm{inf}}(M) \cap \{p!q, q!p\} \neq \emptyset$. Let $P^1, \ldots, P^n$ denote the maximal connected components of this graph. Note that the set $\{P^1, \ldots, P^n\}$ is entirely determined by $\mathit{Types}^{\mathrm{inf}}(M)$. For all $1 \leq j \leq n$, we let

$$E^j = E^{\mathrm{inf}} \cap \bigcup_{p \in P^j} E_p$$

and

$$M^j = \left(E^j, \rightarrow \cap\, E^j \times E^j, \lhd \cap\, E^j \times E^j, \mathit{loc}|_{E^j}, (\lambda \cup \mathit{type})|_{E^j}\right) .$$

The purpose of this decomposition is that, contrarily to arbitrary infinite MSCs, all parts in the decomposition have canonical FO-definable linearizations. This is what will allow us to translate formulas over word linearizations into formulas over MSCs, as we did in the proof of Theorem 6.8.

Lemma 6.20. *For all M^j with $1 \leq j \leq n$, the linear order $\preceq_B$ is of order type ω.*

Proof. This is in fact true of any linear extension of $\leq_B$, and not just the canonical one $\preceq_B$. We want to prove that for every $e \in E^j$, the set $\{f \in E^j \mid f \preceq_B e\}$ is finite. To do so, it suffices to prove that for all $p \in P^j$, there exists $f \in E^j_p$ such that $e \leq_B f$ (and thus $e \preceq_B f$). By definition of P^j, there exists a path $p_1, \ldots, p_\ell$ such that $p_1 = \mathit{loc}(e)$, $p_\ell = p$, and for all $1 \leq i < \ell$, M^j contains infinitely many events of type $p_i!p_{i+1}$ or $p_i?p_{i+1}$. If M^j contains infinitely many events of type $p_i!p_{i+1}$, then for all e' on process p_i, there exist f' on process p_i and g' on process p_{i+1} such that $e' \leq_{\mathsf{proc}} f'$ and $f' \lhd g'$, hence $e' \leq_B g'$. If M^j contains infinitely many events of type $p_i?p_{i+1}$, then for all e' on process p_i, there exist f' on process p_i and g' on process p_{i+1} such that $e' \leq_{\mathsf{proc}} f'$ and f' rev_B g', hence $e' \leq_B g'$. Therefore, we obtain events $e_2, \ldots, e_\ell$ on processes $p_2, \ldots, p_\ell = p$ such that $e \leq_B e_2 \leq_B \cdots \leq_B e_\ell$. □

Decomposition of linearizations. The decomposition of M into $M^{\mathrm{fin}}, M^1, \ldots, M^n$ induces a decomposition of every linearization $w \in \mathit{Lin}_B(M) \subseteq A^\omega_{lin}$. We define subalphabets $A^{\mathrm{fin}}, A^{\mathrm{inf}}, A^1, \ldots, A^n \subseteq A_{lin}$ as follows:

$$\begin{aligned} A^{\mathrm{fin}} &= 2^{\mathsf{Prop}} \times \mathit{Types}^{\mathrm{fin}}(M) \\ A^{\mathrm{inf}} &= 2^{\mathsf{Prop}} \times \mathit{Types}^{\mathrm{inf}}(M) \\ A^j &= 2^{\mathsf{Prop}} \times \left(\mathit{Types}^{\mathrm{inf}}(M) \cap \left\{p, p!q, p?q \mid p, q \in P^j\right\}\right) . \end{aligned}$$

For every linearization $w \in \mathit{Lin}_B(M)$, we denote by

$$w^{\mathrm{fin}} \in A^*_{lin} A^{\mathrm{fin}} \cup \{\varepsilon\} , \qquad w^{\mathrm{inf}} \in \left(A^{\mathrm{inf}}\right)^\omega , \qquad w^j \in \left(A^j\right)^\omega \ (\text{for } 1 \leq j \leq n)$$

the scattered subwords of w induced by the set of positions corresponding respectively to events in M^{fin}, M^{inf}, or M^j. Notice that $w \in w^{\mathrm{fin}} \sqcup\!\sqcup w^{\mathrm{inf}}$, and $w^{\mathrm{inf}} \in \coprod_{1 \leq j \leq n} w^j$.

In general, it is not necessary that $w = w^{\text{fin}} \cdot w^{\text{inf}}$. However, $w^{\text{fin}} \cdot w^{\text{inf}}$ is always also a valid linearization of M. More generally, we can always construct a B-bounded linearization of M from two B-bounded linearizations of M^{fin} and M^{inf}:

Lemma 6.21. *For all $u, v \in \mathit{Lin}_B(M)$, we have $u^{\text{fin}}v^{\text{inf}} \in \mathit{Lin}_B(M)$.*

Proof. Let $\preceq_u$ and $\preceq_v$ be the B-bounded linearizations of M associated respectively with u and v, and let $\preceq$ be the concatenation of their restrictions to E^{fin} and E^{inf}, respectively. That is, $e \preceq f$ if $e \in E^{\text{fin}}$ and $f \in E^{\text{inf}}$, or $e \preceq_u f$ and $e, f \in E^{\text{fin}}$, or $e \preceq_v f$ and $e, f \in E^{\text{inf}}$. Clearly, $\preceq$ is a total order on E, and of order type ω (like $\preceq_v$). Let us show that for all $e \leq_B f$, we have $e \preceq f$. This implies that $\preceq$ contains $\leq$, i.e., is a linearization of M, and that it contains rev_B, i.e., it is B-bounded (by Lemma 6.9). We can first observe that for all $e \leq_B f$, we have $e \preceq_u f$ and $e \preceq_v f$ since $\preceq_u$ and $\preceq_v$ are B-bounded. Therefore, if e and f are both in E^{fin} or both in E^{inf}, then $e \preceq f$. If $e \in E^{\text{fin}}$ and $f \in E^{\text{inf}}$, we have $e \preceq f$ by definition. Finally, we cannot have $e \in E^{\text{inf}}$ and $f \in E^{\text{fin}}$ since E^{fin} is downward-closed with respect to $\leq_B$. We conclude that $\preceq$ is a B-bounded linearization of M. The corresponding word $w \in \mathit{Lin}_B(L)$ is precisely $w = u^{\text{fin}}v^{\text{inf}}$. □

Another important observation is that, since there are no causal dependencies between events in distinct infinite parts M^j, any interleaving of a set of B-bounded linearizations for each M^j yields a B-bounded linearization of M^{inf}. In terms of $\mathit{Lin}_B(M)$, this is reflected as follows:

Lemma 6.22. *For all $w \in \mathit{Lin}_B(M)$, we have $w^{\text{fin}} \cdot \bigsqcup_{j=1}^{n} w^j \subseteq \mathit{Lin}_B(M)$.*

Proof. By Lemma 6.21, we have $w^{\text{fin}}w^{\text{inf}} \in \mathit{Lin}_B(M)$. Moreover, every w^j corresponds to a B-bounded linearization of M^j. As observed above, any interleaving of the linearizations $w^1, \ldots, w^n$, that is, every word $v \in \bigsqcup_{j=1}^{n} w^j$, thus corresponds to a linearization of M^{inf}. Therefore, we also have $w^{\text{fin}}v \in \mathit{Lin}_B(M)$. □

Main proof. We are now ready to start the proof of Theorem 6.19.

Proof of Theorem 6.19. We prove (3) $\implies$ (1), the proof for the other directions is as in Theorem 6.8.

Let $L \subseteq \mathbb{MSC}^{\omega}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$ be a set of MSCs such that $\mathit{Lin}_B(L)$ is ω-regular. Without loss of generality, we can assume that all MSCs $M \in L$ have the same set $T = \mathit{Types}^{\text{inf}}(M)$ of types at infinity. Indeed, properties (3) and (1) hold for L if and only if they hold for every restriction of L to a particular set T of types at infinity. Fixing this set T allows us to uniformly use the notations we introduced in the definition of MSC decompositions ($P^1, \ldots, P^n$, etc.), for all MSCs $M \in L$.

We first show that $\mathit{Lin}_B(L)$ can be decomposed into regular languages describing partial linearizations w^{fin} or w^{inf}:

Lemma 6.23. *There is a finite sequence of pairs of regular languages $(K_i, L_i)_{1\leq i\leq k}$, with $K_i \subseteq A^*_{lin}$ and $L_i \subseteq A^\omega_{lin}$, such that*

$$\Big\{\big(w^{\mathrm{fin}}, w^{\mathrm{inf}}\big) \mid w \in Lin_B(L)\Big\} = \bigcup_{1\leq i\leq k} K_i \times L_i\,.$$

Moreover, the languages L_i are closed under permutation of letters in distinct subalphabets A^j: for all $1 \leq i \leq k$ and tuples $(v_1, \ldots, v_n) \in \big(A^1\big)^\omega \times \cdots \times (A^n)^\omega$, if $L_i \cap \coprod_{j=1}^n v_j \neq \emptyset$ then $\coprod_{j=1}^n v_j \subseteq L_i$.

Proof. Observe that the set E^{fin} is definable in MSO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$]. Using the translation of Proposition 6.18 from MSO[$\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd$] formulas over $\exists B$-bounded MSCs to MSO[$\mathsf{Prop}_{type}, \preceq$] formulas over B-bounded linearizations, we can define an MSO[$\mathsf{Prop}_{type}, \preceq$] formula $\widetilde{fin}(x)$ such that for all $w \in Lin_B(L)$, the formula $\widetilde{fin}(x)$ holds precisely at positions corresponding to events in E^{fin}.

Let $K \subseteq A^*_{lin}$ be the regular language defined by the formula $\forall x.\widetilde{fin}(x)$. Notice that K contains all finite parts w^{fin} of words $w \in Lin_B(L)$.

Let $\sim \subseteq K \times K$ be the equivalence relation defined by $u \sim v$ if $u^{-1}Lin_B(L) = v^{-1}Lin_B(L)$. Since $Lin_B(L)$ is regular, $\sim$ has finitely many equivalence classes $K_1, \ldots, K_k \subseteq K$. For all $1 \leq i \leq k$, we let $L_i = \big(K_i^{-1}Lin_B(L)\big) \cap \big(A^{\mathrm{inf}}\big)^\omega$. Let us show that

$$\Big\{\big(w^{\mathrm{fin}}, w^{\mathrm{inf}}\big) \mid w \in Lin_B(L)\Big\} = \bigcup_{1\leq i\leq k} K_i \times L_i\,.$$

Let $w \in Lin_B(L)$. We have $w^{\mathrm{fin}} \in K$, so there exists i such that $w^{\mathrm{fin}} \in K_i$. Moreover, by Lemma 6.21, $w^{\mathrm{fin}}w^{\mathrm{inf}} \in Lin_B(L)$. Thus, $w^{\mathrm{inf}} \in K_i^{-1}Lin_B(L) \cap \big(A^{\mathrm{inf}}\big)^\omega$, and $(w^{\mathrm{fin}}, w^{\mathrm{inf}}) \in K_i \times L_i$.

Conversely, if $(u, v) \in K_i \times L_i$, then $u'v \in Lin_B(L)$ for some $u' \in K_i$. Since $u \sim u'$, we also have $w = uv \in Lin_B(L)$. In addition, v contains only letters from A^{inf}, so u contains all positions from w^{fin}. Since u is in K, it also contains only positions from w^{fin}, that is, $u = w^{\mathrm{fin}}$. It follows that $v = w^{\mathrm{inf}}$.

Finally, we prove that each L_i is closed under commutation of letters in distinct subalphabets A^j. Let $(v_1, \ldots, v_n) \in \big(A^1\big)^\omega \times \cdots \times (A^n)^\omega$ such that there exists some $v \in L_i \cap \coprod_{j=1}^n v_j$. Since $v \in L_i$, there exist $u \in K_i$ and $w \in Lin_B(L)$ such that $u = w^{\mathrm{fin}}$ and $v = w^{\mathrm{inf}}$. Since the alphabets A^j are disjoint, this implies $v_j = w^j$ for all j. By Lemma 6.22, this means that $u \cdot \coprod_{j=1}^n v_j \subseteq Lin_B(L)$, hence, $\coprod_{j=1}^n v_j \subseteq u^{-1}Lin_B(L) \cap \big(A^{\mathrm{inf}}\big)^\omega = L_i$. □

To further decompose each L_i according to the partition of w^{inf} into $w^1, \ldots, w^n$, we apply the lemma below, proven e.g. in [56, Theorem 4.11].

Lemma 6.24. *Let R be an ω-regular language over a finite alphabet $\Sigma = \bigcup_{i=1}^k \Sigma_i$, where $\Sigma_i \cap \Sigma_j = \emptyset$ for $i \neq j$. Suppose that all words $w \in R$ contain infinitely*

many letters from each alphabet Σ_i, and that for all $(u_1, \ldots, u_k) \in \Sigma_1^\omega \times \cdots \times \Sigma_k^\omega$, $R \cap \coprod_{i=1}^k u_i \neq \emptyset$ implies $\coprod_{i=1}^k u_i \subseteq R$. Then R is a finite union of languages of the form $\coprod_{1 \leq i \leq k} R_i$, where R_i is an ω-regular language over Σ_i.

Combining Lemmas 6.23 and 6.24, we obtain:

Lemma 6.25. *There is a finite sequence $(K'_i, L_i^1, \ldots, L_i^n)_{1 \leq i \leq \ell}$ of tuples of regular languages, with $K'_i \subseteq A_{lin}^*$ and $L_i^j \subseteq (A^j)^\omega$, such that*

$$\Big\{ \big(w^{\mathrm{fin}}, w^1, \ldots, w^n\big) \mid w \in Lin_B(L) \Big\} = \bigcup_{1 \leq i \leq \ell} K'_i \times L_i^1 \times \cdots \times L_i^n \,.$$

Proof. We apply Lemma 6.23 and then Lemma 6.24 to each L_i. We obtain a finite family $(K'_i, L_i^1, \ldots, L_i^n)_{1 \leq i \leq \ell}$ such that all $K'_i \subseteq A_{lin}^*$ are regular languages and all $L_i^j \subseteq (A^j)^\omega$ are ω-regular languages, and

$$\Big\{ \big(w^{\mathrm{fin}}, w^{\mathrm{inf}}\big) \mid w \in Lin_B(L) \Big\} = \bigcup_{1 \leq i \leq \ell} K'_i \times \coprod_{j=1}^n L_i^j \,.$$

Moreover, for all $w \in Lin_B(L)$, $(w^1, \ldots, w^n)$ is the unique decomposition of w^{inf} as a shuffle of words in the subalphabets $A^1, \ldots, A^n$, which means that $w^{\mathrm{inf}} \in \coprod_{j=1}^n L_i^j$ if and only if $(w^1, \ldots, w^n) \in L_i^1 \times \cdots \times L_i^n$. We obtain

$$\Big\{ \big(w^{\mathrm{fin}}, w^1, \ldots, w^n\big) \mid w \in Lin_B(L) \Big\} = \bigcup_{1 \leq i \leq \ell} K'_i \times L_i^1 \times \cdots \times L_i^n \,. \qquad \square$$

All languages K'_i and L_i^j in Lemma 6.25 are regular or ω-regular, which means that they can be defined by EMSO$[\mathsf{Prop}_{type}, \preceq]$ formulas $\widehat{\Phi}_i, \widehat{\Psi}_i^1, \ldots, \widehat{\Psi}_i^n$. As in the case of finite MSCs, we can interpret these formulas into the MSCs $M^{\mathrm{fin}}, M^1, \ldots, M^n$ using their canonical linearizations $\preceq_B$: we define EMSO$[\mathsf{Procs}, \mathsf{Prop}_{type}, \leq, \lhd]$ formulas $\Phi_i, \Psi_i^1, \ldots, \Psi_i^n$ by replacing in $\widehat{\Phi}_i, \widehat{\Psi}_i^1, \ldots, \widehat{\Psi}_i^n$ every predicate $x \preceq y$ with the FO$[\mathsf{Procs}, \mathsf{Prop}_{type}, \leq, \lhd]$ formula $x \preceq_B y$ defined in Lemma 6.16.

Lemma 6.26. *For all $M \in \mathbb{MSC}_{\exists B}^\omega(\mathsf{Procs}, \mathsf{Prop})$ with $Types^{\mathrm{inf}}(M) = T$, we have $M \in L$ if and only if there exists i such that $M^{\mathrm{fin}} \models \Phi_i$ and $M^j \models \Psi_i^j$ for all $1 \leq j \leq n$.*

Proof. We can define a B-bounded linearization w of M as follows. Let $u \in A_{lin}^*$ be the canonical linearization of M^{fin}, and $v_1 \in (A^1)^\omega, \ldots, v_n \in (A^n)^\omega$ the canonical linearizations of the MSCs $M^1, \ldots, M^n$ (by Lemma 6.20, this is well-defined). Let $w \in u \cdot \coprod_{j=1}^n v_j$. Then $w \in Lin_B(M)$.

We have $M \in L$ if and only if $w \in Lin_B(L)$, or equivalently, if and only if there exists i such that $u = w^{\mathrm{fin}} \models \widehat{\Phi}_i$ and $v_j = w^j \models \widehat{\Psi}_i^j$ for all j, that is, if and only if there exists i such that $M^{\mathrm{inf}} \models \Phi_i$ and $M^j \models \Psi_i^j$ for all j. $\square$

Finally, by combining formulas Φ_i and Ψ_i^j, we can construct a formula $\Phi \in \mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ such that, for all $M \in \mathbb{MSC}^{\omega}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$ such that $\mathit{Types}^{\inf}(M) = T$, we have $M \models \Phi$ if and only if there exists i such that $M^{\mathrm{fin}} \models \Phi_i$ and $M^j \models \Psi_i^j$ for all $1 \leq j \leq n$. First, there are $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \leq, \lhd]$ formulas $\mathit{fin}(x), \mathit{inf}^1(x), \ldots, \mathit{inf}^n(x)$ that hold precisely at events in $M^{\mathrm{fin}}, M^1, \ldots, M^n$, respectively. Let $\widetilde{\Phi}_i$ be the $\mathrm{EMSO}[\mathsf{Procs}, \mathsf{Prop}, \lhd, \leq]$ formula obtained from Φ_i by (i) restricting every quantification to events in M^{fin}, for instance, replacing $\exists x.\xi$ with $\exists x.\mathit{fin}(x) \wedge \xi$, and (ii) replacing $\mathsf{Prop}_{\mathit{type}}$ predicates with $\mathrm{FO}[\mathsf{Procs}, \mathsf{Prop}, \lhd, \leq]$ formulas, for instance, the formula $(p?q)(x)$ is replaced with $p(x) \wedge \exists y.(q(y) \wedge y \lhd x)$. We define similarly formulas $\widetilde{\Psi}_i^j$ relativized to M^j. We then let

$$\Phi = \bigvee_{1 \leq i \leq \ell} \left(\widetilde{\Phi}_i \wedge \bigwedge_{1 \leq j \leq n} \widetilde{\Psi}_i^j \right).$$

The language L is then the intersection of $\mathbb{L}(\Phi)$ with the set of MSCs $M \in \mathbb{MSC}^{\omega}_{\exists B}(\mathsf{Procs}, \mathsf{Prop})$ such that $\mathit{Types}^{\inf}(M) = T$. By Theorem 6.3, both of these languages are recognizable, and therefore, so is L. □

Chapter 7

Conclusion

We studied the expressive power of first-order logic and star-free PDL over ordered structures, and in the special case of MSCs, compared them to CFMs. Let us recall the two main results of the thesis:

Interval-preserving structures have the 3-variable property. In Chapter 3, we proved that, given a signature Σ with only unary and binary relation symbols (including a special symbol $\leq$), the logics $\mathrm{FO}[\Sigma]$, $\mathrm{FO}^3[\Sigma]$ and $\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ are all expressively equivalent over any class of Σ-structures in which

- $\leq$ is interpreted as a linear order, and
- binary relation symbols are interpreted as interval-preserving relations.

This applies to various classes, including linear orders, real-time signals, polynomial functions over the reals, Mazurkiewicz traces, or MSCs.

CFMs are expressively equivalent to existential monadic second-order logic. Based on successive simplifications of star-free PDL formulas over MSCs, presented in Chapter 5, we proved in Chapter 6 that any star-free PDL or FO formula over MSCs can be translated into an equivalent CFM. This shows that CFMs have the same expressive power as EMSO over MSCs.

While both results answer questions related purely to first-order logic and CFMs, star-free PDL is at the center of all our constructions. Besides being a convenient tool in the proofs, star-free PDL is also an interesting logic on its own. It combines features from classic formalisms (PDL, the calculus of relations, star-free regular expressions), and captures various temporal logics. Over MSCs, the fragment $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ of star-free PDL (without complement of path formulas, and with path constructs similar to LTL until and since modalities) has the same expressive power

as first-order logic, but the translation to CFMs is only exponential in the size of the formula, while it is non-elementary for FO.

Of course, many questions related to the expressivity of logics and automata over ordered structures remain open, offering interesting directions for further research.

7.1 The k-variable property

The fact that interval-preserving structures have the 3-variable property generalizes several known results. Besides applications to new classes of structures, this unified proof helps us understand the common features of classes for which the 3-variable property was already known. However, it fails to cover other typical examples for which the bounded variable hierarchy collapses: trees.

Immerman and Kozen [52] established k-variable properties for several classes of bounded-degree trees. Another related result is the equivalence of conditional XPath and first-order logic [64], over finite trees of unbounded degree equipped with a descendant relation and a total order on siblings. In fact, to prove that FO and conditional XPath are equivalent, Marx first showed that such trees (as well as ordered binary trees) have the 3-variable property.

Interestingly, conditional XPath is very similar to the fragment $\mathrm{PDL}_{\mathsf{sf}}^{\mathrm{MSC}}$ we defined for MSCs. (The main difference is that it does not contain a Loop operator, which would be redundant over trees.) In a way, our approach is dual to [64]: we go from FO to star-free PDL to FO^3, while [64] goes from FO to FO^3 to conditional XPath. Yet, the equivalence of star-free PDL, FO^3 and FO over MSCs can be seen as analogous to the equivalence of conditional XPath, FO^3 and FO over trees. This raises the question of whether a unified proof of both results can be found.

More generally, it would be interesting to find a more general set of sufficient conditions for a class of structures to have the k-variable property, which would apply both to interval-preserving structures and some tree orders. This is not as simple as it may seem: first, it is not completely clear what a good definition of "interval-preserving" relations would be for trees, and, secondly, the proofs in Chapter 3 rely in several places on the fact that the order is linear.

While our result provides sufficient conditions for a class of structures to have the k-variable property, it could also be interesting to find *necessary* conditions. The absence of a k-variable property can be very difficult to establish: for instance, the question of whether finite ordered graphs have the k-variable property for some k remained open for over 25 years [50, 20] before being answered in the negative by Rossman [75].

7.2 Succinctness

Over arbitrary interval-preserving structures, our translations from first-order logic to star-free PDL, and thus from FO to FO^3, may produce formulas of non-elementary size. Over MSCs, the translation from FO to CFMs is non-elementary as well.

However, we only have matching lower bounds for the translation from FO to CFMs, and the translation from PDL_{sf} to CFMs. These lower bounds are inherited from the case of words, where the translation from first-order logic, as well as from star-free regular expressions, into automata produces automata of non-elementary size [80, 72].

To the best of our knowledge, the question of whether the non-elementary blow-up can be avoided in the translation from FO to FO^3 is open. In [39], Grohe and Schweikardt prove that over words, FO^4 is at least exponentially more succinct than FO^3, but mention that what happens beyond 4 variables is unknown.

We are also missing a lower bound in the complexity of the translation from the temporal logic $TL[\mathsf{Procs}, \mathsf{Prop}, \langle\triangleleft\rangle, \langle\triangleleft^{-1}\rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ into CFMs. Recall that the number of states of the CFM we construct is exponential in the size of the formula, and doubly exponential in the number of processes. The complexity with respect to the input formula is optimal, since it is the same as for LTL. However, we do not know if the double exponential in the number of processes is necessary.

7.3 Expressive completeness of temporal logics

Star-free PDL can be seen as a two-dimensional temporal logic. In that sense, the results of Chapter 3 establish the existence of expressively complete (w.r.t. first-order logic) two-dimensional temporal logics over all classes of interval-preserving structures. For MSCs, the logic $PDL_{\mathsf{sf}}^{\mathrm{MSC}}$ is also a two-dimensional expressively complete temporal logic, and benefits from others of the properties one might look for in a temporal logic: it is somewhat closer to LTL, and the complexity of translating $PDL_{\mathsf{sf}}^{\mathrm{MSC}}$ formulas into CFMs is reasonable.

However, much remains open regarding the expressive completeness of *one*-dimensional temporal logics.

Temporal logics for MSCs. In Section 5.4, we presented a one-dimensional temporal logic $TL[\mathsf{Procs}, \mathsf{Prop}, \langle\triangleleft\rangle, \langle\triangleleft^{-1}\rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ over MSCs, based on classic temporal connectives for partial orders. We conjecture that this logic is strictly less expressive than first-order logic. This leaves us with the following open question:

> Over MSCs, does there exist a finite set of (one-dimensional) FO-definable modalities which is expressively complete w.r.t. first-order logic?

Criteria for the existence of an expressively complete temporal logic. Similarly to how the 3-variable property for interval-preserving structures generalizes several known results, it would be very interesting, though particularly challenging, to find general conditions ensuring the existence of an expressively complete (one-dimensional) temporal logic over a given class of structures.

A first variant of this question would be to find sufficient conditions for a class of structures to admit an expressively complete temporal logic consisting of a a finite set of one-dimensional FO-definable modalities. Here we are only interested in the existence of such a logic, and not the particular modalities used. Note that while having the 3-variable property ensures the existence of *two*-dimensional expressively complete temporal logics [30], this is not the case in the one-dimensional case [46]. Being interval-preserving would not be a strong enough requirement either [45].

Another approach would be to fix a generic, "natural" temporal logic (possibly with an infinite set of modalities, as is the case for MTL), and find conditions for its expressive completeness. Section 3.8 is a first step in this direction, though Loop formulas are still not very satisfactory.

7.4 Expressive power of CFMs

There are still a number of questions worth investigating regarding the expressive power of CFMs.

PDL with converse. Propositional dynamic logic is an interesting specification language, allowing intuitive definitions of many of the classic properties studied in verification, and typically associated with lower complexities than logics such as FO or MSO. The known relations between the expressive power of variants of PDL and CFMs are illustrated in Figure 7.1. For the most general variant, ICPDL[$\mathsf{Procs}, \mathsf{Prop}, \lhd, \rightarrow$] (PDL with converse and intersection), there exists a sentence which is not equivalent to any CFM [12]. However, in the fragment without intersection and complement, PDL[$\mathsf{Procs}, \mathsf{Prop}, \lhd, \rightarrow$], every sentence can be translated into an equivalent CFM [12]. We saw that such a translation is also possible for the logic $\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}$ defined in Chapter 5, which can be seen as another fragment of ICPDL[$\mathsf{Procs}, \mathsf{Prop}, \lhd, \rightarrow$].

One important question remains open: in the fragment CPDL[$\mathsf{Procs}, \mathsf{Prop}, \lhd, \rightarrow$] with converse but without intersection, can every sentence be translated into an equivalent CFM?

CFMs over restrictions of MSCs. Recall that in general, CFMs are not closed under complement [13], which means that while equivalent to EMSO, they are strictly weaker than MSO. However, under several classic restrictions of MSCs, such as universally or existentially bounded MSCs, CFMs become equivalent to full

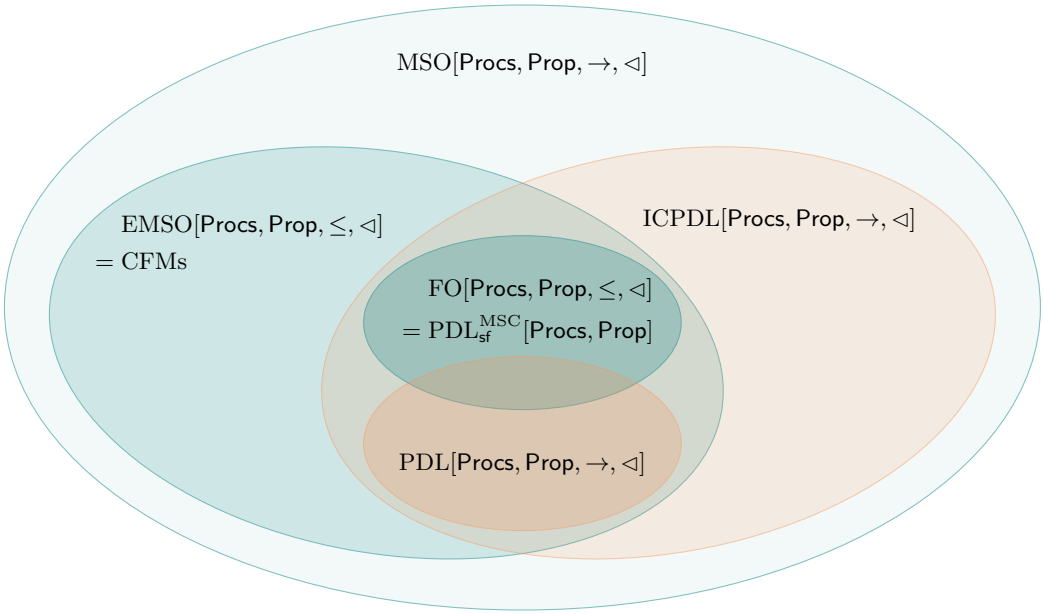


Figure 7.1: Expressive power of logic and automata over MSCs

MSO [44, 56, 34, 35]. This means that over the considered class $\mathcal{C}$ of MSCs, for every MSO sentence Φ, there exists a CFM $\mathcal{A}_\Phi$ such that

$$\mathbb{L}(\Phi) \cap \mathcal{C} = \mathbb{L}(\mathcal{A}_\Phi) \cap \mathcal{C}\,.$$

It would be interesting to have a better understanding of what are the classes of MSCs over which CFMs (i.e., EMSO) are equivalent to MSO. Most of the usual restrictions of MSCs studied in the literature have bounded tree-width, so a first question might be:

> Over MSCs of bounded tree-width, are CFMs expressively equivalent to MSO?

Note that the question of whether EMSO is equivalent to MSO over arbitrary *graphs* of bounded tree-width has already been studied, with a negative answer [76].

Beyond CFMs. Relations between logic and automata are also worth investigating for slightly different models of message-passing systems. One possibility is to consider recursive (rather than finite-state) processes, which can be modeled by adding well-nested call-return relations to MSCs. We could also include dynamic creations of processes.

Interestingly, the equivalence of EMSO and CFMs also has consequences for *parameterized* communicating automata, which are similar to CFMs but can be

run over any number of processes rather than having a fixed communication network. In [7], the equivalence of EMSO[$\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd$] and CFMs [13] is used as a black box to study the expressive power of parameterized communicating automata. Our result, about the logic EMSO[$\mathsf{Procs}, \mathsf{Prop}, \leq_{\mathsf{proc}}, \lhd$] rather than EMSO[$\mathsf{Procs}, \mathsf{Prop}, \rightarrow, \lhd$], could be used instead to obtain stronger results.

Bibliography

[1] R. Alur and P. Madhusudan. Adding nesting structure to words. *Journal of the ACM*, 56(3):16:1–16:43, 2009. doi:10.1145/1516512.1516518. (Cited on page 3.)

[2] R. Alur, D. A. Peled, and W. Penczek. Model-checking of causality properties. In *LICS'95*, pages 90–100. IEEE Computer Society, 1995. doi:10.1109/LICS.1995.523247. (Cited on page 110.)

[3] T. Antonopoulos, P. Hunter, S. Raza, and J. Worrell. Three variables suffice for real-time logic. In *FoSSaCS'15*, volume 9034 of *LNCS*, pages 361–374. Springer, 2015. doi:10.1007/978-3-662-46678-0_23. (Cited on pages 5, 9, 19, 23, and 37.)

[4] J. Araújo. Formalizing sequence diagrams. In *Proceedings of the OOPSLA'98 Workshop on Formalizing UML. Why? How?*, volume 33, 10 of *ACM SIGPLAN Notices*, New York, 1998. ACM Press. (Cited on page 65.)

[5] A. Arnold. An extension of the notions of traces and of asynchronous automata. *RAIRO – Informatique Théorique et Applications*, 25:355–393, 1991. doi:10.1051/ita/1991250403551. (Cited on page 40.)

[6] M. Bojańczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data words. *ACM Transactions on Computational Logic*, 12(4):27:1–27:26, 2011. doi:10.1145/1970398.1970403. (Cited on page 3.)

[7] B. Bollig. Logic for communicating automata with parameterized topology. In *CSL-LICS'14*, pages 18:1–18:10. ACM, 2014. doi:10.1145/2603088.2603093. (Cited on page 142.)

[8] B. Bollig, M. Fortin, and P. Gastin. Communicating finite-state machines and two-variable logic. In *STACS'18*, volume 96 of *LIPIcs*, pages 17:1–

17:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.STACS.2018.17. (Cited on page 74.)

[9] B. Bollig, M. Fortin, and P. Gastin. It is easy to be wise after the event: Communicating finite-state machines capture first-order logic with "happened before". In *CONCUR'18*, volume 118 of *LIPIcs*, pages 7:1–7:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.CONCUR.2018.7. (Cited on pages 13, 24, and 74.)

[10] B. Bollig, M. Fortin, and P. Gastin. Communicating finite-state machines, first-order logic, and star-free propositional dynamic logic. *Journal of Computer and System Sciences*, to appear. (Cited on pages 13 and 74.)

[11] B. Bollig and D. Kuske. Muller message-passing automata and logics. *Information and Computation*, 206(9-10):1084–1094, 2008. doi:10.1016/j.ic.2008.03.010. (Cited on pages 6, 7, 70, 73, and 121.)

[12] B. Bollig, D. Kuske, and I. Meinecke. Propositional dynamic logic for message-passing systems. *Logical Methods in Computer Science*, 6(3), 2010. doi:10.2168/LMCS-6(3:16)2010. (Cited on pages 7, 11, 73, and 140.)

[13] B. Bollig and M. Leucker. Message-passing automata are expressively equivalent to EMSO logic. *Theoretical Computer Science*, 358(2-3):150–172, 2006. doi:10.1016/j.tcs.2006.01.014. (Cited on pages 6, 7, 68, 73, 129, 140, and 142.)

[14] D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2):323–342, 1983. doi:10.1145/322374.322380. (Cited on pages 6, 65, and 70.)

[15] J. R. Büchi. Weak second order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960. doi:10.1002/malq.19600060105. (Cited on pages 2 and 72.)

[16] J. R. Büchi. On a decision method in restricted second order arithmetic. In E. Nagel, P. Suppes, and A. Tarski, editors, *LMPS'60*, pages 1–11. Stanford University Press, 1962. (Cited on page 3.)

[17] A. Church. Logic, arithmetic, and automata. In *ICM'62*, pages 23–35, 1963. (Cited on page 1.)

[18] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Proceedings of the Workshop on Logics of Programs*, volume 131 of *LNCS*, pages 52–71. Springer, 1981. doi:10.1007/BFb0025774. (Cited on page 2.)

[19] R. Danecki. Nondeterministic propositional dynamic logic with intersection is decidable. In *SCT'84*, volume 208 of *LNCS*, pages 34–53. Springer, 1985. doi:10.1007/3-540-16066-3_5. (Cited on page 10.)

[20] A. Dawar. How many first-order variables are needed on finite ordered structures? In *We Will Show Them! Essays in Honour of Dov Gabbay, Volume One*, pages 489–520. College Publications, 2005. (Cited on pages 5, 9, 19, and 138.)

[21] G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *AAAI'94 (vol. 1)*, pages 205–212. AAAI Press / The MIT Press, 1994. URL: http://www.aaai.org/Library/AAAI/1994/aaai94-032.php. (Cited on pages 10 and 20.)

[22] V. Diekert. A partial trace semantics for petri nets. *Theoretical Computer Science*, 134(1):87–105, 1994. doi:10.1016/0304-3975(94)90280-1. (Cited on page 40.)

[23] V. Diekert and P. Gastin. Pure future local temporal logics are expressively complete for mazurkiewicz traces. *Information and Computation*, 204(11):1597–1619, 2006. doi:10.1016/j.ic.2006.07.002. (Cited on pages 4, 9, 39, 109, and 110.)

[24] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, 1995. doi:10.1142/2563. (Cited on pages 38 and 73.)

[25] J. Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4(5):406–451, 1970. doi:10.1016/S0022-0000(70)80041-1. (Cited on page 3.)

[26] M. Droste, P. Gastin, and D. Kuske. Asynchronous cellular automata for pomsets. *Theoretical Computer Science*, 247(1-2):1–38, 2000. doi:10.1016/S0304-3975(00)00166-3. (Cited on page 40.)

[27] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98:21–52, 1961. doi:10.2307/1993511. (Cited on pages 2 and 72.)

[28] M. J. Fischer and R. E. Ladner. Propositional Dynamic Logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979. doi:10.1016/0022-0000(79)90046-1. (Cited on pages 2, 10, 20, and 22.)

[29] M. Fortin. FO = FO^3 for linear orders with monotone binary relations. In *ICALP'19*, volume 132 of *LIPIcs*, pages 116:1–116:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.116. (Cited on pages 13 and 24.)

[30] D. M. Gabbay. Expressive functional completeness in tense logic. In U. Mönnich, editor, *Aspects of Philosophical Logic: Some Logical Forays into Central Notions of Linguistics and Philosophy*, pages 91–117. Springer Netherlands, Dordrecht, 1981. doi:10.1007/978-94-009-8384-7_4. (Cited on pages 4, 5, 19, and 140.)

[31] D. M. Gabbay, I. Hodkinson, and M. A. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects, vol. 1.* Oxford University Press, 1994. (Cited on page 4.)

[32] D. M. Gabbay, I. M. Hodkinson, and M. A. Reynolds. *Temporal expressive completeness in the presence of gaps*, volume 2 of *Lecture Notes in Logic*, pages 89–121. Springer-Verlag, 1993. (Cited on pages 4 and 37.)

[33] P. Gastin and D. Kuske. Uniform satisfiability in PSPACE for local temporal logics over Mazurkiewicz traces. *Fundamenta Informaticae*, 80(1-3):169–197, 2007. (Cited on pages 109 and 110.)

[34] B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Information and Computation*, 204(6):920–956, 2006. doi:10.1016/j.ic.2006.01.005. (Cited on pages 6, 7, 11, 13, 69, 72, 73, 117, 122, 123, 129, and 141.)

[35] B. Genest, D. Kuske, and A. Muscholl. On communicating automata with bounded channels. *Fundamenta Informaticae*, 80(1-3):147–167, 2007. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi80-1-3-09. (Cited on pages 6, 7, 71, and 141.)

[36] S. Givant. The calculus of relations as a foundation for mathematics. *Journal of Automated Reasoning*, 37(4):277–322, 2006. doi:10.1007/s10817-006-9062-x. (Cited on page 25.)

[37] S. Göller, M. Lohrey, and C. Lutz. PDL with intersection and converse: satisfiability and infinite-state model checking. *Journal of Symbolic Logic*, 74(1):279–314, 2009. doi:10.2178/jsl/1231082313. (Cited on pages 10, 20, and 22.)

[38] M. Grohe. Finite variable logics in descriptive complexity theory. *Bulletin of Symbolic Logic*, 4(4):345–398, 1998. doi:10.2307/420954. (Cited on page 4.)

[39] M. Grohe and N. Schweikardt. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science*, 1(1), 2005. doi:10.2168/LMCS-1(1:6)2005. (Cited on page 139.)

[40] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(2):319–379, 1992. doi:10.1016/0004-3702(92)90049-4. (Cited on pages 10 and 20.)

[41] L. Hella, M. Järvisalo, A. Kuusisto, J. Laurinharju, T. Lempiäinen, K. Luosto, J. Suomela, and J. Virtema. Weak models of distributed computing, with connections to modal logic. *Distributed Computing*, 28(1):31–53, 2015. doi:10.1007/s00446-013-0202-3. (Cited on page 3.)

[42] L. Henkin. *Logical Systems Containing Only a Finite Number of Symbols*. Séminaire de Mathématiques Supérieures: Publications. Presses de l'Université de Montréal, 1967. (Cited on page 23.)

[43] J. G. Henriksen, J. L. Jensen, M. E. Jørgensen, N. Klarlund, R. Paige, T. Rauhe, and A. Sandholm. Mona: Monadic second-order logic in practice. In *TACAS'95*, volume 1019 of *LNCS*, pages 89–110. Springer, 1995. doi:10.1007/3-540-60630-0_5. (Cited on page 3.)

[44] J. G. Henriksen, M. Mukund, K. N. Kumar, M. A. Sohoni, and P. S. Thiagarajan. A theory of regular MSC languages. *Information and Computation*, 202(1):1–38, 2005. doi:10.1016/j.ic.2004.08.004. (Cited on pages 6, 7, 69, 71, 72, and 141.)

[45] Y. Hirshfeld and A. Rabinovich. Expressiveness of metric modalities for continuous time. *Logical Methods in Computer Science*, 3(1), 2007. doi:10.2168/LMCS-3(1:3)2007. (Cited on pages 4, 46, and 140.)

[46] I. M. Hodkinson. Finite H-dimension does not imply expressive completeness. *Journal of Philosophical Logic*, 23(5):535–573, 1994. doi:10.1007/BF01049409. (Cited on pages 4, 5, and 140.)

[47] I. M. Hodkinson and A. Simon. The k-variable property is stronger than H-dimension k. *Journal of Philosophical Logic*, 26(1):81–101, 1997. doi:10.1023/A:1017951631048. (Cited on pages 5 and 19.)

[48] P. Hunter. When is metric temporal logic expressively complete? In *CSL'13*, volume 23 of *LIPIcs*, pages 380–394. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.CSL.2013.380. (Cited on pages 4 and 46.)

[49] P. Hunter, J. Ouaknine, and J. Worrell. Expressive completeness for metric temporal logic. In *LICS'13*, pages 349–357. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.41. (Cited on pages 4 and 46.)

[50] N. Immerman. Upper and lower bounds for first order expressibility. *Journal of Computer and System Sciences*, 25(1):76–98, 1982. doi:10.1016/0022-0000(82)90011-3. (Cited on pages 5 and 138.)

[51] N. Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999. doi:10.1007/978-1-4612-0539-5. (Cited on page 4.)

[52] N. Immerman and D. Kozen. Definability with bounded number of bound variables. *Information and Computation*, 83(2):121–139, 1989. doi:10.1016/0890-5401(89)90055-2. (Cited on pages 5, 19, 23, 37, and 138.)

[53] ITU-TS. ITU-TS Recommendation Z.120: Message Sequence Chart 1999 (MSC99). Technical report, ITU-TS, Geneva, 1999. (Cited on page 65.)

[54] H. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968. (Cited on pages 2, 19, and 37.)

[55] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983. doi:10.1016/0304-3975(82)90125-6. (Cited on page 2.)

[56] D. Kuske. Regular sets of infinite message sequence charts. *Information and Computation*, 187:80–109, 2003. doi:10.1016/S0890-5401(03)00123-8. (Cited on pages 6, 7, 70, 71, 72, 130, 134, and 141.)

[57] A. Kuusisto. Modal logic and distributed message passing automata. In *CSL'13*, volume 23 of *LIPIcs*, pages 452–468. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.CSL.2013.452. (Cited on page 3.)

[58] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978. doi:10.1145/359545.359563. (Cited on pages 6 and 65.)

[59] M. Lange. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4(1):39–49, 2006. doi:10.1016/j.jal.2005.08.002. (Cited on pages 10 and 20.)

[60] M. Lange and C. Lutz. 2-ExpTime lower bounds for propositional dynamic logics with intersection. *Journal of Symbolic Logic*, 70(4):1072–1086, 2005. doi:10.2178/jsl/1129642115. (Cited on pages 10 and 20.)

[61] M. Lohrey and A. Muscholl. Bounded MSC communication. *Information and Computation*, 189(2):160–181, 2004. doi:10.1016/j.ic.2003.10.002. (Cited on pages 69 and 123.)

[62] C. Lutz, U. Sattler, and F. Wolter. Modal logic and the two-variable fragment. In *CSL'01*, volume 2142 of *LNCS*, pages 247–261. Springer, 2001. doi:10.1007/3-540-44802-0_18. (Cited on page 28.)

[63] C. Lutz and D. Walther. PDL with negation of atomic programs. *Journal of Applied Non-Classical Logics*, 15(2):189–213, 2005. doi:10.3166/jancl.15.189-213. (Cited on page 10.)

[64] M. Marx. Conditional xpath. *ACM Transactions on Database Systems*, 30(4):929–959, 2005. doi:10.1145/1114244.1114247. (Cited on pages 4 and 138.)

[65] B. Meenakshi and R. Ramanujam. Reasoning about layered message passing systems. *Computer Languages, Systems and Structures*, 30(3-4):171–206, 2004. doi:10.1016/j.cl.2004.02.003. (Cited on page 7.)

[66] R. Mennicke. Propositional dynamic logic with converse and repeat for message-passing systems. *Logical Methods in Computer Science*, 9(2), 2013. doi:10.2168/LMCS-9(2:12)2013. (Cited on page 7.)

[67] D. A. Peled. Specification and verification of message sequence charts. In *Formal Techniques for Distributed System Development, FORTE/PSTV 2000*, volume 183 of *IFIP Conference Proceedings*, pages 139–154. Kluwer, 2000. (Cited on page 7.)

[68] A. Pnueli. The temporal logic of programs. In *FOCS'77*, pages 46–57. IEEE Computer Society, 1977. doi:10.1109/SFCS.1977.32. (Cited on page 2.)

[69] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *POPL'89*, pages 179–190. ACM Press, 1989. doi:10.1145/75277.75293. (Cited on page 1.)

[70] B. Poizat. Deux ou trois choses que je sais de Ln. *Journal of Symbolic Logic*, 47(3):641–658, 1982. doi:10.2307/2273594. (Cited on pages 5, 19, and 23.)

[71] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969. doi:10.1090/S0002-9947-1969-0246760-1. (Cited on page 3.)

[72] K. Reinhardt. The complexity of translating logic to finite automata. In *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *LNCS*, pages 231–238. Springer, 2001. doi:10.1007/3-540-36387-4_13. (Cited on page 139.)

[73] F. Reiter. Distributed graph automata. In *LICS'15*, pages 192–201. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.27. (Cited on page 3.)

[74] F. Reiter. Asynchronous distributed automata: A characterization of the modal mu-fragment. In *ICALP'17*, volume 80 of *LIPIcs*, pages 100:1–100:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.100. (Cited on page 3.)

[75] B. Rossman. On the constant-depth complexity of k-clique. In *STOC'08*, pages 721–730. ACM, 2008. doi:10.1145/1374376.1374480. (Cited on pages 5, 9, 19, and 138.)

[76] I. Schiering. A hierarchical approach to monadic second-order logic over graphs. In *CSL'97*, volume 1414 of *LNCS*, pages 424–440. Springer, 1997. doi:10.1007/BFb0028029. (Cited on page 141.)

[77] B. Schlingloff. Expressive completeness of temporal logic of trees. *Journal of Applied Non-Classical Logics*, 2(2):157–180, 1992. doi:10.1080/11663081.1992.10510780. (Cited on page 4.)

[78] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985. doi:10.1145/3828.3837. (Cited on pages 2 and 40.)

[79] P. H. Starke. Processes in petri nets. *Elektronische Informationsverarbeitung und Kybernetik*, 17(8/9):389–416, 1981. (Cited on page 40.)

[80] L. J. Stockmeyer. *The Complexity of Decision Problems in Automata Theory and Logic*. PhD thesis, MIT, 1974. (Cited on pages 2, 3, 7, 40, 121, and 139.)

[81] R. S. Streett. Propositional dynamic logic of looping and converse. In *STOC'81*, pages 375–383. ACM, 1981. doi:10.1145/800076.802492. (Cited on pages 10 and 20.)

[82] A. Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6(3):73–89, 1941. doi:10.2307/2268577. (Cited on page 25.)

[83] A. Tarski and S. R. Givant. *A formalization of set theory without variables*. American Mathematical Society Providence, R.I, 1987. (Cited on pages 10 and 25.)

[84] J. W. Thatcher and J. B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2(1):57–81, 1968. doi:10.1007/BF01691346. (Cited on page 3.)

[85] P. S. Thiagarajan. A trace based extension of linear time temporal logic. In *LICS'94*, pages 438–447. IEEE Computer Society, 1994. doi:10.1109/LICS.1994.316047. (Cited on pages 109 and 110.)

[86] P. S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. *Information and Computation*, 179(2):230–249, 2002. doi:10.1006/inco.2001.2956. (Cited on pages 4, 124, and 130.)

[87] W. Thomas. On logical definability of trace languages. In *Proceedings of Algebraic and Syntactic Methods in Computer Science (ASMICS)*, Report TUM-I9002, Technical University of Munich, pages 172–182, 1990. (Cited on page 3.)

[88] W. Thomas. On logics, tilings, and automata. In *ICALP'91*, volume 510 of *LNCS*, pages 441–454. Springer, 1991. doi:10.1007/3-540-54233-7_154. (Cited on page 3.)

[89] W. Thomas. Elements of an automata theory over partial orders. In *Partial Order Methods in Verification, Proceedings of a DIMACS Workshop*, volume 29 of *DIMACS*, pages 25–40. AMS, 1996. doi:10.1090/dimacs/029/02. (Cited on page 3.)

[90] B. A. Trakhtenbrot. Finite automata and monadic second order logic. *Siberian Mathematical Journal*, 3:103–131, 1962. In Russian; English translation in Amer. Math. Soc. Transl. 59, 1966, 23–55. (Cited on pages 2 and 72.)

[91] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *LICS'86*, pages 332–344. IEEE Computer Society, 1986. (Cited on pages 2, 3, and 40.)

[92] Y. Venema. Two-dimensional modal logics for relation algebras and temporal logic of intervals. *ITLI prepublication series*, LP-1989-03, 1989. URL: https://eprints.illc.uva.nl/id/eprint/1140. (Cited on page 10.)

[93] Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990. (Cited on pages 5 and 9.)

[94] Y. Venema. A modal logic for chopping intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991. doi:10.1093/logcom/1.4.453. (Cited on page 10.)

[95] I. Walukiewicz. Local logics for traces. *Journal of Automata, Languages and Combinatorics*, 7(2):259–290, 2002. doi:10.25596/jalc-2002-259. (Cited on page 110.)

[96] W. Zielonka. Notes on finite asynchronous automata. *RAIRO – Informatique Théorique et Applications*, 21(2):99–135, 1987. doi:10.1051/ita/1987210200991. (Cited on page 3.)

Overview of logics

MSO[Σ] (Monadic Second-Order Logic) p.17

$$\Phi ::= P(x) \mid \alpha(x,y) \mid x = y \mid x \in X \mid \Phi \vee \Phi \mid \neg\Phi \mid \exists x.\Phi \mid \exists X.\Phi$$

and FO[Σ] ((Monadic) First-Order Logic) p.18

$$\Phi ::= P(x) \mid \alpha(x,y) \mid x = y \mid x \in X \mid \Phi \vee \Phi \mid \neg\Phi \mid \exists x.\Phi$$

$P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel}$, x, y first-order variables, X a second-order variable.

PDL[Σ] (Propositional Dynamic Logic) p.20

$$\begin{aligned} \xi &::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi && \text{(sentences)} \\ \varphi &::= P \mid \mathit{true} \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi && \text{(event formulas)} \\ \pi &::= \alpha \mid \{\varphi\}? \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^* && \text{(path formulas)} \end{aligned}$$

and ICPDL[Σ] (PDL with converse and intersection) p.20

$$\begin{aligned} \xi &::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi && \text{(sentences)} \\ \varphi &::= P \mid \mathit{true} \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi && \text{(event formulas)} \\ \pi &::= \alpha \mid \{\varphi\}? \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^* \mid \pi \cap \pi \mid \pi^{-1} && \text{(path formulas)} \end{aligned}$$

$P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel}$.

$\mathrm{PDL}_{\mathsf{sf}}[\Sigma]$ (Star-free PDL) p.24

$$\xi ::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi$$
$$\varphi ::= \mathit{true} \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi$$
$$\pi ::= \alpha \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^{\mathsf{c}}$$

$P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel}$.

$\mathrm{PDL}^{\mathsf{int}}_{\mathsf{sf}}[\Sigma]$ (Interval-preserving fragment of star-free PDL) p.33

$$\xi ::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi$$
$$\varphi ::= \mathit{true} \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$
$$\pi ::= \alpha \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid (\lambda \cdot \pi \cdot \mu)^{\mathsf{c}}$$

$P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel}$, $\lambda, \mu \in \{\leq, \geq\}$.

$\mathrm{PDL}^{<\bullet}_{\mathsf{sf}}[\Sigma]$ (A fragment of star-free PDL without complement) p.42

$$\xi ::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi$$
$$\varphi ::= \mathit{true} \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$
$$\sigma ::= \alpha \mid \mathsf{inf}\ (\lambda \cdot \alpha) \mid \mathsf{sup}\ (\lambda \cdot \alpha)$$
$$\pi ::= \sigma \mid \sigma^{-1} \mid \{\varphi\}? \mid <_{\varphi} \mid >_{\varphi} \mid \pi \cdot \pi$$

$P \in \mathsf{Prop}$, $\alpha \in \mathsf{Rel} \setminus \{\leq\}$, $\lambda \in \{\leq, <, \geq, >\}$.

$\mathrm{PDL}^{\mathrm{MSC}}_{\mathsf{sf}}[\mathsf{Procs}, \mathsf{Prop}]$ (A simpler fragment of $\mathrm{PDL}_{\mathsf{sf}}$ for MSCs) p.79

$$\xi ::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi$$
$$\varphi ::= \mathit{true} \mid p \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$
$$\pi ::= \top \mid \lhd_{p,q} \mid \lhd^{-1}_{p,q} \mid \{\varphi\}? \mid (<_{\mathsf{proc}})_{\varphi} \mid (>_{\mathsf{proc}})_{\varphi} \mid \pi \cdot \pi$$

$p \in \mathsf{Procs}$, $P \in \mathsf{Prop}$, $(p, q) \in \mathit{Ch}$.

$\mathrm{TL}[\mathsf{Procs}, \mathsf{Prop}, \langle\lhd\rangle, \langle\lhd^{-1}\rangle, \mathsf{Co}, \mathsf{SU}, \mathsf{SS}]$ (A temporal logic for MSCs) p.109

$$\varphi ::= \mathit{true} \mid p \mid P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\lhd\rangle\,\varphi \mid \langle\lhd^{-1}\rangle\,\varphi \mid \mathsf{Co}\,\varphi \mid \varphi\,\mathsf{SU}\,\varphi \mid \varphi\,\mathsf{SS}\,\varphi$$

$p \in \mathsf{Procs}$, $P \in \mathsf{Prop}$.

Index

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Expressivité de la logique du premier ordre, de la logique dynamique propositionnelle sans étoile et des automates communicants

Mots clés: logique du premier ordre, logique avec k variables, logique dynamique propositionnelle, logiques temporelles, automates communicants

Résumé: Cette thèse porte sur l'expressivité de la logique du premier ordre et d'autres formalismes sur différentes classes de structures ordonnées, parmi lesquelles les MSC (Message Sequence Charts), un modèle standard pour les exécutions de systèmes concurrents avec échange de messages. Cette étude est motivée par deux questions classiques : celle de l'équivalence, pour certaines classes de structures, entre la logique du premier ordre et son fragment avec k variables, et celle de la comparaison entre automates et logique, dans l'esprit du théorème de Büchi-Elgot-Trakhtenbrot. Notre approche repose sur la logique dynamique propositionnelle sans étoile (PDL sans étoile), une variante de PDL équivalente à la logique du premier ordre avec 3 variables.

On étudie d'abord l'expressivité de PDL sans étoile sur des structures linéairement ordonnées avec des prédicats unaires et binaires. On montre que sous certaines conditions de monotonie, PDL sans étoile devient aussi expressive que la logique du premier ordre. Cela implique que toute formule de la logique du premier ordre peut alors être réécrite en une formule équivalente qui utilise au plus 3 variables. Ce résultat s'applique, directement ou indirectement, à un certain nombre de classes naturelles, généralisant des résultats connus et répondant à des questions ouvertes.

On se concentre ensuite sur les MSC, auxquels ce premier résultat s'applique également. PDL sans étoile nous permet d'aborder un autre problème important: celui de la synthèse d'automates communicants à partir de spécifications écrites en logique du premier ordre. Les automates communicants sont un modèle de systèmes concurrents dans lequel un nombre fixé d'automates finis échangent des messages via des canaux FIFO. Ils définissent des langages de MSC. Bien que des caractérisations de l'expressivité des automates communicants aient déjà été établies pour certaines restrictions (borne sur la taille des canaux de communications, ou omission de la relation "arrivé-avant" au niveau de la logique), la question suivante restait ouverte dans le cas général : toute formule du premier ordre sur les MSC peut-elle être traduite en un automate communicant équivalent ? On montre que c'est le cas, en utilisant PDL sans étoile comme langage intermédiaire.

Title: Expressivity of first-order logic, star-free propositional dynamic logic and communicating automata

Keywords: first-order logic, k-variable property, propositional dynamic logic, temporal logics, communicating finite-state machines

Abstract: This thesis is concerned with the expressive power of first-order logic and other formalisms over different classes of ordered structures, among which MSCs (Message Sequence Charts), a standard model for executions of message-passing systems. This study is motivated by two classic problems: the k-variable property, that is, the equivalence of first-order logic and its k-variable fragment over certain classes of structures, and the study of logic-automata connections, in the spirit of Büchi-Elgot-Trakhtenbrot theorem. Our approach relies on star-free propositional dynamic logic (star-free PDL), a variant of PDL with the same expressive power as the 3-variable fragment of first-order logic.

We start by studying the expressive power of star-free PDL over linearly ordered structures with unary and binary predicates. We show that under certain monotonicity conditions, star-free PDL becomes as expressive as first-order logic. This implies that any first-order formula can then be rewritten into an equivalent formula with at most 3 variables. This result applies to various natural classes of structures, generalizing several known results and answering some open questions.

We then focus on MSCs, to which this first result also applies. We use star-free PDL to address another important problem: the synthesis of communicating finite-state machines (CFMs) from first-order specifications. CFMs are a model of concurrent systems in which a fixed number of finite-state automata communicate through unbounded FIFO channels. They accept languages of MSCs. While logical characterizations of the expressive power of CFMs have been established under different restrictions (bounding the size of the communication channels, or removing the "happened-before" relation from the logic), the following question had remained open in the general case: can every first-order formula over MSCs be translated into an equivalent CFM? We prove that this is the case, using star-free PDL as an intermediate language.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France