Chapter 1

Introduction

Modal and temporal logics have seen widespread usage in the industry as languages for expressing property about the evolution of complex systems. Various models based on graphs and automaton have model checkers developed upon. Most of them are based upon propositional variation of modal and temporal logics, however some reasearcher started working on first-order or second-order variants. These logics suffer mainly two issues: they are in general undecidable and the semantics evaluation model suffer various issue with trans-world identity. Counterpart relations solve most of these issues and recent works successfully use counterpart relations with monadic second-order modal logics to obtain a decidable calculus in finite domains and a companion model checker prototype. However they are based upon fixpoints and modal operators which are not really intuitive. The semantics evaluation function is complex and hard to follow by hand, and work is still needed regarding completeness and inference systems. The aim of this work is to develop a a counterpart semantics model for first-order/second-order linear temporal logic, and show that we can obtain a far simpler and understandable semantics but being still able to model plenty of properties and with an axiomatic model and an inference system.

We will first give an introduction to linear temporal logic and the issue of extending it to the first or second-order. Then we will present the mathematical tools used in our semantics, many-sorted algebras and counterpart models. We will defined the counterpart semantics for first-order/second-order linear temporal logic with many examples of properties and how to compute the semantics. Finally we will show some results on axioms and inference rules.

The main running example will be about multi-process communication over channels, modeled via graphs with processes as nodes and channels as edges. Some of the properties we will be able to express with our calculus are:

- Will all the channel be closed at the end of the execution?
- Which channels will be merged during execution?

- Are there isolated process at any point in time?
- Do the channels connection change at each step?
- Are new channels created during the execution?

Chapter 2

Linear Temporal Logic

In the following sections we will shortly summarise both syntax and semantics of linear temporal logic in the propositional form and show the issue of directly translating it to the quantified first-order version. Since many combinations of both unary and binary operators have been studied as forms of linear temporal logic, we will investigate only the somewhat common case of a logic with a singular primitive unary operation, *next*, and a singularo primitive binary operation, *until*. Nonetheless, both the classical approach and the one presented in this work can be trivially extended with other operators known in the literature, like *unless*.

2.1 Syntax

Definition 2.1.1 (Linear Temporal Logic)**.** *Let PROP be a set of atomic propositions. The set $\mathcal{F}$ of formulae for Linear Temporal Logic is the set generated by the following grammar:*

$$\phi ::= tt \mid p \mid \neg\phi \mid \phi \vee \phi \mid X\phi \mid \phi \, U \phi$$

where $p \in PROP$, X is a unary operator which states that ϕ must hold at the next step and U is a binary operator which states that the first formula must hold until the second formula holds at some point in the current or next steps.

Classical propositional logic connetives can be derived trivially:

$$\phi_1 \wedge \phi_2 \equiv \neg(\neg\phi_1 \vee \neg\phi_2) \qquad \phi_1 \rightarrow \phi_2 \equiv \neg\phi_1 \vee \phi_2$$

Other temporal operators used in temporal logic literature are derivable as such:

$$F\phi \equiv tt \, U \phi \qquad G\phi \equiv \neg F(\neg\phi) \qquad \phi_1 \, W \phi_2 \equiv (\phi_1 \, U \phi_2) \vee G\phi_1$$

Intuitively $F\phi$ means that eventually at some next step in time the formula ϕ holds, $G\phi$ means that the formula ϕ holds at each possible next step.

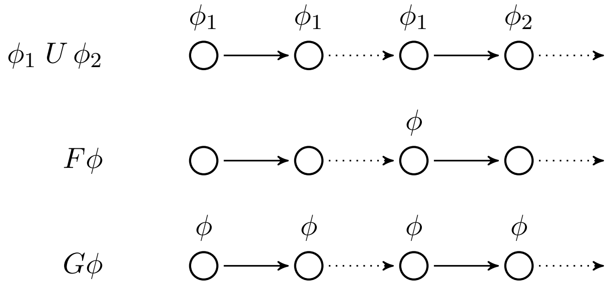

Figure 2.1: Example of temporal traces for operators

LTL formulae can encode safety properties, commonly with the form $G\neg\phi$, and liveness properties of systems, commonly with the form $GF\phi$ or $G(\phi_1 \to F\phi_2)$. Let's assume that we have two concurrent processes and that we can encode with two atomic propositions, crit_1 and crit_2, when the two processes can access a critical section, then the property of mutual exclusivity can be model by the formula $G(\neg\text{crit}_1 \vee \neg\text{crit}_2)$. If we also have atomic propositions for wait conditions, wait_1, then we can model liveness conditions, for example $G(\text{wait}_1 \to F\text{crit}_1)$ which means that whenever a process reaches a wait condition, eventually will enter the critical section; we can also model fairness conditions, for example $GF\text{wait}_1 \to GF\text{crit}_1$ means that if the process infinitely often reaches a wait condition than infinitely often will reach the critical section.

2.2 Semantic

The de-facto standard semantics for Linear Temporal Logic and Temporal Logics in general are Kripke-style semantics, as for modal logics. In modal logic, the so-called Kripke-frames are pairs of set of worlds and an accessibility relation between worlds. For temporal logic, each world is a point in time and the accessibility relation models the flow of time, thus a temporal model is defined as:

Definition 2.2.1 (Temporal model)**.** *A* temporal model M *is a triple* $(T, \prec, L)$ *where* T *is a set of time-points,* $\prec$ *is an accessibility relations over* T *and* $L : T \to \mathcal{P}(PROP)$ *is a labelling function that for each point in time returns the subset of all atomic propositions which are valid at that point.*

Additionally, the pair $(T, \prec)$ is called a temporal frame.

Definition 2.2.2 (LTL semantic)**.** *The truthness of a LTL formula* ϕ *at the time-point* t *over the temporal model* M*, denoted as* $M, t \vDash \phi$ *is defined* *inductively as follows:*

$$\begin{array}{ll}
M, t \vDash tt & \\
M, t \vDash p & \textit{iff } p \in L(t) \\
M, t \vDash \neg\phi & \textit{iff } M, t \nvDash \phi \\
M, t \vDash \phi_1 \vee \phi_2 & \textit{iff } M, t \vDash \phi_1 \ \textit{or } M, t \vDash \phi_2 \\
M, t \vDash X\phi & \textit{iff } \forall t \prec t'. \, M, t' \vDash \phi \\
M, t \vDash \phi_1 \, U \phi_2 & \textit{iff } M, t \vDash \phi_2 \ \textit{or } (M, t \vDash \phi_1 \ \textit{and } \forall t \prec t'. \, M, t' \vDash \phi_1 \, U \phi_2)
\end{array}$$

2.3 First-Order Extension

LTL can be extended to a first-order calculus by introducing predicates and variables, in the same vein as the transition from propositional to first-order logic in the classical framework. Naturally, because we are additionally working with time-dependent components, predicate symbols and variables can be interpreted differently at different point in time. Symbols that are established to be time-dependent are called *flexible*, conversely time-independent symbols are called *rigid*.

Nonetheless both classical logic and linear temporal logic have in common the trait of becoming indecidable when passing from propositional calculi to first-order calculi. Furthermore First-Order Linear Temporal Logic is proven to be also incomplete.

In the following paragraphs we will show a general syntax and semantics of First-Order Linear Temporal Logic, what are the condition for completeness and some of the issue with the most common semantic.

Definition 2.3.1 (First-Order Linear Temporal Logic)**.** *Let $\mathcal{P}$ be a set of predicates each with a specific arity, and let X be a denumerable set of variables. The set $\mathcal{F}_{FO}$ of formulae for First-Order Linear Temporal Logic is the set generated by the following grammar:*

$$\phi ::= tt \mid P(x_1, \ldots, x_n) \mid \neg\phi \mid \phi \vee \phi \mid X\phi \mid \phi \, U \phi \mid \exists x. \, \phi$$

where $P \in \mathcal{P}$ and $x_1, \ldots, x_n \in X$ are individual variables that match the arity required by P.

Universal quantification can be modeled with the translation from the existential quantifier and negation:

$$\forall x. \, \phi \equiv \neg\exists x. \, \neg\phi$$

Example of properties can be modeled with the first-order extension are:

$$\exists x. \, G(\text{channel}(x) \wedge \text{open}(x))$$

with the intuitive meaning that there must always exists at least one open channel; or:

$$\forall x.\, G(\text{channel}(x) \rightarrow (\text{open}(x)\, U(\neg\exists y.\, \text{message}(y) \wedge \text{pending}(x, y))))$$

which intuitively means that channels must always remain open until there are no more pending messages on that channel.

For quantified LTL we need to extend the temporal frames with the domain of values that makes sense to talk about at each points in time.

Definition 2.3.2 (Kripke frame)**.** *A* Kripke frame *M is a quadruple $(T, \prec, D, d)$ where T is a set of time-points, $\prec$ is an accessibility relation over T, D is a function assigning to each point in time t a non-empty set $D(t)$ s.t. if $t \prec t'$ then $D(t) \subseteq D(t')$, d is a function assigning to each point in time t a set $d(t) \subseteq D(t)$.*

Intuitively, the so-called outer domains $D(t)$ represent the collection of object that can be referenced at the point in time t, conversely the so-called inner domains $d(t)$ represent the collection of object actually existing the point in time t. An assignement function σ is a function from the set of variables X to the outer-domain $D(t)$ of a given point in time t. Finally, we need an interpretation function I such that $I(P, t)$ assign for each predicate constant $P \in \mathcal{P}$ and point in time t a subset of $D^n(t)$ where n is the arity of P, and $I(x, t) = \sigma(x)$ where $x \in X$ and σ is a suitable assignment that is said to be inducing I, noted as I^σ.

Definition 2.3.3 (FO-LTL semantic)**.** *The truthness of a FO-LTL formula ϕ at the point in time t over the Kripke model M and the induced interpretation I^σ, denoted as $M, t, I^\sigma \vDash \phi$ is defined inductively as follows:*

$$\begin{array}{ll}
M, t, I^\sigma \vDash tt & \\
M, t, I^\sigma \vDash P(x_1, \ldots, x_n) & \textit{iff } (\sigma(x_1), \ldots, \sigma(x_2)) \in I^\sigma(P, t) \\
M, t, I^\sigma \vDash \neg\phi & \textit{iff } M, t, I^\sigma \nvDash \phi \\
M, t, I^\sigma \vDash \phi_1 \vee \phi_2 & \textit{iff } M, t, I^\sigma \vDash \phi_1 \textit{ or } M, t, I^\sigma \vDash \phi_2 \\
M, t, I^\sigma \vDash X\phi & \textit{iff } \forall t \prec t'.\, M, t', I^\sigma \vDash \phi \\
M, t, I^\sigma \vDash \phi_1\, U\phi_2 & \textit{iff } M, t, I^\sigma \vDash \phi_2 \textit{ or } (M, t, I^\sigma \vDash \phi_1 \textit{ and } M, t', I^\sigma \vDash \phi_1\, U\phi_2) \\
M, t, I^\sigma \vDash \exists x.\, \phi & \textit{iff } \exists v \in d(t).\, M, t, I^{\sigma[v/x]} \vDash \phi
\end{array}$$

2.4 Trans-World Identity

Kripke-frames requires that the so-called outer domains $D(t)$ are always increasing with time, i.e. if $t \prec t'$ then $D(t) \subseteq D(t')$. This condition is required to evaluate temporal operators, otherwise it would not be possible to denote a variable x in the future. Other works fix a universal domain equal for each point in time, thus $D(t) = D(t')$ for each $t, t' \in T$, which also follows from accessibility relations that contain loops. However, by imposing such condition we are identifying object *a priori* and universally, irrespective of time. This problem is called *trans-world identity* and extensive literature about it has been produced in the last half-century, be either philosophycal questions or possible pratical solutions with Kripke-style semantics.

Even with the additional constraints on object domains, there are still other issues with Kripke-style semantics and the system we are trying to model. Assume we are trying to model resource allocation in a complex system. Let i be a resource. Due to the outer domain condition, we can identify the exact point in time where the resource i is allocated, i.e. $i \in D(t)$ for some time point t. Note that the resource can be allocated, but still can be unused for a potentially infinite amount of time, since it may not belong to any inner domain, not even $d(t)$. However, at the same time, we cannot deallocate the resource i since it must always be referentiable at future point in times. This can be solved with infinite outer domains to ensure uniqueness or by restricting the class of admissible evolutions, but such solutions tend to hamper usability.

Another desiderable behaviour is merging, for example while modeling memory allocations, we may want to merge two memory allocated segments into a single segment and treat it as a single resource.

In the next chapter we will explore an alternative approach based on *counterpart relations*, which rejects the possibility of universally identifying objects among possible worlds.

Chapter 3

Counterpart Semantic

In this chapter we will introduce both syntax and semantics of the calculus, however we will use many-sorted algebras as models for the evaluation of the formulae.

3.1 Algebra

Definition 3.1.1 (Many-sorted Signature). *A* many-sorted signature Σ *is a pair* (S_Σ, F_Σ) *where* $S_\Sigma = \{\, \tau_1, \ldots, \tau_n \,\}$ *is a set of sorts, and* $F_\Sigma = \{\, f_\Sigma : \tau_1 \times \cdots \times \tau_n \to \tau \mid \tau_i, \tau \in \Sigma_\tau, i = 1, \ldots, n \,\}$ *is a set of function symbols.*

We show a couple of signatures that will be used throughout the work. First, let's model simple directed graphs, which have two sorts: vertices, τ_v, and edges τ_e. Naturally to identify edges we require two operations, s, t both of type $\tau_e \to \tau_v$, which respectively identify the source and target of a given edge. Thus the complete signature for simple directed graph is $\Sigma = (\{\, \tau_v, \tau_e \,\}, \{\, s : \tau_e \to \tau_v, t : \tau_e \to \tau_v \,\})$.

Next we present the signature for a simplified memory management model. Once again we have two sorts: objects, τ_o, and memory locations τ_l. The only operation is the function that associate to each object it's memory location, ρ. Thus the complete signature is $\Sigma = (\{\, \tau_o, \tau_l \,\}, \{\, \rho : \tau_o \to \tau_l \,\})$.

Definition 3.1.2 (Many-sorted Algebra). *A* many-sorted algebra $\mathbf{A}$ *with signature* $\Sigma = (S_\Sigma, F_\Sigma)$*, or* Σ*-algebra, is a pair* $(A, F_\Sigma^{\mathbf{A}})$ *such that:*

- A *is a family of carrier sets, indexed by the sorts of* Σ*;*
- $F_\Sigma^{\mathbf{A}}$ *is a family of functions, indexed by the function symbols in* F_Σ*,* $\{\, f_\Sigma^{\mathbf{A}} : A_{\tau_1} \times \cdots \times A_{\tau_n} \to A_\tau \mid f_\Sigma : \tau_1 \times \cdots \times \tau_n \to \tau \in F_\Sigma \,\}$.

Definition 3.1.3 (Homomorphism). *Given two* Σ*-algebras* $\mathbf{A}$ *and* $\mathbf{B}$*, a* (partial) homomorphism *is a family of (partial) functions indexed by the*

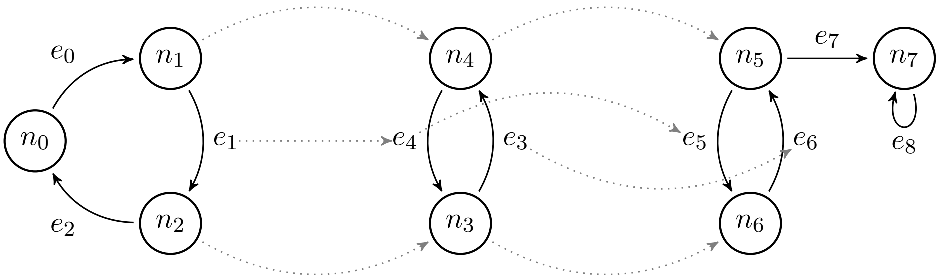

Figure 3.1: Examples of simple directed graphs

sorts of Σ, $\{\, \rho_\tau : A_\tau \rightharpoonup B_\tau \mid \tau \in S_\Sigma \,\}$, *such that for each function symbol* $f_\Sigma : \tau_1 \times \cdots \times \tau_n \in F_\Sigma$ *and list of elements* $a_1 \in A_{\tau_1}, \ldots, a_n \in A_{\tau_n}$*, if each function* ρ_{τ_i} *is defined for the element* a_i *then* ρ_τ *is defined for the element* $f^{\mathbf{A}}_\Sigma(a_1, \ldots, a_n)$ *and* $\rho_\tau(f^{\mathbf{A}}_\Sigma(a_1, \ldots, a_n)) = f^{\mathbf{B}}_\Sigma(\rho_{\tau_1}(a_1), \ldots, \rho_{\tau_n}(a_n))$*.*

Recall the signature for simple directed graphs defined earlier, a Σ-algebra for that signature is a particular instance of a simple directed graph and homomorphism between them are partial homomorphism between graphs. For example the graphs in Figure 3.1 are visual representation of the following algebras: $\mathbf{G}_0 = (\{\, \{\, n_0, n_1, n_2 \,\}, \{\, e_0, e_1, e_2 \,\} \,\}, \{\, s^{\mathbf{G}_0}, t^{\mathbf{G}_0} \,\})$, where $s^{\mathbf{G}_0} = \{\, e_0 \mapsto n_0, e_1 \mapsto n_1, e_2 \mapsto n_2 \,\}$ and $t^{\mathbf{G}_0} = \{\, e_0 \mapsto n_1, e_1 \mapsto n_2, e_2 \mapsto n_0 \,\}$; $\mathbf{G}_1 = (\{\, \{\, n_3, n_4 \,\}, \{\, e_3, e_4 \,\} \,\}, \{\, s^{\mathbf{G}_1}, t^{\mathbf{G}_1} \,\})$, where $s^{\mathbf{G}_1} = \{\, e_3 \mapsto n_3, e_4 \mapsto n_4 \,\}$ and $t^{\mathbf{G}_1} = \{\, e_3 \mapsto e_3, e_4 \mapsto n_3 \,\}$. $\mathbf{G}_2 = (\{\, \{\, n_5, n_6, n_7 \,\}, \{\, e_5, e_6, e_7, e_8 \,\} \,\}, \{\, s^{\mathbf{G}_2}, t^{\mathbf{G}_2} \,\})$, where $s^{\mathbf{G}_2} = \{\, e_5 \mapsto n_5, e_6 \mapsto n_6, e_7 \mapsto n_5, e_8 \mapsto n_7 \,\}$ and $t^{\mathbf{G}_2} = \{\, e_5 \mapsto n_6, e_6 \mapsto n_5, e_7 \mapsto n_7, e_8 \mapsto n_7 \,\}$

In the same figure are also visually represented partial homomorphism between those graphs that can be translated to: $\mathbf{G}_0 \to \mathbf{G}_1 = \{\, \{\, n_1 \mapsto n_4, n_2 \mapsto n_3 \,\}, \{\, e_1 \mapsto e_4 \,\} \,\}$ and $\mathbf{G}_1 \to \mathbf{G}_2 = \{\, \{\, n_4 \mapsto n_5, n_3 \mapsto n_6 \,\}, \{\, e_4 \mapsto e_5, e_3 \mapsto e_6 \,\} \,\}$, where the requirements for homomorphism are trivially checkable. Notice that the first example of homomorphism shows an interesting behaviour for our purpose. Due to partiality it removes the node n_0 and the edges e_0, e_2 while preserving the structure of the nodes n_1, n_2 and the edge e_1 between them.

To model existential properties, we can extend a signature Σ with a denumerable set X of variables typed over S_Σ, obtaining the signature Σ_X. The τ-typed subset of X is denoted with X_τ, and typed variables are denoted with x_τ or $x : \tau$. τ-sorted terms are denoted with ϵ_τ or $\epsilon : \tau$.

Definition 3.1.4 (Term)**.** *Let* Σ *be a signature, let* X *be a denumerable set of individual variables typed over* S_Σ*, and let* Σ_X *be the signature obtained by extending* Σ *with* X*. The (many-sorted) set* $T(\Sigma_X)$ *of* terms *is the smallest set such that:*

$$\frac{}{X \subseteq T(\Sigma_X)} \qquad \frac{f : \tau_1 \times \cdots \tau_n \to \tau \in F_\Sigma \qquad \forall i \in [1, n].\, \epsilon_i : \tau_i \in T(\Sigma_X)}{f(\epsilon_1, \ldots, \epsilon_n) : \tau \in T(\Sigma_X)}$$

Example 3.1.1 (Terms). *Let be $\Sigma = (\{\tau\}, \{1 : \tau, \otimes : \tau \to \tau\})$ a monoidal signature with a single sort. Let be X a denumerable set of variables with $x, y, z \in X$, then some example of terms in $T(\Sigma_X)$ are $1 \otimes 1 : \tau$, $x \otimes (y \otimes 1) : \tau$ and $(x \otimes 1) \otimes (1 \otimes y) : \tau$. Examples from the previously defined graph algebra Σ are $s(x)$ and $t(y)$ which represent respectively the source vertex for an edge $x : \tau_e$ and the target vertex for an edge $y : \tau_e$.*

3.2 Counterpart model

We now introduce the concept of *counterpart model*, as in "counterpart theory" of David Lewis.

Definition 3.2.1. *Let Σ be a signature, and $\mathcal{A}$ the set of algebras over the signature Σ. A* counterpart model *M is a triple $(W, \rightsquigarrow, d)$ such that:*

- *W is a set, representing worlds, i.e. points in time;*
- *$d : W \to \mathcal{A}$ is a function, assigning to each world $w \in W$ a Σ-algebra, $\mathbf{A} \in \mathcal{A}$;*
- *$\rightsquigarrow \subseteq W \times (\mathcal{A} \rightharpoonup \mathcal{A}) \times W$ is the* accessibility relation *over W, enriched with (partial) homomorphisms* (counterpart relations) *between the algebras of the connected worlds, i.e. for every $(w_1, cr, w_2) \in \rightsquigarrow$ it must hold that $cr : d(w_1) \rightharpoonup d(w_2)$ is a (partial) homomorphism.*

As a shorthand notation, counterpart relations between two worlds in the accessibility relation, $(w_1, cr, w_2) \in \rightsquigarrow$, will be also denoted as $w_1 \overset{cr}{\rightsquigarrow} w_2$. The accessibility relation $\rightsquigarrow$ defines the counterparts in the target world of the source world, effectively modeling the evolution of a system, modeled by the algebras, and avoiding the *trans-world identity* problem. Names are local to the belonging world, and components are identified across worlds only by the relation between names instead of by a universal name. This, and partiality, allows for creation, deletion, renaming and merging of elements in a type-preserving manner, however duplication is not permitted as counterpart relation are functions thus can only associate a single element of the target world to an element of the source world. In other terminology, the counterpart model can be seen as a generalisation of a *graph transition system* where transition are labelled with homomorphism between algebras, which are the state of the transition system and can be arbitrarily complex within themselves.

For the rest of the work, we will implicitly assume that every counterpart model M does not have deadlock worlds, i.e. worlds without outgoing transitions. This condition is not a limitation since a counterpart model that does not satisfy the condition can be transformed into one that does satisfy it by adding a reflexive transition to the accessibility relation, i.e. for each

w that is a deadlock world, $(w, \mathrm{id}_w, w) \in \rightsquigarrow$, or an ad-hoc world with only a reflexive transition. Such modification is typical of other works in the field, it does not impact the results of this work and it meaningful simplifies the semantics presented here.

Definition 3.2.2. *Let X be a denumerable sets of variables, and $M = (W, \rightsquigarrow, d)$ be a counterpart model over a signature Σ. A* variable assignment *σ for a world $w \in W$ is a (partial) function such that $\sigma : X \rightharpoonup d(w)$.*

Given a term $\epsilon \in T(\Sigma_X)$ and an assignment σ, we will denote as $\sigma(\epsilon)$ the lifting of σ to the set $T(\Sigma_X)$, i.e. applying the substition σ to each free variable in the term ϵ. If any of the free variables in ϵ are not in the domain of σ than $\sigma(\epsilon)$ is undefined.

From now on, we will consider only *formulae-in-context*, i.e. formulae with an appropriate context Γ that must contain at least all the free variables in the formula. Consequently, substitutions for a formula ϕ in a context Γ, will be defined over the context gamma. Note that the substitution can still be undefined over some or all the values in the domain of Γ, addressing the need of modeling deallocation of items.

3.3 Syntax

Definition 3.3.1 (First-Order Linear Temporal Logic)**.** *Let Σ be a (multi-sorted) signature and X a denumerable set of variables typed over S_Σ. The set $\mathcal{F}_\Sigma$ of formulae for First-Order Linear Temporal Logic is the set generated by the following grammar:*

$$\phi ::= tt \mid \epsilon_\tau = \epsilon_\tau \mid \neg\phi \mid \phi \vee \phi \mid \exists x_\tau.\, \phi \mid X\phi \mid \phi U \phi$$

where $\epsilon \in T(\Sigma_X)$, $\exists x_\tau$ ranges over variables of sort $\tau \in S_\Sigma$, O is a unary operator which states that ϕ must hold at the next step and U is a binary operator which states that the first formula must hold until the second formula holds at some point in the current or next steps.

Classical propositional logic and other temporal operators can be derived as for LTL, with the addition of the trivially derivable universal quantifier, $\forall x_\tau.\, \phi \equiv \neg\exists x_\tau.\, \neg\phi$. We will also define a shorthand for inequality, $x_\tau \neq \epsilon_\tau \equiv \neg(x_\tau = \epsilon_\tau)$.

The syntax is almost identical to the one introduced in definition 2.3.1, but predicates are replaced by equality of terms, which we will further specify in the next section, and the existential quantifier is instead a family of operators indexed by the sorts of the signature.

Example 3.3.1. *Let's model some general properties that can be resonable in a large space of signatures, as a consequence these will really be a family of typed predicates. Let's start with a predicate that identifies entities that will be* *deleted in the next step: will-be-deleted*$(x_\tau) \equiv (\exists y_\tau .\, x = y) \wedge X(\neg\exists y_\tau .\, x = y)$. *Another example is will-merge*$(x_\tau, y_\tau) \equiv (\exists z_\tau .\, \exists w_\tau .\, (x \neq y \wedge x = z \wedge y = w)\, U (\exists z_\tau .\, x = y \wedge x = z \wedge y =$ *which intuitively means that the arguments of the predicate will necessarily merge into a single entity in the current world or at some point in the future. Recall the signature for simple directed graphs introduced in section 3.1, we can define predicates on the structure of the worlds, e.g. loop*$(x_{\tau_e}) \equiv s(x) = t(x)$ *which means that the edge* x *must be a loop in the worlds that satisfy the predicate. We can also model the evolution of complex structures, e.g. at-most-2*$_\tau \equiv \forall x_\tau .\, \forall y_\tau .\, \forall z_\tau .\, x = y \vee y = z \vee z = x$ *is true for worlds where the number of entities of sort* τ *is bounded by 2. This predicate can be extended to larger boundaries or enforced during the evolution of the system, e.g. assume that the graph signature is used to model a net of unidirectional inter-process communication channel, we may want to verify that the number of channel, or edges, spawned during the whole execution is bounded by some number* n *than we must verify the formula* G*at-most-n; or we may want to make sure that after reaching maximum capacity, the programs continously deallocate channels until a minimum bound* m *is reached, thus* $G($*at-most-n* $\wedge$ $($*exactly-n* $\rightarrow$ $(\exists x.$ *will-be-deleted(x)*$)\, U$ *at-most-m*$))$*, where exactly-n is a predicate that is true when there are exactly* n *distinct entities of a given sort.*

3.4 Semantics

Definition 3.4.1 (Semantic of FO-LTL)**.** *Let* M *be a counterpart model,* Γ *a context and* (σ, w) *a pair substitution-world in the context* Γ*. The validity of a formula-in-context* ϕ*, denoted* $(\sigma, w) \vDash_\Gamma \phi$*, is defined inductively as follows:*

$$
\begin{array}{ll}
(\sigma, w) \vDash_\Gamma tt & \\
(\sigma, w) \vDash_\Gamma \epsilon_\tau = \eta_\tau & \textit{iff } \sigma(\epsilon) = \sigma(\eta) \\
(\sigma, w) \vDash_\Gamma \neg\phi & \textit{iff } (\sigma, w) \nvDash_\Gamma \phi \\
(\sigma, w) \vDash_\Gamma \phi_1 \vee \phi_2 & \textit{iff } (\sigma, w) \vDash_\Gamma \phi_1 \textit{ or } (\sigma, w) \vDash_\Gamma \phi_2 \\
(\sigma, w) \vDash_\Gamma \exists x_\tau .\, \phi & \textit{iff } \exists v \in d(w).\, (ext_x(\sigma, v), w) \vDash_{\Gamma, x} \phi \textit{ with } x \notin \Gamma \\
(\sigma, w) \vDash_\Gamma X\phi & \textit{iff } \forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \vDash_\Gamma \phi \\
(\sigma, w) \vDash_\Gamma \phi_1\, U \phi_2 & \textit{iff } (\sigma, w) \vDash_\Gamma \phi_2 \textit{ or } ((\sigma, w) \vDash_\Gamma \phi_1 \textit{ and } \forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \vDash_\Gamma \phi_1\, U \phi_2)
\end{array}
$$

Where $ext_x : (A \rightharpoonup B) \times B \rightarrow (A \cup \{\, x \,\} \rightharpoonup B)$ *is defined as:*

$$
ext_x(\sigma, v)(y) = \begin{cases} v & \textit{if } y = x \\ \sigma(y) & \textit{otherwise} \end{cases}
$$

The formula tt holds for any possible pair assignement-world. The predicate $=$ models equality for typed terms, where $\epsilon_\tau = \eta_\tau$ is valid if the evaluation is either undefined for both terms or they are equal according to the standard notion of equality. The negation of a formula is satisfied if the formula without the negation is not valid. The disjunction of two formulae is valid if either of them is valid. Existentially quantified formulae are valid if the formula is valid in the context extended with the quantified variable. The sub-formula must be valid in the same world and with the assignment extended to the context Γ, x with a new value within the domain of the algebra, thus behaving identically on all the variables in the original context Γ. Next we look at temporal operators. Formulae containing the next operator are valid if the sub-formula is valid in all the world accessible from w. The sub-formula is evaluated with the assignement composed with the counterpart relation between the accessible worlds. Formulae with the until operator, instead, are satisfied either if the post-condition is valid in the current world or if the pre-condition is valid and the whole formula is valid in all the accessible world, the same as for the next operator.

Definition 3.4.2. *A formula $\phi \in \mathcal{F}_\Sigma$ is called* valid for the counterpart model *M, denoted by $\vDash_M \phi$, if $(\sigma, w) \vDash_\Gamma \phi$ for every context Γ, world w and substitution σ.*

Definition 3.4.3. *A formula $\phi \in \mathcal{F}_\Sigma$ is called a* consequence *of a set $F \subseteq \mathcal{F}_\Sigma$, denoted $F \vDash \phi$, if $\vDash_M \phi$ holds for every model M with $\vDash_M \psi$ for all $\psi \in F$.*

If F is empty, then the formula is called valid*, denoted by $\vDash \phi$.*

3.5 Other operators

Let's explore the semantics of the other common operators defined in the previous chapters.

$$
\begin{aligned}
(\sigma, w) \vDash_\Gamma \phi_1 \wedge \phi_2 &\Leftrightarrow (\sigma, w) \nvDash_\Gamma \neg\phi_1 \vee \neg\phi_2 \\
&\Leftrightarrow (\sigma, w) \nvDash_\Gamma \neg\phi_1 \text{ and } (\sigma, w) \nvDash_\Gamma \neg\phi_2 \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi_1 \text{ and } (\sigma, w) \vDash_\Gamma \phi_2
\end{aligned} \tag{3.1}
$$

$$
\begin{aligned}
(\sigma, w) \vDash_\Gamma \phi_1 \rightarrow \phi_2 &\Leftrightarrow (\sigma, w) \vDash_\Gamma \neg\phi_1 \vee \phi_2 \\
&\Leftrightarrow (\sigma, w) \nvDash_\Gamma \neg\phi_1 \text{ or } (\sigma, w) \vDash_\Gamma \phi_2 \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi_1 \text{ implies } (\sigma, w) \vDash_\Gamma \phi_2
\end{aligned} \tag{3.2}
$$

$$
\begin{aligned}
(\sigma, w) \vDash_\Gamma \phi_1 \leftrightarrow \phi_2 &\Leftrightarrow ((\sigma, w) \vDash_\Gamma \phi_1 \rightarrow \phi_2) \text{ and } ((\sigma, w) \vDash_\Gamma \phi_2 \rightarrow \phi_1) \\
&\Leftrightarrow ((\sigma, w) \vDash_\Gamma \phi_1 \text{ implies } (\sigma, w) \vDash_\Gamma \phi_2) \\
&\qquad \text{and } ((\sigma, w) \vDash_\Gamma \phi_2 \text{ implies } (\sigma, w) \vDash_\Gamma \phi_1) \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi_1 \text{ iff } (\sigma, w) \vDash_\Gamma \phi_2
\end{aligned} \tag{3.3}
$$

$$
\begin{aligned}
(\sigma, w) \vDash_\Gamma \forall x.\, \phi &\Leftrightarrow (\sigma, w) \nvDash_\Gamma \exists x.\, \neg\phi \\
&\Leftrightarrow \nexists v \in d(w).\, (\sigma[v/x], w) \nvDash_\Gamma \phi \\
&\Leftrightarrow \forall v \in d(w).\, (\sigma[v/x], w) \vDash_\Gamma \phi
\end{aligned} \tag{3.4}
$$

$$
\begin{aligned}
(\sigma, w) \vDash_\Gamma F\phi &\Leftrightarrow (\sigma, w) \vDash_\Gamma tt\, U\phi \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi \text{ or } ((\sigma, w) \vDash_\Gamma tt \text{ and } \forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \vDash_\Gamma tt\, U\phi) \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi \text{ or } \forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \vDash_\Gamma tt\, U\phi \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi \text{ or } (\sigma, w) \vDash_\Gamma XF\phi \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi \vee XF\phi
\end{aligned} \tag{3.5}
$$

$$
\begin{aligned}
(\sigma, w) \vDash_\Gamma G\phi &\Leftrightarrow (\sigma, w) \vDash_\Gamma \neg F \neg\phi \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma (\sigma, w) \vDash_\Gamma \phi \text{ and } \neg\forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \vDash_\Gamma \neg\phi \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma (\sigma, w) \vDash_\Gamma \phi \text{ and } \exists w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \vDash_\Gamma \phi \\
&\Leftarrow (\sigma, w) \vDash_\Gamma \phi \wedge XG\phi
\end{aligned} \tag{3.6}
$$

The semantics of the derived operator matches our intuition, however the expansion rule for the forever operator is not bidirectional, as for an arbitrary pair (σ, w) there could be multiple accessible world from w and only some of them satisfy the body of the quantifier. Nonetheless, this is issue is marginal for two reason. First, if $G\phi$ is valid within the model M then the implication becomes bidirectional. Second, typically the accessibility relation in the models are linear order, in particular traces of a transition system, which once again solves the issue.

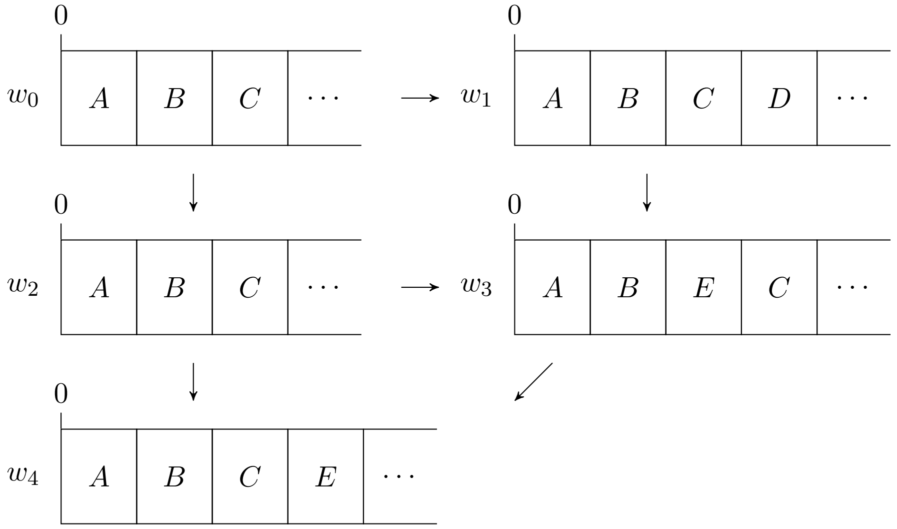

Figure 3.2: Example of counterpart model for a memory layout signature.

3.6 Example of execution

Here we will provide the evaluation of some formulae, using some of the predicates introduced earlier in the chapter. Assume we are evaluating according to the counterpart model visually represented by Figure 3.2, with the signature being the idealised memory model in Section 3.1 and the counterpart relations acting as the identity on the objects sort, and on memory location sort as visually represented. Let's evaluate the formula $\phi_1 \equiv \exists x_{\tau_o}. \exists l_{\tau_l}. \rho(x) = l \wedge X(\rho(x) \neq l)$, intuitively it identifies worlds where there is at least one object that is moved in memory after one step of computation. Because this formula has no free variables it will be evaluated in the empty context, thus the only admissible substitution is the empty substitution, which we will denote with $\perp$. First, by virtue of the definition of the existential quantifier, we need to evaluate the conjunction in the empty context extended by the two quantified variables, $(\sigma, w) \vDash_{x,l} \rho(x) = l \wedge X(\rho(x) \neq l)$. The left-hand side of the conjunction is valid in all worlds when both variables are evaluated to be undefined or if both are defined and the location is exactly the one represented visually in the figure, e.g. in world w_0, with the assignement σ such that $\sigma(x) = A$, $\sigma(l) = 0$ it follows that $(\sigma, w) \vDash_{x,l} \rho(x) = l$. For the right-hand side we need to check the accessibility relation of the model. The pairs for which the formula $\rho(x) \neq l$ is valid are exactly the complement of the left-hand side, therefore we need to check which pair evolve due to the counterpart relations into one of those, for each possible accessible world. All the assignements where only one of the two variables are undefined, evolve via the counterpart relations into one in this set, however they are not interesting as they cannot satisfy the conjuction. There are also pair for which the subformula inside the next operator is valid in one trasition but not all of them, an example is (σ, w_2) with σ such that $\sigma(x) = C$, $\sigma(l) = 2$. After applying the counterpart relation $w_2 \overset{cr}{\rightsquigarrow} w_3$, $(cr \circ \sigma)(l) = 3$ that satisfies the inequality $\rho(C) \neq 3$, however with the counterpart relation $w_2 \overset{cr}{\rightsquigarrow} w_4$, $(cr \circ sigma)(l) = 2$ does not satisfy the inequality, thus $(\sigma, w_2) \not\models_{x,l} X(\rho(x) \neq l)$. An example of pair that, instead, makes the whole conjuction valid is (σ, w_3) with σ such that $\sigma(x) = E$,$\sigma(l) = 2$. Readers may notice that there is still the assignement always undefined for which equalities are always trivially sastisfied, however, the existential quantifier requires that assignments are extended with defined values, thus ultimately matching our intuition for the semantic of the formula. As noted before, the final pairs must be composed of empty substitution, therefore the only pairs for which the formula ϕ_1 is valid are $(\bot, w_1)$ and $(\bot, w_3)$.

If we wanted to recover the information about the specific object and/or location that moved with time, we need to evaluate open formulae, thus we need to remove one or both existential quantification. However, since there is no quantification that avoids undefined values that trivially satisfy equalities, we need a presence predicate, i.e. $\text{presence}(x_\tau) = \exists y_\tau . x = y$, which is satisfiable only if y is equal to x and y must be defined, therefore also x must be defined.

Chapter 4

Proof System

4.1 Axioms

Definition 4.1.1 (Axiomatic system). *The axiomatic system for FO-LTL contains the following axioms:*

1. $\phi_1 \to (\phi_2 \to \phi_1)$;
2. $(\phi_1 \to (\phi_2 \to \phi_3)) \to ((\phi_1 \to \phi_2) \to (\phi_1 \to \phi_3))$;
3. $(\neg\phi_1 \to \neg\phi_2) \to (\phi_2 \to \phi_1)$;
4. $\neg X\phi \to X\neg\phi$;
5. $X(\phi_1 \to \phi_2) \to (X\phi_1 \to X\phi_2)$;
6. $\phi \to X\phi$ *if* ϕ *is rigid;*
7. $\phi_1 \, U \phi_2 \leftrightarrow \phi_2 \vee (\phi_1 \wedge X(\phi_1 \, U \phi_2))$;
8. $\phi_1 \, U \phi_2 \to F\phi_2$;
9. $\phi[v/x] \to \exists x.\, \phi$, *with* x *free in* ϕ *and* v *a value in the domain;*
10. $x = x$;
11. $x = y \to (\phi \to \phi[y/x])$.

And the following induction rules:

mp $\phi_1, \phi_1 \to \phi_2 \vdash \phi_2$;

nex $\phi \vdash X\phi$;

ind $\phi_1 \to \phi_3 \vee (\phi_2 \wedge X\phi_1) \vdash \phi_1 \to \phi_2 \, U \phi_3$;

par $\phi_1 \to \phi_2 \vdash (\exists x.\, \phi_1) \to \phi_2$ *with* x *not free in* ϕ_2.

Lemma 4.1.1. *Let $\phi_1, \phi_2, \phi_3 \in \mathcal{F}_\Sigma$.*

1. $\vDash \phi_1 \rightarrow (\phi_2 \rightarrow \phi_1)$;

2. $\vDash (\phi_1 \rightarrow (\phi_2 \rightarrow \phi_3)) \rightarrow ((\phi_1 \rightarrow \phi_2) \rightarrow (\phi_1 \rightarrow \phi_3))$;

3. $\vDash (\neg\phi_1 \rightarrow \neg\phi_2) \rightarrow (\phi_2 \rightarrow \phi_1)$.

Proof. Let M be a counterpart model and a context Γ, a world $w \in W$ and an assignment σ:

1. From trivial applications of the semantic rules it follows that $(\sigma, w) \vDash_\Gamma \phi_1 \rightarrow (\phi_2 \rightarrow \phi_1) \Leftrightarrow (\sigma, w) \nvDash_\Gamma \phi_1$ or $(\sigma, w) \nvDash_\Gamma \phi_2$ or $(\sigma, w) \vDash_\Gamma \phi_1$ which is a tautology by the law of excluded middle.

2. By Equation (3.3) we show that the stronger formula $\vDash (\phi_1 \rightarrow (\phi_2 \rightarrow \phi_3)) \leftrightarrow ((\phi_1 \rightarrow \phi_2) \leftrightarrow (\phi_1 \rightarrow \phi_3))$. From the semantic rules:

$$\begin{aligned}
&(\sigma, w) \vDash_\Gamma \phi_1 \rightarrow (\phi_2 \rightarrow \phi_3)\\
&\quad \Leftrightarrow (\sigma, w) \nvDash_\Gamma \phi_1 \text{ or } (\sigma, w) \nvDash_\Gamma \phi_2 \text{ or } (\sigma, w) \vDash_\Gamma \phi_3\\
&\quad \Leftrightarrow ((\sigma, w) \vDash_\Gamma \phi_1 \text{ and } (\sigma, w) \nvDash_\Gamma \phi_2) \text{ or } (\sigma, w) \nvDash_\Gamma \phi_1 \text{ or } (\sigma, w) \vDash_\Gamma \phi_3\\
&\quad \Leftrightarrow (\sigma, w) \vDash_\Gamma (\phi_1 \rightarrow \phi_2) \rightarrow (\phi_1 \rightarrow \phi_3)
\end{aligned}$$

3. By Equation (3.3) we show the stronger formula $\vDash (\neg\phi_1 \rightarrow \neg\phi_2) \leftrightarrow (\phi_2 \rightarrow \phi_1)$. From the semantic rules:

$$\begin{aligned}
(\sigma, w) \vDash_\Gamma \neg\phi_1 \rightarrow \neg\phi_2 &\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi_1 \text{ or } (\sigma, w) \nvDash_\Gamma \phi_2\\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi_2 \rightarrow \phi_1
\end{aligned}$$

□

Lemma 4.1.2. *Let $\phi \in \mathcal{F}_\Sigma$, $\vDash \neg X\phi \rightarrow X\neg\phi$.*

Proof. Let M be a counterpart model and a context Γ, a world $w \in W$ and an assignment σ.

$$\begin{aligned}
(\sigma, w) \vDash_\Gamma \neg X\phi &\Leftrightarrow \neg\forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \vDash_\Gamma \phi\\
&\Rightarrow \neg\exists w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \vDash_\Gamma \phi\\
&\Leftrightarrow \forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, w') \nvDash_\Gamma \phi\\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma X\neg\phi
\end{aligned}$$

□

Lemma 4.1.3. *Let $\phi_1, \phi_2 \in \mathcal{F}_\Sigma$, $\vDash X(\phi_1 \rightarrow \phi_2) \rightarrow (X\phi_1 \rightarrow X\phi_2)$.*

Proof. Let M be a counterpart model and a context Γ, a world $w \in W$ and an assignment σ:

$$\begin{aligned}
&(\sigma, w) \vDash_\Gamma X(\phi_1 \to \phi_2) \\
&\quad \Leftrightarrow \forall w \overset{cr}{\rightsquigarrow} w'. (cr \circ \sigma, w') \vDash_\Gamma \phi_1 \text{ implies } (cr \circ \sigma, w') \vDash_\Gamma \phi_2 \\
&\quad \Rightarrow \forall w \overset{cr}{\rightsquigarrow} w'. (cr \circ \sigma, w') \vDash_\Gamma \phi_1 \text{ implies } \forall w \overset{cr}{\rightsquigarrow} w'. (cr \circ \sigma, w') \vDash_\Gamma \phi_2 \\
&\quad \Leftrightarrow (\sigma, w) \vDash_\Gamma X\phi_1 \to X\phi_2
\end{aligned}$$

□

Lemma 4.1.4. *Let $\phi \in \mathcal{F}_\Sigma$ be a rigid formula, $\vDash \phi \to X\phi$.*

Proof. Let M be a counterpart model and a context Γ, a world $w \in W$ and an assignment σ. Let $(\sigma, w) \vDash_\Gamma \phi$. By definition, $(\sigma, w) \vDash_\Gamma X\phi$ iff $\forall w \overset{cr}{\rightsquigarrow} w'. (cr \circ \sigma, w') \vDash_\Gamma \phi$. Let w' be counterpart related to w. Since ϕ is a rigid formula, $(cr \circ \sigma, w') \vDash_\Gamma \phi$ must hold. □

Lemma 4.1.5. *Let $\phi_1, \phi_2 \in \mathcal{F}_\Sigma$, $\vDash \phi_1 U \phi_2 \leftrightarrow \phi_2 \vee (\phi_1 \wedge X(\phi_1 U \phi_2))$.*

Proof. Follows trivially by the definition of the operators until and next. □

Lemma 4.1.6. *Let $\phi_1, \phi_2 \in \mathcal{F}_\Sigma$, $\vDash \phi_1 U \phi_2 \to F\phi_2$.*

Proof. Let M be a counterpart model and a context Γ, a world $w \in W$ and an assignment σ. By the definitions of the operators:

$$\begin{aligned}
(\sigma, w) \vDash_\Gamma \phi_1 U \phi_2 &\Leftrightarrow (\sigma, w) \vDash_\Gamma \phi_2 \text{ or } ((\sigma, w) \vDash_\Gamma \phi_1 \text{ and } (\sigma, w) \vDash_\Gamma X(\phi_1 U \phi_2)) \\
&\Rightarrow (\sigma, w) \vDash_\Gamma \phi_2 \text{ or } (\sigma, w) \vDash_\Gamma X(\phi_1 U \phi_2) \\
&\Rightarrow (\sigma, w) \vDash_\Gamma \phi_2 \text{ or } (\sigma, w) \vDash_\Gamma X(tt\, U \phi_2) \\
&\Leftrightarrow (\sigma, w) \vDash_\Gamma F\phi_2
\end{aligned}$$

□

Lemma 4.1.7. *Let $\phi \in \mathcal{F}_\Sigma$ with $x \in X$ free in ϕ, $\vDash \phi[v/x] \to \exists x. \phi$, if v is a value in the domain.*

Proof. Let M be a counterpart model and a context Γ, a world $w \in W$ and an assignment σ. By hypothesis $(\sigma, w) \vDash_\Gamma \phi[v/x]$, with $v \in d(w)$, then it holds that $(\text{ext}_x(\sigma, v), w) \vDash_{\Gamma,x} \phi$, therefore $(\sigma, w) \vDash_\Gamma \exists x. \phi$ holds. □

Lemma 4.1.8. *Let $\phi \in \mathcal{F}_\Sigma$ and $x, y \in X$,*

1. *$\vDash x = x$;*

2. *$\vDash x = y \to (\phi \to \phi[y/x])$.*

Proof. Let M be a counterpart model and a context Γ, a world $w \in W$ and an assignment σ.

1. By definition, $(\sigma, w) \vDash_\Gamma x = x$ holds if $\sigma(x) = \sigma(x)$, which is trivially true.

2. By assumption it holds that $(\sigma, w) \vDash_\Gamma x = y$, thus $\sigma(x) = \sigma(y)$. Assume $(\sigma, w) \vDash_\Gamma \phi$ holds, then by the congruence property of equality it must also hold $(\sigma, w) \vDash_\Gamma \phi[y/x]$.

□

Lemma 4.1.9. *If $F \vDash \phi_1$ and $F \vDash \phi_1 \to \phi_2$, then $F \vDash \phi_2$.*

Proof. Let M be a counterpart model that satisfies $\vDash_M \psi$ for every $\psi \in F$. By definition $(\sigma, w) \vDash_\Gamma \phi_1 \to \phi_2 \Leftrightarrow (\sigma, w) \vDash_\Gamma \phi_1$ implies $(\sigma, w) \vDash_\Gamma \phi_2$. By assumption it holds $(\sigma, w) \vDash_\Gamma \phi_1$, thus it must hold that $(\sigma, w) \vDash_\Gamma \phi_2$. □

Lemma 4.1.10. *If $F \vDash \phi$, then $F \vDash X\phi$ and $F \vDash G\phi$.*

Proof. Let M be a counterpart model that satisfies $\vDash_M \psi$ for every $\psi \in F$ and Γ. By assumption $(\sigma, w) \vDash_\Gamma \phi$ holds for every pair (σ, w). In particular given a pair (σ, w), it must also hold $(cr \circ \sigma, w') \vDash_\Gamma \phi$ for every $w \overset{cr}{\rightsquigarrow} w'$. □

Lemma 4.1.11. *If $F \vDash \phi_1 \to \phi_2$ and $F \vDash \phi_1 \to X\phi_1$, then $F \vDash \phi_1 \to G\phi_2$.*

Proof. Let M be a counterpart model that satisfies $\vDash_M \psi$ for every $\psi \in F$ and Γ. If $(\sigma, w) \nvDash_\Gamma \phi_1$ than the conclusion trivially holds. Assume, instead, $(\sigma, w) \vDash_\Gamma \phi_1$. By expanding the formula we need to show that $(\sigma', w') \vDash_\Gamma \phi_2$ holds for any pair (σ', w') such that is the composition of an arbitrary amount of counterpart relations, if $w \overset{cr_0}{\rightsquigarrow} \cdots \overset{cr_n}{\rightsquigarrow} w'$ then $\sigma' = cr_n \circ \cdots \circ cr_0 \circ \sigma$. However, by hypothesis $(\sigma, w) \vDash_\Gamma \phi_2$ and $(\sigma, w) \vDash_\Gamma X\phi_1$ are valid in every (σ, w) such that $(\sigma, w) \vDash_\Gamma \phi_1$ is valid, therefore ϕ_1 is valid in every sequence described before and thus ϕ_2 is valid in every sequence. □

Lemma 4.1.12. *If $F \vDash \phi_1 \to \phi_2$ and x not free in ϕ_2, then $F \vDash (\exists x.\, \phi_1) \to \phi_2$.*

Proof. Let M be a counterpart model that satisfies $\vDash_M \psi$ for every $\psi \in F$ and Γ. Assume $(\sigma, w) \nvDash_\Gamma (\exists x.\, \phi_1) \to \phi_2$ for some pair (σ, w). Then $(\sigma, w) \vDash_\Gamma \exists x.\, \phi_1$ and $(\sigma, w) \nvDash_\Gamma \phi_2$. Then there is a $v \in d(w)$ such that $(\text{ext}_x(\sigma, v), w) \vDash_{\Gamma,x} \phi_1$, by definition of existential quantifier. Since ϕ_2 does not contain x as a free variable, adding an evaluation for x does not change the evaluation of ϕ_2, i.e. $(\text{ext}_x(\sigma, v), w) \nvDash_{\Gamma,x} \phi_2$, therefore $(\text{ext}_x(\sigma, v), w) \nvDash_{\Gamma,x} \phi_1 \to \phi_2$, which is a contradiction to the hypothesis $\vDash_M \phi_1 \to \phi_2$. □

Theorem 4.1.13 (Soundness)**.** *Let $\phi \in \mathcal{F}_\Sigma$ and $F \subseteq \mathcal{F}_\Sigma$, if $F \vdash \phi$ then $F \vDash \phi$.*

Proof. By induction on the derivation of ϕ from F:

1. if ϕ is an axiom: $F \vDash \phi$ is proven by Lemma 4.1.1 for axioms 1,2,3; by Lemma 4.1.2 for axiom 4; by Lemma 4.1.3 for axiom 5; by Lemma 4.1.4 for axiom 6; by Lemma 4.1.5 for axiom 7; by Lemma 4.1.6 for axiom 8; by Lemma 4.1.7 for axiom 9; by Lemma 4.1.8 for axioms 10, 11;

2. if $\phi \in F$ then $F \vDash \phi$ holds trivially;

3. if ϕ is the conclusion of a (mp) rule with premises $F \vdash \psi$ and $F \vdash \psi \to \phi$: by induction hypothesis we have $F \vDash \psi$ and $F \vDash \psi \to \phi$, hence $F \vDash \phi$ follows by Lemma 4.1.9;

4. if ϕ is the conclusion of a (nex) rule with premises $F \vdash \psi$, thus $\phi \equiv X\psi$: by induction hypothesis we have $F \vDash \psi$, hence $F \vDash X\psi$ follows by Lemma 4.1.10;

5. if ϕ is the conclusion of a (ind) rule with premises $F \vdash \psi_1 \to \psi_2$ and $F \vdash \psi_1 \to X\psi_1$, thus $\phi \equiv \psi_1 \to G\psi_2$: by induction hypothesis we have $F \vDash \psi_1 \to \psi_2$ and $F \vDash \psi_1 \to X\psi_2$, hence $F \vDash \psi_1 \to G\psi_2$ follows by Lemma 4.1.11;

6. if ϕ is the conclusion of a (par) rule with premises $F \vdash \psi_1 \to \psi_2$, thus $\phi \equiv \exists x.\, \psi_1 \to \psi_2$ with x not free in ψ_2: by induction hypothesis we have $F \vDash \psi_1 \to \psi_2$, hence $F \vDash \exists x \psi_1 \to \psi_2$ follows by Lemma 4.1.12.

□

Example 4.1.1. *We show the derivation of some rules to show the capabilities of the deduction system. First a variant of the (ind) rule, $\phi \to X\phi \vdash \phi \to G\phi$, which we will call (ind1):*

(1)	$\phi \to X\phi$	*assumption*
(2)	$\phi \to \phi$	*tautology*
(3)	$\phi \to G\phi$	*(ind), (1), (2)*

Next we show the (alw) rule, $\phi \vdash G\phi$, which will be useful later for the deduction theorems.

(1)	ϕ	*assumption*
(2)	$X\phi$	*(nex), (1)*
(3)	$\phi \to X\phi$	*(axiom 1), (2)*
(4)	$\phi \to G\phi$	*(ind1), (3)*
(5)	$G\phi$	*(mp), (1), (4)*

Theorem 4.1.14. *Let $\phi_1, \phi_2 \in \mathcal{F}_\Sigma$ and $F \subseteq \mathcal{F}_\Sigma$. If $F \cup \{\, \phi_1 \,\} \vdash \phi_2$ and this deriviation of ϕ_2 does not contains application of the rule (par) for a variable occurring free in ϕ_1, then $F \vdash G\phi_1 \to \phi_2$.*

Proof. TODO □

Theorem 4.1.15. *Let $\phi_1, \phi_2 \in \mathcal{F}_\Sigma$ and $F \subseteq \mathcal{F}_\Sigma$. If $F \vdash G\phi_1 \to \phi_2$, then $F \cup \{ \phi_1 \} \vdash \phi_2$.*

Proof. Assume $F \vdash G\phi_1 \to \phi_2$, then $F \cup \{ \phi_1 \} \vdash G\phi_1 \to \phi_2$ also holds. By the (alw) rule (Example 4.1.1) and the trivial derivation $F \cup \{ \phi_1 \} \vdash \phi_1$ it follows that $F \cup \{ \phi_1 \} \vdash G\phi_1$, then by (mp) rule it follows $F \cup \{ \phi_1 \} \vdash \phi_2$. □

4.2 Missing axioms

Readers may have notice that some formulae that intuitively should be bidirectional are instead unidirectional in our model, e.g. the expansion of the forever operator (Equation (3.6)) and the axiom 4 (Definition 4.1.1). Moreover, another well studied formula that is often available in axiomatic systems for temporal logic, is the so-called Barcan formula, $X\exists x_\tau.\, \phi \to \exists x_\tau.\, X\phi$, which describes distributivity between first-order quantifiers and temporal operators. These issue are direct consequence of the increased generality of our model. In particular the first two examples can be solved by constraining the accessibility relation to a function, and thus each world has a single accessible world in its future, a standard assumption for the common linear temporal logic models. With this model we can transform axiom 4 into the formula $\neg X\phi \leftrightarrow X\neg\phi$, since the uniqueness of the transition solves the issue with the interplay between negation and universal quantification:

$$\begin{aligned}(\sigma, w) \vDash_\Gamma \neg X\phi &\Leftrightarrow (cr \circ \sigma, w') \nvDash_\Gamma \phi \text{ where } w \overset{cr}{\rightsquigarrow} w' \\ &\Leftrightarrow (\sigma, w) \vDash_\Gamma X\neg\phi\end{aligned}$$

This change, however, is still not sufficient for the Barcan formula.

$$\begin{aligned}(\sigma, w) \vDash_\Gamma X\exists x_\tau.\, \phi &\Leftrightarrow \forall w \overset{cr}{\rightsquigarrow} w'.\, \exists v \in d(w').\, (\mathrm{ext}_x(cr \circ \sigma, v), w') \vDash_{\Gamma,x} \phi \\ (\sigma, w) \vDash_\Gamma \exists x_\tau.\, X\phi &\Leftrightarrow \exists v \in d(w).\, \forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \mathrm{ext}_x(\sigma, v), w') \vDash_{\Gamma,x} \phi\end{aligned}$$

The existential operators on the two sides quantifies on different domains. Assume we have an accessibility relation where a world w is related only to a world w' where both domains are non-empty and the same algebra. We will see if the Barcan formula holds for the formula $\phi \equiv x_\tau = c$ where c is a constant of type τ in the algebra. Assume the two worlds are related by the counterpart relation cr that is the empty relation. The formula $X\exists x_\tau.\, x = c$ is trivially satisfied by extending the empty substitution, since the formula is closed, with the value c for the variable x. The other side is problematic, because the substitution can still be extended with the value c for the variable x, however the composition with the counterpart relation reduces the association to be undefined, which does not satisfy the equality. Systems that include some form of the Barcan formula as an axiom, require domains be constant while traversing accessible worlds, or that domains do not grow or shrink, allowing at least one side of the implication. Beside philosophical question regarding the value of such axiom, we will not extend our model to include such axiom, as one of the main drive of this alternative interpretation of temporal formulae is the necessity of modeling systems with creation and deallocation of resources.

4.3 Completeness and Decidability

The model presented in this work is not exempt from all the consideration on completeness available in literature for quantified temporal logic. Since the signature and the axioms for the theory of natural numbers can be encoded in our system, it follows that our logic is incomplete.

Example 4.3.1. *The theory of natural numbers can be encoded with the following signature* $\Sigma = (S_\Sigma, F_\Sigma)$ *where*

$$\begin{aligned}&S_\Sigma = \{\, \tau_{\mathbb{N}} \,\}\\&F_\Sigma = \{\, 0 : \tau_{\mathbb{N}}, s : \tau_{\mathbb{N}} \to \tau_{\mathbb{N}}, + : \tau_{\mathbb{N}} \times \tau_{\mathbb{N}} \to \tau_{\mathbb{N}}, \times : \tau_{\mathbb{N}} \times \tau_{\mathbb{N}} \to \tau_{\mathbb{N}} \,\}\end{aligned}$$

and by models where the following additional axioms are true:

- $s(x) \neq 0$
- $s(x) = s(y) \to x = y$
- $s(x) \neq 0 \to \exists y_{\tau_{\mathbb{N}}}.\, x = s(y)$
- $x + 0 = x$
- $x + s(y) = s(x + y)$
- $x \times 0 = 0$
- $x \times s(y) = (x \times y) + x$

and the following axiom for the scheme of induction, where ϕ *is a formula with* x *free:*

$$\phi[0/x] \wedge (\forall x_{\tau_{\mathbb{N}}}.\, \phi \to \phi[s(x)/x]) \to \forall x_{\tau_{\mathbb{N}}}.\, \phi$$

More articulated is, instead, the case for decidability. Previous results show that FO-LTL is not only undecidable in general, but in a multi-sorted context is not even semi-decidable. However, researches also showed that there are reasonable fragments that are, indeed, decidable but with serious restrictions. The main issue is the interaction between temporal operators and quantifiers, in particular the until operator. One approach is to limit the number of variables quantified within the scope of temporal operators, in particular restricting temporal formulae to be monodic, thanks to decidability results on monodic second-order logic. Another approach is to limit the nestedness of operators, for example we must avoid nested forever operators, or the usage of temporal operators inside the scope of a universal quantifier. Nonetheless, there are still issues with classical operators, and we must restrict also the usage of those to an arbitrary decidable fragment of first-order classical logic. Lastly, we can limit the domains of the models to finitary ones, therefore not only the set of worlds is finite, but also the algebras must be finite. These restriction are strict enough to obtain a decidable fragment of the second-order μ-calculus, which subsumes temporal logic.

A future objective is to investigate further this matter of decidability, as one of the objective of this model is to be useful, and even simplier, for model checking purposes.

Chapter 5

Future work

5.1 Second-order quantification

Definition 5.1.1 (Second-Order Linear Temporal Logic)**.** *Let Σ be a (multi-sorted) signature and X a denumerable set of first-order variables typed over S_Σ, and $\mathcal{X}$ a denumerable set of second-order variables typed over S_Σ. The set $\mathcal{F}_\Sigma$ of formulae for Second-Order Linear Temporal Logic is the set generated by the following grammar:*

$$\phi ::= tt \mid \epsilon_\tau \in \chi_\tau \mid \neg\phi \mid \phi \vee \phi \mid \exists x_\tau . \phi \mid \exists \chi_\tau . \phi \mid X\phi \mid \phi U \phi$$

where $\epsilon \in T(\Sigma_X)$, $\exists x_\tau$ ranges over first-order variables of sort $\tau \in S_\Sigma$, $\exists \chi_\tau$ ranges over second-order variables of sort $\tau \in S_\Sigma$, O is a unary operator which states that ϕ must hold at the next step and U is a binary operator which states that the first formula must hold until the second formula holds at some point in the current or next steps.

Other operators are derived the same as for First-Order Linear Temporal Logic, with the addition of equality, which in this case becomes itself a derived operator, $\epsilon_\tau = \eta_\tau \equiv \forall \chi_\tau . (\epsilon_\tau \in \chi_\tau \leftrightarrow \eta_\tau \in \chi_\tau)$.

Definition 5.1.2 (Semantic of SO-LTL)**.** *Let M be a counterpart model, Γ, Δ respectively a first-order and second-order context, (σ, w) a pair substitution-world in the context Γ, Δ. The validity of a formula-in-context ϕ, denoted*

$(\sigma, \xi, w) \models^{\Delta}_{\Gamma} \phi$, *is defined inductively as follows:*

$$
\begin{array}{ll}
(\sigma, \xi, w) \models^{\Delta}_{\Gamma} tt & \\
(\sigma, \xi, w) \models^{\Delta}_{\Gamma} \epsilon_{\tau} \in \chi_{\tau} & \mathit{iff}\ \sigma(\epsilon)\ \mathit{defined\ and}\ \sigma(\epsilon) \in \xi(\chi) \\
(\sigma, \xi, w) \models^{\Delta}_{\Gamma} \neg\phi & \mathit{iff}\ (\sigma, \xi, w) \not\models^{\Delta}_{\Gamma} \phi \\
(\sigma, \xi, w) \models^{\Delta}_{\Gamma} \phi_1 \vee \phi_2 & \mathit{iff}\ (\sigma, \xi, w) \models^{\Delta}_{\Gamma} \phi_1\ \mathit{or}\ (\sigma, \xi, w) \models^{\Delta}_{\Gamma} \phi_2 \\
(\sigma, \xi, w) \models^{\Delta}_{\Gamma} \exists x_{\tau}.\, \phi & \mathit{iff}\ \exists v \in d(w).\, (ext_x(\sigma, v), \xi, w) \models^{\Delta}_{\Gamma, x} \phi\ \mathit{with}\ x \notin \Gamma \\
(\sigma, \xi, w) \models^{\Delta}_{\Gamma} \exists \chi_{\tau}.\, \phi & \mathit{iff}\ \exists v \in 2^{d(w)}.\, (\sigma, ext_{\chi}(\xi, v), w) \models^{\Delta, \chi}_{\Gamma} \phi\ \mathit{with}\ \xi \notin \Delta \\
(\sigma, \xi, w) \models^{\Delta}_{\Gamma} X\phi & \mathit{iff}\ \forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, 2^{cr} \circ \xi, w') \models^{\Delta}_{\Gamma} \phi \\
(\sigma, \xi, w) \models^{\Delta}_{\Gamma} \phi_1 \, U \phi_2 & \mathit{iff}\ (\sigma, \xi, w) \models^{\Delta}_{\Gamma} \phi_2\ \mathit{or} \\
& \qquad ((\sigma, \xi, w) \models^{\Delta}_{\Gamma} \phi_1\ \mathit{and}\ \forall w \overset{cr}{\rightsquigarrow} w'.\, (cr \circ \sigma, 2^{cr} \circ \xi, w') \models^{\Delta}_{\Gamma} \phi_1 \, U \phi_2)
\end{array}
$$

Where 2^{cr} *is the lifting of the partial function* cr *to the power-set of the image, and* ext *defined as in Definition 3.4.1.*