On topological hierarchies of temporal properties

Christel Baier and Marta Kwiatkowska

Abstract. The classification of properties of concurrent programs into safety and liveness was first proposed by Lamport [**22**]. Since then several characterizations of hierarchies of properties have been given, see e.g. [**3, 20, 9, 21**]; this includes *syntactic* characterizations (in terms classes of formulas of logics such as the linear temporal logic) as well as *extensional* (as sets of computations in some abstract domain). The latter often admits a *topological* characterization with respect to the natural topologies of the domain of computations. We introduce a general notion of a linear time model of computation which consists of partial and completed computations satisfying certain axioms. The model is endowed with a natural topology. We show that the usual topologies on strings, Mazurkiewicz traces and pomsets arise as special cases. We then introduce a hierarchy of properties including safety, liveness, guarantee, response and persistence properties, and show that our definition subsumes the hierarchies of: Alpern & Schneider [**3**]; Chang, Manna & Pnueli [**9**]; and Kwiatkowska, Peled & Penczek [**21**]. Syntactic characterizations of the properties in the hierarchy in terms of temporal logic are also studied.

1. Introduction

The classification of properties of concurrent programs into *safety* and *liveness* was first proposed by Lamport [**22**]. According to the informal intuition introduced there, safety properties assert that "nothing bad happens", whereas liveness properties ensure that "something good will happen". Thus, a safety property is satisfied in a program if and only if at no point during its execution something "bad" happens. Examples of safety properties are: mutual exclusion (where the bad thing is two processes being in their critical sections at the same time), deadlock freedom (the bad thing is deadlock, i.e. a state in which no progress can be made) or partial correctness (where the bad thing is violating the postcondition assuming the execution started in a state satisfying the precondition). Safety properties are proved by means of arguments involving invariants; such arguments are usually too weak to guarantee that something will happen at all (e.g. partial correctness is no guarantee of termination).

In contrast to safety, proofs of liveness properties often employ well-founded induction. A liveness property states that at some point during the execution the program enters a desirable state. Termination is a typical liveness property; in the context of a mutual exclusion protocol it is the statement that a process trying to enter its critical section will *eventually* be allowed to enter. Some authors include also

1991 *Mathematics Subject Classification.* Subject Classification. Primary (68Q05, 68Q60); Secondary (03B70).

The second author was in part supported by EPSRC grant GR/K42028.

starvation freedom (every process ready to make progress infinitely often is allowed to do so infinitely often) within the class of liveness properties; here the desirable state (the process making progress) has to be entered *infinitely often*.

Apart from the underlying proof methodology, the distinction between safety and liveness can be made at other levels as well. This includes *syntactic* characterizations, i.e. classes of formulas that denote the given properties (e.g. safety is stated in terms of the 'always' modality in temporal logics, whereas liveness in terms of 'eventually'), see e.g. [**25, 9, 21**]; *automata-theoretic* characterizations (i.e. classes of automata which accept precisely the properties of a given class), see e.g. [**4, 25, 9**]; and *extensional* characterizations as certain sets of computations in some domain, see e.g. [**3, 20, 17, 25, 9, 21**]. The latter often admit the corresponding *topological* characterization for some topology on the domain of computations.

While all concerned agree that safety properties are the closed sets, disagreement between what precisely constitutes a liveness property in an abstract domain of computations persists. For example, Alpern & Schneider [**3**], working with the Cantor topology in the domain of infinite sequences of states, define liveness as the dense sets. In contrast, Chang, Manna & Pnueli [**9**], see also [**25, 10**], who work with the same domain but focus on the syntactic classes of properties expressed in Linear Time Temporal Logic (LTL), formulate a finer-grain hierarchy of four classes of properties (*safety*, *guarantee*, *response* and *persistence*) and show that they correspond to the two lower levels of the Borel hierarchy; they also show that the Alpern & Schneider characterization is orthogonal to their hierarchy. When considering a partial order semantic domain of computations, e.g. Mazurkiewicz traces, pomsets, etc, together with a partial order temporal logic, the picture complicates further, as the natural topologies of such domains (the relativised Scott topology) are coarser than, and need not coincide with, their metric topologies[1]; only certain aspects of the hierarchy of [**9**] generalise to this case. For example, in [**20, 17**], where the domain of Mazurkiewicz traces is used, liveness is defined as a G_δ-set, and fairness as a dense G_δ-set. In [**21**] this is developed further to a hierarchy of properties which reduces to the Chang, Manna & Pnueli hierarchy by considering a syntactic classification in the partial order temporal logic GISTL, together with a corresponding topological characterization in terms of the relativised Scott topology. While several aspects of [**9**] generalise to the partial order case, the automata-theoretic characterization does not, and only a subset of formulas of GISTL is considered.

This paper aims to define hierarchies of properties in terms of an abstract, axiomatically given, semantic domain of computations, which is a common generalisation of domains such as Mazurkiewicz traces [**26**], pomsets [**31**] or partial order executions [**15**]. The starting point is a *linear time* model $\overline{A}$, i.e. a semantic domain $\overline{A}$ subdivided into the 'finite elements' (the set $\mathcal{K}(A)$ of *partial* computations) and 'infinite elements' (the set A of *complete* computations). Partial computations can be thought of as finite execution fragments of complete computations; we suppose the existence of a mapping $x \mapsto \mathcal{K}(x)$

[1]This problem does not arise in the Cantor topology: it is simultaneously Hausdorff and the Scott topology of the finite and infinite sequences relativised to the maximal (infinite) sequences.

which assigns to each computation $x \in \overline{A}$ the set of its partial computations. The behaviour of a program is denoted by the *set* of its (complete) computations. The activities of the program include the usual execution of actions in sequence, in parallel, or through non-deterministic choice/branching. A (complete) computation of a program P denotes a possible behaviour of P which arises by resolving all the non-deterministic choices in advance: if there is a non-deterministic choice in a program then exactly one computation will record the specific choice made in that execution (i.e. computations do not contain branches). Note that if an interleaving abstraction is used, the non-deterministic choices arising from parallel execution of events (i.e. those made by a scheduler) will also be thus resolved; in the case of non-interleaving ('true concurrency') abstraction, this will apply only to *genuine* non-deterministic choices made in the program, i.e. those arising from a choice operator. Hence, each computation is a 'linear-time' structure, and can be viewed as a 'sequence' of actions or concurrently executed groups of actions. For convenience, we assume that all terminating computations are modelled by the complete (infinite) elements of $\overline{A}$; this can be achieved by extending each terminating computation with infinitely many occurrences of a special action which does not affect the state of the system.

Observe that our linear-time model $\overline{A}$ admits both interleaving and partial order (i.e. 'true concurrency') models, but does not faithfully represent branching behaviour. In *interleaving* models – where parallelism is reduced to non-determinism and sequential composition – no distinction is made between non-determinism arising from parallelism and that arising from explicit choice. In contrast to this, such distinctions can be made in *true concurrency* models; there, the execution of concurrent events can happen in any order or in parallel. Synchronization among processes is treated in the same way as explicit non-determinism. Hence, in interleaving models the linearization of a computation is uniquely determined, whereas in true concurrency models there may exist more than one linearization of a computation (which differ in the order of concurrent events).

A *property* can be viewed as a set of programs (namely exactly those programs for which the property holds). That is, a given property E (specified in any formalism) can be identified with the set of programs $Sat(E) = \{P : Me[P] \models E\}$. Here, we assume an appropriate semantics for the programs under consideration which assigns to each program P its meaning $Me[P]$ in a domain D where D is equipped with a satisfaction relation $\models$ for the properties.

For the purposes of this paper, we assume a fixed linear time model $\bar{A}$ and a meaning function Me which assigns to any program P a subset $Me[P]$ of 2^A. (I.e. $Me[P]$ are the *possible computations* of P.) The properties rely on the linear time view and make assertions about the possible computations of the programs (but not about the branching structure of the programs). Thus, we assume that for any property E there is a satisfaction relation $\models$ on A (the set of computations) where $x \models E$ asserts that the computation x meets the condition formalized by E. This satisfaction relation on A can be lifted to a satisfaction relation for the programs as follows:

$$P \models E \text{ iff } \forall x \in Me[P] \,.\, x \models E.$$

Thus, a program is considered to satisfy the property E iff E holds for all the possible computations of P. The requirement that all computations of P must satisfy E is motivated by the assumption that the decision as to which of the possible computations is executed is made by the environment, and not by the program itself.

When we identify the given property E with the set $\{x \in A \mid x \models E\}$ of all computations x where E holds, then any property can be viewed as a subset of A. Vice versa, any subset T of A can be viewed as a property that holds exactly for the programs P where $Me[P] \subseteq T$. For this reason we refer to any subset of A as a property.

Now, for an abstract model $\overline{A}$, following [**9, 21**] we define a hierarchy of four types of properties which can be verified by observing finite execution fragments: safety, guarantee, response and persistence properties. This is achieved by means of operators $\mathcal{A}$, $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{P}$ that assign to each finitary property F (i.e. $F \subseteq \mathcal{K}(A)$) a subset of A. For example, a safety property asserts that *all* finite approximations fulfill a certain finitary property F (i.e. a safety property consists of those computations x such that $\mathcal{K}(x) \subseteq F$), while a guarantee property states that all executions may pass a state which satisfies a certain finitary property F (i.e. a guarantee property consists of those computations x where $\mathcal{K}(x) \cap F \neq \emptyset$). Recurrence $\mathcal{R}$ and persistence $\mathcal{P}$ are defined similarly. Furthermore, we endow the model with natural topologies (order-theoretic and, in the presence of the length function, a metric) and give the corresponding, topological, characterizations of the classes of properties as described above. Finally, we compare our results with existing hierarchies defined for the domains of strings and Mazurkiewicz traces. We show that the result of [**3**] that safety, resp. liveness, properties are the closed, resp. dense, subsets carries over to arbitrary linear time models $\overline{A}$. In addition, the hierarchy of [**9**] corresponds to ours, while that of [**21**] does not w.r.t. the operators $\mathcal{R}$ and $\mathcal{P}$ unless the definition of response in [**21**] is appropriately strengthened.

Our definitions are general enough to admit the transfer of our results to other interleaving, as well as partial order, models, e.g. pomsets.

The paper is organized as follows. Section 2 presents the axiomatization of our model $\overline{A}$, and in Section 2.1 we show that the semantic domains of strings, Mazurkiewicz traces and pomsets are linear time models in our sense. Later we show that linear time models as defined here are closely related to algebraic dcpo's (directed-complete partial orders) and metric spaces (Section 2.2 and 2.3). In Section 3 we define the properties of: safety, liveness, guarantee, response and persistence, and give a topological characterization of each class of properties. In Section 4 we show how temporal logic can be used to describe properties in any linear time model. We intrepret the linear time logic LTL [**9**] over the 'interleaving models' (where the next step of a computation in a given state is uniquely determined, see Section 4.2) and the partial order temporal logic $ISTL^*$ [**15**] over 'true concurrency models' (where there might be several alternatives – arising from the way in which concurrent events are executed – to proceed in a given state, see Section 4.3).

2. Linear time models

In this section we define the notion of an abstract linear time model. We then show (Section 2.2) that linear time models in our sense can be endowed with a natural ordering and that (under additional assumptions) they form algebraic dcpo's. Furthermore, in Section 2.3 we consider the class of models equipped with a length function (which counts the number of atomic steps that a partial computation has to perform) and show that they can be endowed with a distance. We assume that the reader is familiar with the basic notions of domain theory, see e.g. [**2**], and metric spaces, see e.g. [**12**].

DEFINITION 2.1. A *linear time model* is a set $\overline{A}$ which is divided into disjoint subsets $\mathcal{K}(A)$ and A, together with a mapping $x \mapsto \mathcal{K}(x)$ which assigns to each $x \in \overline{A}$ a subset $\mathcal{K}(x)$ of $\mathcal{K}(A)$ such that:

(1) If $\xi \in \mathcal{K}(x)$ then $\mathcal{K}(\xi) \subseteq \mathcal{K}(x)$.
(2) For each $\xi \in \mathcal{K}(A)$ there is some $x \in A$ with $\xi \in \mathcal{K}(x)$.
(3) $\xi \in \mathcal{K}(\xi)$ for each $\xi \in \mathcal{K}(A)$.
(4) If $\mathcal{K}(x) = \mathcal{K}(y)$ then $x = y$.
(5) For each $x \in A$ there exists an x-*path*, i.e. a sequence $(\xi_n)_{n\geq 0}$ in $\mathcal{K}(x)$ such that

$$\mathcal{K}(\xi_0) \subset \mathcal{K}(\xi_1) \subset \mathcal{K}(\xi_2) \subset \ldots \quad \text{and} \quad \mathcal{K}(x) \;=\; \bigcup_{n\geq 0} \mathcal{K}(\xi_n).$$

The elements of $\mathcal{K}(A)$ should be thought of as *partial* computations (the 'finite' elements), and the elements of A as *complete* computations, or briefly computations (the 'infinite', or 'maximal' elements). Intuitively, the partial computations are initial fragments of the complete computations. Any complete computation describes a possible maximal behaviour of a program that might be obtained when all non-deterministic choices are resolved according to a fixed 'scheduler' (where a 'scheduler' can be viewed as an oracle that selects one of the alternatives at any branching point). The set $\mathcal{K}(x)$ is the set of all partial computations of x. Condition (1) states that if ξ is a partial computation of some computation x then all partial computations of ξ are partial computations of x. (2) ensures that only those partial computations are considered which are execution fragments of complete computations, or, in other words, which can be extended to a complete computation. (3) says that each partial computation approximates itself. (4) ensures that different computations can be distinguished by their partial computations. By condition (5) each complete computation can be approximated by its partial computations. Observe that by (2) and the requirement that $\mathcal{K}(x) \subseteq \mathcal{K}(A)$ we have the equality:

$$\mathcal{K}(A) \;=\; \bigcup \{\mathcal{K}(x) \mid x \in A\}.$$

Each x-path should be viewed as a fragment of a possible execution (linearization) of x: if $\xi_0, \xi_1, \xi_2, \ldots$ is an x-path we think of ξ_i as an intermediate state which the execution reaches after performing the partial computation described by ξ_i. It might be the case that there are additional intermediate states which are not represented by an element of the x-path. When considering the next step relation as in Section 4, which determines the possible next steps in an intermediate state, the *executions* of a computation x are defined to be those x-paths which obey the next step relation. In this case the x-paths are exactly the subsequences of the executions of x. The criterion for an x-path to approximate x (in the sense that $\bigcup \mathcal{K}(\xi_n) = \mathcal{K}(x)$) imposes a fairness (or maximal progress) constraint, since every partial computation must be subsumed by a partial computation of the x-path, i.e. each action which is performed in some execution is executed in every execution. This corresponds to the notion of an 'acceptable path' as in [**15**], 'maximality' of [**20**], or 'justice' in the sense of [**32**]. We extend the notion of an x-path to partial computations as follows. If $\xi \in \mathcal{K}(A)$ then a ξ-path is a sequence $(\xi_n)_{n\geq 0}$ in $\mathcal{K}(A)$ such that

$$\mathcal{K}(\xi_0) \subset \mathcal{K}(\xi_1) \subset \mathcal{K}(\xi_2) \subset \ldots \subset \mathcal{K}(\xi_n) = \mathcal{K}(\xi_{n+1}) = \ldots = \mathcal{K}(\xi).$$

DEFINITION 2.2. A linear time model *with an initial state* is a linear time model $\overline{A}$ satisfying:

(6) There exists $\perp \in \mathcal{K}(A)$ with $\mathcal{K}(\perp) = \{\perp\}$ and $\perp \in \mathcal{K}(x)$ for all $x \in \overline{A}$.

Because of conditions (3) and (4), the element $\perp$ in condition (6) is unique if it exists. $\perp$ can be interpreted as the partial computation which represents the state in which no action has been performed. By our intrepretation of the partial computations as the intermediate states of executions the element $\perp$ can be considered as the (common) initial state. (This explains the notion 'linear time models with an initial state').

2.1. Concrete examples of linear time models. Throughout the paper we illustrate the use of our framework by means of examples defined for the linear time models of *strings*, *Mazurkiewicz traces* and *pomsets*. In this section we recall basic definitions.

We suppose Σ to be a countable set of atomic actions including special symbols $\surd$ and δ which model termination and deadlock. Both $\surd$ and δ are assumed not to affect the state of the system, and which cannot be performed except when the system has reached its final state.

2.1.1. *The domain of strings.* By a string over the alphabet Σ we mean a (finite or infinite) sequence $s = \alpha_0\alpha_1\alpha_2\ldots$ of elements in Σ such that either the actions $\surd$ and δ do not occur in s or there is some $k \geq 0$ such that $\alpha_i \neq \surd, \delta$, for all $0 \leq i < k$ and either $\alpha_i = \surd$ for all $i \geq k$ or $\alpha_i = \delta$ for all $i \geq k$. Infinite strings containing $\surd$ represent successfully terminating computations, those containing δ model deadlocked computations, while those not containing any occurrence of $\surd$ and δ non-terminating computations. Finite prefices of a string represent its partial computations. Σ^* denotes the set of finite strings over Σ, Σ^ω the set of infinite strings over Σ. $\overline{A} = \Sigma^\infty$ is a linear time model in our sense; take $\mathcal{K}(A) = \Sigma^*$, $A = \Sigma^\omega$, and define $\mathcal{K}(x)$ to be the set of all finite prefices of x.

If $x \in \Sigma^\infty$ then $x[n]$ denotes the n-th prefix of x. (If the length of x is $\leq n$ put $x[n] = x$.) We assume Σ^∞ to be endowed with the usual distance

$$d(s,t) = \inf \left\{ \frac{1}{2^n} : s[n] = t[n] \right\}$$

and the usual prefix order (denoted by $\sqsubseteq$). Then Σ^∞ is a complete ultrametric space and an algebraic dcpo. The finite strings are the compact elements in Σ^∞ viewed as an algebraic dcpo. Σ^* is a dense subset of isolated elements in Σ^∞ when viewed as a metric space.

2.1.2. *Mazurkiewicz traces.* An independency relation on alphabet Σ is an irreflexive and symmetric binary relation $\iota \subseteq \Sigma \times \Sigma$ such that $\surd$ and δ are dependent on every action, i.e. $\neg(\ \alpha\ \iota\ \surd) \ \wedge\ \neg(\ \alpha\ \iota\ \delta\ )$ for all $\alpha \in \Sigma$. The pair (Σ, ι) is called a concurrent alphabet. The independency relation ι identifies those actions in the system which can happen concurrently; thus, if $\alpha\ \iota\ \beta$ then α, β are independent actions of two concurrent processes P and Q, i.e. P and Q cannot communicate via α and β. Let $\equiv^{\rm fin}$ [**26**] be the smallest equivalence relation on Σ^* such that

$$\text{whenever } s,t \in \Sigma^*,\ \alpha,\ \beta \in \Sigma,\ \alpha\ \iota\ \beta \text{ then } \ s\alpha\beta t\ \equiv^{\rm fin}\ s\beta\alpha t.$$

Define preorder $\preceq^{\rm tr}$ [**19**] on Σ^∞ by:

$$u \preceq^{\rm tr} v \ \text{ iff } \ \forall x \sqsubseteq u, \exists y \sqsubseteq v.(\exists z.x \sqsubseteq z \text{ and } z \equiv^{\rm fin} y)$$

where $\sqsubseteq$ is ordinary string prefix relation on strings. Now define *trace equivalence* $\equiv$ on Σ^∞ by:

$$u \equiv v \ \text{ iff } \ u \preceq^{\rm tr} v \text{ and } v \preceq^{\rm tr} u.$$

A *trace* is an equivalence class $[s]$ of a string $s \in \Sigma^\infty$. If s is (in)finite then $[s]$ is called (in)finite. The length of a trace is the length of one of its representatives. $[\Sigma^*]$ denotes the set of finite traces, while $[\Sigma^\omega]$ the set of infinite traces. Clearly, $[\Sigma^\infty] \ = \ [\Sigma^*] \cup [\Sigma^\omega]$ forms a linear time model in our sense; to see this take the finite traces as partial computations, the infinite traces as complete computations, and define the set $\mathcal{K}(x)$ of partial computations of x as consisting of all those finite traces $[s]$ where s is a prefix of some representative of x.

If x is a trace then $x^{(n)}$ denotes the set of finite traces $[s]$ where s is a prefix of some representative $t \in \Sigma^\infty$ of x and where the length of s is at most n. As in [**19, 18**], we consider the linear time model $[\Sigma^\infty]$ of traces in the following sense. We suppose $[\Sigma^\infty]$ to be equipped with the prefix order (by abuse of notation also denoted $\sqsubseteq$):

$$x \ \sqsubseteq \ y \quad \Longleftrightarrow \quad \exists\ s,t \in \Sigma^\infty \ \ s \sqsubseteq t,\ x = [s],\ y = [t]$$

Then $[\Sigma^\infty]$ is an algebraic dcpo (see e.g. [**19**]), with finite traces being the compact elements. Moreover, $[\Sigma^\infty]$ also has an associated metric d given by:

$$d(x,y) \ = \ \inf \left\{ \frac{1}{2^n} \ : \ x^{(n)} = y^{(n)} \right\}.$$

Then $[\Sigma^\infty]$ is a complete ultrametric space and $[\Sigma^*]$ is a dense subset of isolated elements (see [**18**]).

2.1.3. *Pomsets.* Pomsets (partially ordered multisets) were first introduced in [**31**]. Several variants of pomsets are known from the literature; here we use the notion of a pomset as a labelled prime event structure without conflicts in the sense of [**35**]. The underlying partial order is that of [**35**] restricted to pomsets, and the underlying metric is due to [**7**].

A *pomset* is a partially ordered set $(S, \leq)$ which is endowed with a labelling function $l : S \to \Sigma$ that maps the elements of S (called events) to actions and such that either all events are labelled with actions $\alpha \neq \surd, \delta$, or there exists an event $e \in S$ labelled by $\surd$ or δ such that:

- $e\uparrow = \{e' \in S : e \leq e'\}$ is totally ordered and $l(e) = l(e')$ for all events $e' \in e\uparrow$.
- No event $e' \in S$, $e' < e$, is labelled by $\surd$ or δ.

By a finite pomset we mean a pomset where the underlying partially ordered set is finite. Pomsets represent computations in the following sense. The execution of an event $e \in E$ means the execution of the associated action $l(e)$. If $e < e'$ (i.e. $e \leq e'$ and $e \neq e'$) then e must be executed before e'. If e, e' are independent events (i.e. neither $e \leq e'$ nor $e' \leq e$) then e and e' may be executed in parallel. In addition, we require that each event is reachable, i.e. for each $e \in E$ the set of predecessors of e is finite. We use two symbols, $\surd$ and δ, to denote (the event of) termination and deadlock respectively. (There exist several alternative treatments of deadlock and termination; our approach avoids the difficulty with sequential composition, see [**5**].) Infinite (non-terminating) computations are represented by infinite pomsets where no event is labelled by $\surd$ or δ. Terminating computations correspond to infinite pomsets where some events are labelled by $\surd$. Deadlocked computations are modelled by those infinite pomsets in which almost all events are labelled by δ. Partial computations are denoted by finite pomsets.

If $x = (S, \leq, l)$ is a pomset and $e \in S$ then the depth of e in x is given by:

$$depth_x(e) \ = \ \sup \ \{ \ n \ : \ \exists e_1, \ldots, e_n \in S \ \ e_1 < \ldots < e_n = e \ \}$$

If $S' \subseteq S$ is left-closed (i.e. whenever $e \in S'$ and $e' \leq e$ then $e' \in S'$) then we define

$$x\lceil S' \ = \ (\ S', \ \leq \cap \ S' \times S', \ l|S' \ ).$$

We put $x[n] \ = \ x \lceil S[n]$, where $S[n] = \{e \in S : depth_x(e) \leq n\}$. Pom^∞ denotes the set of all (finite and infinite) pomsets, and Pom^* the subset of finite pomsets. For convenience we assume that the set of events is contained in a fixed countable set *Events*[2]. Clearly, Pom^∞ forms a linear time model in our sense. To see this take Pom^* as the set of partial computations and $Pom^\omega \ = \ Pom^\infty \setminus Pom^*$ as the set of complete computations. If $x = (S, \leq, l)$ is a pomset then define $\mathcal{K}(x)$ to be all the pomsets $x\lceil S'$ where S' is a finite and left-closed subset of S. Pom^∞ can be endowed with the distance

$$d(x,y) \ = \ \inf \ \left\{ \ \frac{1}{2^n} \ : \ x[n] = y[n] \ \right\}$$

and the partial order $x \ \sqsubseteq \ y \quad \iff \quad \exists S \ \ x = y\lceil S$. Then Pom^∞ is a complete ultrametric space (see e.g. [**7**]) and an algebraic dcpo. The compact elements in Pom^∞, when viewed as an algebraic dcpo, are the finite pomsets. Since the underlying set *Events* is countable, the set S of events of a pomset is also countable. Hence, for each pomset x the set of finite pomsets ξ with $\xi \sqsubseteq x$ is countable (since the set of finite subsets of a countable set is countable). Pom^* is a dense subspace of isolated elements in Pom^∞ as a metric space.

[2]This assumption is essential to ensure that Pom^∞ is a set.

2.2. Linear time models and algebraic dcpo's. The relation of 'being a partial computation of' on linear time models induces a partial order in the following sense. Let $\overline{A}$ be a linear time model and define

$$x \sqsubseteq y \iff \mathcal{K}(x) \subseteq \mathcal{K}(y)$$

Then $\sqsubseteq$ is a partial order on $\overline{A}$ (called the *natural order* on $\overline{A}$). The partial computations of $\overline{A}$ are the compact elements. Conditions (1) and (5) of Definition 2.1 ensure that for each $x \in \overline{A}$ the set $\mathcal{K}(x)$ is an ideal (i.e. left-closed and directed), and x is the least upper bound of $\mathcal{K}(x)$. In linear time models with an initial state the unique element $\perp$ with $\perp \in \mathcal{K}(x)$ for all $x \in \overline{A}$ is the bottom element.

Definition 2.3. An *order-enriched* linear time model is a linear time model $\overline{A}$ with an initial state and which satisfies:

(7) For each directed and non-empty subset X of $\mathcal{K}(A)$ there exists $z \in \overline{A}$ with

$$\mathcal{K}(z) = \bigcup_{\xi \in X} \mathcal{K}(\xi).$$

The following two theorems show that order-enriched linear time models correspond to the algebraic dcpo's satisfying the condition that the set of compact elements below any element is countable.

Theorem 2.4. *Each order-enriched linear time model $\overline{A}$ is an algebraic dcpo. $\mathcal{K}(A)$ is the set of compact elements and $\perp$ the bottom element. $\mathcal{K}(x)$ is the set of compact elements $\xi \sqsubseteq x$. Whenever $X \subseteq \overline{A}$ is directed then the (unique) element $z \in \overline{A}$ with*

$$\mathcal{K}(z) = \bigcup_{x \in X} \mathcal{K}(x)$$

is the least upper bound of X.

Proof. We only show that for each directed subset X of $\overline{A}$ the least upper bound $\bigsqcup X$ exists. The remaining statements are easy verifications. Let X be a directed subset of X and let $K = \bigcup_{x \in X} \mathcal{K}(x)$. Then K is a directed subset of $\mathcal{K}(A)$ (this is because X and the sets $\mathcal{K}(x)$ are directed). By condition (7) there exists $z \in \overline{A}$ with $\mathcal{K}(z) = \bigcup_{\xi \in K} \mathcal{K}(\xi)$. It is easy to see that then $\mathcal{K}(z) = K$. Hence, $\mathcal{K}(x) \subseteq \mathcal{K}(z)$ for all $x \in X$, i.e. z is an upper bound of X. If $y \in \overline{A}$ is also an upper bound of X then $\mathcal{K}(x) \subseteq \mathcal{K}(y)$ for all $x \in X$. Thus, $\mathcal{K}(z) = K \subseteq \mathcal{K}(y)$ and therefore $z \sqsubseteq y$. Hence, $z = \bigsqcup X$. □

Theorem 2.5. *If $\overline{D}$ is an algebraic dcpo such that*

(i) *For every $x \in \overline{D}$ the set of compact elements ξ with $\xi \sqsubseteq x$ is countable.*
(ii) *For every compact element ξ there exists a non-compact element $x \in \overline{D}$ with $\xi \sqsubseteq x$.*

Then $\overline{D}$ is an order-enriched linear time model where the natural order on $\overline{D}$ as a linear time model agrees with the original partial order on $\overline{D}$. The finite elements are the compact elements in $\overline{D}$. The set $\mathcal{K}(x)$ is the set of compact elements $\xi \sqsubseteq x$.

PROOF. We define $\mathcal{K}(D)$ to be the set of compact elements of $\overline{D}$ and $D = \overline{D} \setminus \mathcal{K}(D)$. Then it is easy to see that conditions (1), (3), (4), (6) and (7) are satisfied. Condition (2) follows by (ii), condition (5) by (i) and the fact that $\bigsqcup \mathcal{K}(x) = x$. □

EXAMPLE 2.6. The algebraic dcpo's Σ^∞, $[\Sigma^\infty]$ and Pom^∞ satisfy the conditions (i) and (ii) of Theorem 2.5, and hence all are order-enriched linear time models.

If $\overline{D}$ is an algebraic dcpo satisfying condition (i) of Theorem 2.5 then $\overline{D}$ can be embedded into an order-enriched linear time model $\overline{A}$ such that for each $x \in \overline{D}$ the set $\mathcal{K}(x)$ is the set of compact elements $\xi \in \overline{D}$ with $\xi \sqsubseteq x$. Notice that in $\overline{D}$ condition (2) might be violated. In order to fulfill condition (2), for each compact element $\xi \in \overline{D}$ which does not have a non-compact upper bound in $\overline{D}$ we create new elements (ξ, n) where $n \in I\!N_0 \cup \{\infty\}$, and we extend the original partial order $\sqsubseteq$ on $\overline{D}$ as follows: $\sqsubseteq'$ is the smallest partial order on $\overline{A}$ (which contains $\overline{D}$ and the new elements (ξ, n)) which satisfies

$$\xi \ \sqsubset' \ (\xi, 0) \ \sqsubset' \ (\xi, 1) \ \sqsubset' \ \ldots \ \sqsubset' \ (\xi, \infty).$$

Then the elements (ξ, n), $n \in I\!N_0$, are compact in $\overline{A}$ and (ξ, ∞) is a non-compact upper bound of ξ in $\overline{A}$.

COROLLARY 2.7. Each ω-algebraic cpo $\overline{D}$ can be embedded into an order-enriched linear time model $\overline{A}$ such that for each $x \in \overline{D}$ the set $\mathcal{K}(x)$ is the set of compact elements $\xi \in \overline{D}$, $\xi \sqsubseteq x$.

2.3. Linear time models and metric spaces. If partial computations consist of executions of finitely many atomic actions then we have a natural notion of a length on $\mathcal{K}(A)$: the length of a partial computation ξ is the maximum number of atomic steps which an execution of ξ needs. This notion is similar to that defined for a partial order in [**24, 6**]. We show that linear time models with a suitable length function are metric spaces.

DEFINITION 2.8. A *length function* on a linear time model $\overline{A}$ with initial state $\perp$ is a function

$$|\cdot| : \mathcal{K}(A) \to I\!N_0$$

such that:

(8) $|\perp| = 0$ and $\xi \in \mathcal{K}(\eta)$ implies $|\xi| \ \leq \ |\eta|$.
(9) For each $x \in \overline{A}$ there exists an x-path $(\xi_n)_{n \geq 0}$ with $|\xi_n| = \min \{ \, |x|, \, n \, \}$ for all $n \geq 0$. Here we put $|x| = \infty$ if $x \in A$.

Condition (8) ensures that partial computations of η do not require more steps than η itself. Condition (9) asserts that each computation x can be approximated by a length-increasing sequence (ξ_n) of partial computations of x, where the length of ξ_n is exactly n or $|x|$. Given a length function on a linear time model $\overline{A}$ we put

$$\mathcal{K}_n(A) \ = \ \{ \ \xi \in \mathcal{K}(A) \ : \ |\xi| \leq n \ \}$$

and $\mathcal{K}_n(x) = \mathcal{K}_n(A) \cap \mathcal{K}(x)$. Then

$$d(x,y) = \inf\left\{ \frac{1}{2^n} : \mathcal{K}_n(x) = \mathcal{K}_n(y) \right\}$$

is an ultrametric on $\overline{A}$. Note that condition (4) ensures that $d(x,y) = 0$ implies $x = y$.

The reader should note that we stipulate the existence of a length function on the domain of computations as defined in [**24, 6**]. It is an interesting open question to find sufficient and/or necessary conditions on domains which would guarantee the existence of such a length function.

NOTATION 2.9. If (M,d) is a metric space, $x \in M$ and $r > 0$ then $B(x,r)$ denotes the open ball with centre x and radius r. $\overline{B}(x,r)$ denotes the closure of $B(x,r)$, i.e.

$$\overline{B}(x,r) = \{ y \in M : d(x,y) \leq r \}.$$

Since the induced distance can only be given values 0 or $1/2^n$ for some natural number n, for all elements x of a linear time model with a length function we have that $B(x,r) = \overline{B}(x, 1/2^n)$ where $n = 0$ if $r \geq 1$ and n is the unique natural number satisfying $1/2^n < r \leq 1/2^{n-1}$ otherwise.

LEMMA 2.10. *Let $\overline{A}$ be a linear time model with a length function. Then $\mathcal{K}(A)$ is a dense subset of $\overline{A}$ and all elements of $\mathcal{K}(A)$ are isolated in $\overline{A}$ when $\overline{A}$ is viewed as a metric space.*

PROOF. Easy verification. □

In general, the induced metric space of a linear time model with a length function is not complete. In order to ensure completeness the following condition is needed:

(10) *If $(x_n)_{n\geq 0}$ is a sequence in $\overline{A}$ with $\mathcal{K}_n(x_n) = \mathcal{K}_n(x_{n+1})$ for all $n \geq 0$ then there exists $x \in \overline{A}$ with $\mathcal{K}(x) = \mathcal{K}_n(x_n)$ for all $n \geq 0$.*

EXAMPLE 2.11. The linear time models Σ^∞ and $[\Sigma^\infty]$ can be endowed with the length function which assigns to each finite string/trace its usual length. On Pom^∞ the function

$$|\cdot| : Pom^* \to I\!N_0, \quad |\xi| = \max \{ \, depth_\xi(e) : e \text{ is an event in } \xi \, \}$$

is a length function. In all three cases the ultrametric induced by the underlying length function coincides with the usual metric (cf. Section 2.1).

THEOREM 2.12. *Let $\overline{M}$ be an ultrametric space, M_0 a subspace of $\overline{M}$ and $|\cdot| : M_0 \to I\!N_0$ a function such that:*

(i) *For all $\xi \in M_0$, $|\xi| = n$, there exists $x \in \overline{M} \setminus M_0$ with $d(\xi, x) \leq 1/2^n$.*
(ii) *For each $x \in \overline{M}$ with either $x \notin M_0$ or $|x| \geq n$ there exists a unique element $x[n] \in M_0$ with*

$$|x[n]| = n \quad \textit{and} \quad d(\, x[n], \, x) \leq \frac{1}{2^n}.$$

We put $\xi[n] = \xi$ *if* $\xi \in M_0$, $|\xi| < n$ *and* $|x| = \infty$ *if* $x \in \overline{M} \setminus M_0$. *Then* $\overline{M}$ *is a linear time model with* $\mathcal{K}(M) = M_0$ *and*

$$\mathcal{K}(x) = \{ x[n] : n \geq 0 \}.$$

In addition, we have for all $x, y \in \overline{M}$*:*

(a) $x[n]$ *is the unique element* $\xi \in \mathcal{K}(x)$ *with* $|\xi| = \min\{|x|, n\}$.

(b) $d(x,y) \leq 1/2^n$ *iff* $x[n] = y[n]$

(c) $(x[m])[n] = (x[n])[m] = x[n]$ *for all* $0 \leq n \leq m$

(d) $|x| = \sup \{ |\xi| : \xi \in \mathcal{K}(x) \}$

Proof. Let $M = \overline{M} \setminus M_0$. (a), (b), (c) and (d) are easy verifications. Conditions (1) and (3) are satisfied because of (c). Condition (2) follows by (i), conditions (5) and (9) by (ii), condition (8) by (d).

To see that (4) holds, let x, y be such that $\mathcal{K}(x) = \mathcal{K}(y)$, then $x[n] = y[n]$ for all $n \geq 0$. This is because of (a). Hence, $x = \lim x[n] = \lim y[n] = y$. $\square$

Definition 2.13. A linear time model with a length function satisfying the conditions (i) and (ii) of Theorem 2.12 is called *metric-enriched*.

Example 2.14. The linear time model Σ^∞ and the linear time model of pomsets $x \in Pom^\infty$ such that $x[n] \in Pom^*$ for all $n \geq 0$ are metric-enriched.

In Example 2.14 it is essential that we deal with pomsets whose n-cuts $x[n]$ are finite ('finitely approximable' pomsets in the sense of [**14**]), as otherwise condition (ii) of Theorem 2.12 would be violated since if x is a pomset where $x[n]$ is infinite then there is no pomset $\xi \in Pom^*$ with $|\xi| = n$ and $d(x, \xi) \leq 1/2^n$. Condition (ii) of Theorem 2.12 is also violated when we deal with the linear time model of Mazurkiewicz traces with a non-empty independency relation. For instance, for the trace x induced by the string $s = \alpha\beta\gamma\gamma\gamma\ldots$ with $\alpha \,\iota\, \beta$ there does not exist a finite trace ξ of length 1 with $d(x, \xi) = 1/2$. This is because $\mathcal{K}_1(x)$ contains the traces $[\alpha]$ and $[\beta]$, and the distance $d(x, [\alpha]) = d(x, [\beta]) = 1$. An alternative length function for traces can be found by embedding traces into pomsets; with this length function traces form a metric-enriched model.

3. Defining properties on linear time models

In this section we give general definitions of *safety*, *guarantee*, *response*, *persistence* and *liveness* properties. Following [**3**] we define liveness properties to be those properties $T \subseteq A$ such that each partial computation $\xi \in \mathcal{K}(A)$ has a complete computation $x \in T$ which is above it in the ordering. As in [**9, 21**], we define safety, guarantee, response and persistence properties by operators $\mathcal{A}$, $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{P}$ acting on sets of partial computations (the *finitary* properties). When applied to the linear time model Σ^∞, our definitions agree with those of [**3, 9**]; some early work due to Landweber, see e.g. [**34**], introduces a similar topological hierarchy for accepting conditions of automata on infinite sequences. We show that the hierarchy and the topological characterizations stated in [**3, 9**] carry over to arbitrary linear time models.

For simplicity assume from now on that $\overline{A}$ is a fixed linear time model. If $F \subseteq \mathcal{K}(A)$ then F is called a *finitary property.* Following [**9**] we put:

$$\begin{aligned}
\mathcal{A}(F) &= \{\ x \in A\ :\ \mathcal{K}(x) \subseteq F\ \}\\
\mathcal{E}(F) &= \{\ x \in A\ :\ \mathcal{K}(x) \cap F \neq \emptyset\ \}\\
\mathcal{R}(F) &= \{\ x \in A\ :\ \text{there exists an } x\text{-path } (\xi_n) \text{ with } \xi_n \in F \text{ for all } n\ \}\\
\mathcal{P}(F) &= \{\ x \in A\ :\ \text{if } (\xi_n) \text{ is an } x\text{-path then } \xi_n \in F \text{ for almost all } n\ \}
\end{aligned}$$

and

$$\mathcal{A}_{\text{fin}}(F) = \{\xi \in \mathcal{K}(A) : \mathcal{K}(\xi) \subseteq F\}, \quad \mathcal{E}_{\text{fin}}(F) = \{\xi \in \mathcal{K}(A) : \mathcal{K}(\xi) \cap F \neq \emptyset\}.$$

$\mathcal{A}(F)$, $\mathcal{E}(F)$, $\mathcal{P}(F)$ and $\mathcal{R}(F)$ respectively denote the sets of all the complete computations x such that: *all* partial computations of x are contained in F; *some* partial computation of x belongs to F; whenever (ξ_n) is an x-path then *almost all* (ξ_n) belong to F; and there exists an x-path (ξ_n) such that *infinitely many* (ξ_n) belong to F.

The above definitions of $\mathcal{A}$, $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{P}$ correspond precisely to those of [**9**] when applied to the linear time model of strings. In the linear time model of traces, our definitions of the operators $\mathcal{A}$ and $\mathcal{E}$ coincide with those of [**21**], but the definitions of $\mathcal{R}$ and $\mathcal{P}$ do not. In [**21**], where a partial order temporal logic is used, $\mathcal{R}(F)$ is defined as the set of all the infinite traces whose infinitely many prefices belong to F, and $\mathcal{P}(F)$ as the set of all the infinite traces whose almost all prefices belong to F. This is not compatible with our definition since we require an x-path to approximate x (in the sense that x is the least upper bound of a x-path w.r.t. the natural order). For instance, let F be the set of all finite traces

$$\underbrace{\alpha\alpha\ldots\alpha}_{n}, \quad n \geq 0.$$

Let $\alpha\ \iota\ \beta$ and let $x = [\beta\alpha\alpha\alpha\ldots]$. Then x belongs to $\mathcal{R}(F)$ in the sense of [**21**], but $x \notin \mathcal{R}(F)$ according to the definition in this paper. The operators $\mathcal{R}$ and $\mathcal{P}$ as defined above admit an alternative definition shown below.

LEMMA 3.1. *Let $F \subseteq \mathcal{K}(A)$. Then:*

(a) *$x \in \mathcal{R}(F)$ iff for every $\xi \in \mathcal{K}(x)$ there exists $\xi' \in \mathcal{K}(x) \cap F$ with $\xi \in \mathcal{K}(\xi')$.*

(b) *$x \in \mathcal{P}(F)$ iff there exists $\xi \in \mathcal{K}(x)$ such that $\xi' \in \mathcal{K}(x) \ \wedge\ \xi \in \mathcal{K}(\xi') \implies \xi' \in F$.*

PROOF. (a) If $x \in \mathcal{R}(F)$ then there exists an x-path (ξ_n) in F. Let $\xi \in \mathcal{K}(x)$. Then $\xi \in \mathcal{K}(x) = \bigcup_{n \geq 0} \mathcal{K}(\xi_n)$. Hence, there exists $n \geq 0$ such that $\xi \in \mathcal{K}(\xi_n)$.

Assume that the condition on the right hand side of (a) is fulfilled. Let (η_n) be an x-path. For each $n \geq 0$ there exists $\xi_n \in \mathcal{K}(x) \cap F$ with $\eta_n \in \mathcal{K}(\xi_n)$. Then a suitable subsequence of (ξ_n) is an x-path in F.

(b) Follows by the duality of $\mathcal{R}$ and $\mathcal{P}$ and part (a).

□

DEFINITION 3.2. A *safety*, *guarantee*, *response*, resp. *persistence* property is any property of the form $\mathcal{A}(F)$, $\mathcal{E}(F)$, $\mathcal{R}(F)$, resp. $\mathcal{P}(F)$, where F is a finitary property. A subset T of A is called a *liveness property* iff for each $\xi \in \mathcal{K}(A)$ there exists $x \in T$ such that $\xi \in \mathcal{K}(x)$. An *obligation property* is a property of the form

$$T \;=\; \bigcap_{1 \le i \le m} (S_i \cup G_i)$$

where $S_1, \ldots, S_m$ are safety properties and $G_1, \ldots, G_m$ are guarantee properties. A *reactivity property* is a property of the form

$$T \;=\; \bigcap_{1 \le i \le m} (R_i \cap P_i)$$

where $R_1, \ldots, R_m$ are response properties and $P_1, \ldots, P_m$ are persistence properties.

The hierarchy of safety, guarantee, response, persistence, obligation and reactivity properties, and the duality of $\mathcal{A}$ and $\mathcal{E}$, resp. $\mathcal{R}$ and $\mathcal{P}$, as stated in [**9**] carry over to our general framework:

THEOREM 3.3. *Persistence properties subsume safety properties, guarantee properties are special kinds of response properties.*

$$\mathcal{A}(F) \;=\; \mathcal{P}(\,\mathcal{A}_{\mathrm{fin}}(F)\,), \quad \mathcal{E}(F) \;=\; \mathcal{R}(\,\mathcal{E}_{\mathrm{fin}}(F)\,)$$

Guarantee properties are complements of safety properties, while response properties complements of persistence properties.

$$A \setminus \mathcal{A}(F) \;=\; \mathcal{E}(\,\mathcal{K}(A) \setminus F\,), \quad A \setminus \mathcal{P}(F) \;=\; \mathcal{R}(\,\mathcal{K}(A) \setminus F\,)$$

Obligation properties are special kinds of response and persistence properties.

$$\bigcap_{1 \le i \le m} (\mathcal{A}(F_i) \cup \mathcal{E}(F_i')) \;=\; \mathcal{R}\left(\bigcap_{1 \le i \le m} H_i\right) \;=\; \mathcal{P}\left(\bigcap_{1 \le i \le m} H_i\right)$$

where

$$H_i \;=\; \mathcal{A}_{\mathrm{fin}}(F_i) \cup \mathcal{E}_{\mathrm{fin}}(F_i').$$

Reactivity properties subsume response and persistence properties, obligation properties subsume safety and guarantee properties.

PROOF. The duality of $\mathcal{A}$ and $\mathcal{E}$ resp. $\mathcal{R}$ and $\mathcal{P}$ is an easy verification. It is clear that each safety or guarantee property is an obligation property since:

$$\mathcal{A}(F) \;=\; \mathcal{A}(F) \cup \mathcal{E}(\emptyset), \quad \mathcal{E}(F') \;=\; \mathcal{A}(\emptyset) \cup \mathcal{E}(F')$$

and that each response or persistence property is a reactivity property:

$$\mathcal{R}(F) \;=\; \mathcal{R}(F) \cup \mathcal{P}(\emptyset), \quad \mathcal{P}(F') \;=\; \mathcal{R}(\emptyset) \cup \mathcal{P}(F')$$

The equation $\mathcal{E}(F) = \mathcal{R}(\mathcal{E}_{\text{fin}}(F))$ follows by $\mathcal{A}(F) = \mathcal{P}(\mathcal{A}_{\text{fin}}(F))$ and the duality of $\mathcal{A}$ and $\mathcal{E}$, resp. $\mathcal{R}$ and $\mathcal{P}$.

(1) We show $\mathcal{A}(F) = \mathcal{P}(\mathcal{A}_{\text{fin}}(F))$. If $x \in \mathcal{A}(F)$ then $\mathcal{K}(x) \subseteq F$. Hence, for all $\xi \in \mathcal{K}(x)$, $\mathcal{K}(\xi) \subseteq \mathcal{K}(x) \subseteq F$. Therefore, $\xi \in \mathcal{A}_{\text{fin}}(F)$. We conclude that $\mathcal{K}(x) \subseteq \mathcal{A}_{\text{fin}}(F)$, and hence $x \in \mathcal{P}(\mathcal{A}_{\text{fin}}(F))$.

If $x \in \mathcal{P}(\mathcal{A}_{\text{fin}}(F))$ then (by Lemma 3.1(b)) there exists $\xi \in \mathcal{K}(x)$ such that whenever $\xi' \in \mathcal{K}(x)$, $\xi \in \mathcal{K}(\xi')$, then $\xi' \in \mathcal{A}_{\text{fin}}(F)$. Let $\eta \in \mathcal{K}(x)$. There exists $\xi' \in \mathcal{K}(x)$ with $\xi, \eta \in \mathcal{K}(\xi')$. Then $\xi' \in \mathcal{A}_{\text{fin}}(F)$ and therefore $\eta \in \mathcal{K}(\xi') \subseteq F$. It follows that $x \in \mathcal{A}(F)$.

(2) Next we prove that if $F, F' \subseteq \mathcal{K}(A)$ then $\mathcal{A}(F) \cup \mathcal{E}(F') = \mathcal{R}(H) = \mathcal{P}(H)$ where $H = \mathcal{A}_{\text{fin}}(F) \cup \mathcal{E}_{\text{fin}}(F')$.

If $x \in \mathcal{A}(F)$ then by part (1) of this proof $x \in \mathcal{P}(\mathcal{A}_{\text{fin}}(F)) \subseteq \mathcal{R}(\mathcal{A}_{\text{fin}}(F))$. Hence, $x \in \mathcal{P}(\mathcal{A}_{\text{fin}}(F)) \subseteq \mathcal{P}(H)$ and $x \in \mathcal{R}(\mathcal{A}_{\text{fin}}(F)) \subseteq \mathcal{R}(H)$. If $x \in \mathcal{E}(F')$ then there is some $\xi \in \mathcal{K}(x) \cap F'$. Let (ξ_n) be an x-path. Then $\xi \in \mathcal{K}(\xi_{n_0})$ for some n_0. Then $\xi \in \mathcal{K}(\xi_n)$ for all $n \geq n_0$. Hence, $\xi_n \in \mathcal{E}_{\text{fin}}(F')$ for almost all n. Therefore, $x \in \mathcal{P}(\mathcal{E}_{\text{fin}}(F'))$. We conclude $x \in \mathcal{R}(\mathcal{E}_{\text{fin}}(F')) \subseteq \mathcal{R}(H)$ and $x \in \mathcal{P}(H)$.

Let $x \in \mathcal{R}(H)$. (Since $\mathcal{P}(H) \subseteq \mathcal{R}(H)$ this includes the case $x \in \mathcal{P}(H)$.) Then there is an x-path (ξ_n) in H.

- If there exists $n \geq 0$ such that $\xi_n \in \mathcal{E}_{\text{fin}}(F')$ then $\mathcal{K}(\xi_n) \cap F' \neq \emptyset$. Since $\mathcal{K}(\xi_n) \subseteq \mathcal{K}(x)$, $\mathcal{K}(x) \cap F' \neq \emptyset$. Thus $x \in \mathcal{E}(F')$.
- If $\xi_n \notin \mathcal{E}_{\text{fin}}(F')$ for all n then $\xi_n \in \mathcal{A}_{\text{fin}}(F)$ for all n. Hence, $\mathcal{K}(\xi_n) \subseteq F$ for all n. Therefore, $\mathcal{K}(x) = \bigcup_{n \geq 0} \mathcal{K}(\xi_n) \subseteq F$. It follows that $x \in \mathcal{A}(F)$.

(3) To show $\mathcal{P}\left(\bigcap_{1 \leq i \leq m} F_i\right) = \bigcap_{1 \leq i \leq m} \mathcal{P}(F_i)$ observe that $\subseteq$ is clear. If $x \in \bigcap \mathcal{P}(F_i)$ then there exists $\xi_i \in \mathcal{K}(x)$ such that, for all $\xi \in \mathcal{K}(x)$, $\xi_i \in \mathcal{K}(\xi)$ implies $\xi \in F_i$. Since $\mathcal{K}(x)$ is directed there is some $\eta \in \mathcal{K}(x)$ with $\xi_1, \ldots, \xi_m \in \mathcal{K}(\eta)$. Hence, whenever $\xi \in \mathcal{K}(x)$, $\eta \in \mathcal{K}(\xi)$ then $\xi_i \in \mathcal{K}(\xi)$, and therefore $\xi \in \bigcap F_i$. We conclude $x \in \mathcal{P}\left(\bigcap F_i\right)$.

(4) $\bigcap_{1 \leq i \leq m} (\mathcal{A}(F_i) \cup \mathcal{E}(F'_i)) = \mathcal{R}\left(\bigcap_{1 \leq i \leq m} H_i\right) = \mathcal{P}\left(\bigcap_{1 \leq i \leq m} H_i\right)$ where $H_i = \mathcal{A}_{\text{fin}}(F_i) \cup \mathcal{E}_{\text{fin}}(F'_i)$.

By part (2) of this proof, $\mathcal{A}(F_i) \cup \mathcal{E}(F'_i) = \mathcal{R}(H_i) = \mathcal{P}(H_i)$. and part (3) we have that

$$\bigcap (\mathcal{A}(F_i) \cup \mathcal{E}(F'_i)) = \bigcap \mathcal{P}(H_i) = \mathcal{P}\left(\bigcap H_i\right)$$

and

$$\bigcap \mathcal{R}(H_i) = \bigcap \mathcal{P}(H_i) = \mathcal{P}\left(\bigcap H_i\right) \subseteq \mathcal{R}\left(\bigcap H_i\right) \subseteq \bigcap \mathcal{R}(H_i).$$

Therefore

$$\mathcal{R}\left(\bigcap H_i\right) = \bigcap \mathcal{R}(H_i) = \bigcap \mathcal{P}(H_i) = \bigcap (\mathcal{A}(F_i) \cup \mathcal{E}(F'_i)).$$

□

It is an open question whether liveness properties are special kinds of reactivity properties.

Liveness does not subsume safety or guarantee properties. This is because $\emptyset$ is a safety and a guarantee property, but not a liveness property. In general, neither response nor persistence properties subsume liveness properties, as can be seen from the example below.

EXAMPLE 3.4. In the linear time model Σ^∞ the set

$$T_1 = \{ x \in \Sigma^\omega : \alpha^\omega \text{ is a suffix of } x \}$$

is a liveness property (eventually always α), but not a response property. The set

$$T_2 = \{ x \in \Sigma^\omega : \alpha^\omega \text{ is not a suffix of } x \}$$

is a liveness property (always eventually not α), but not a persistence property. (Here α^ω stands for the infinite string $\alpha\alpha\alpha\ldots$.)

PROOF. It is clear that T_1 and T_2 are liveness properties. Suppose $T_1 = \mathcal{R}(F)$ for some $F \subseteq \Sigma^*$. Then $x_1 = \beta\alpha^\omega \in T_1$. Hence, there exists $n_1 \geq 1$ such that $\xi_1 = \beta\alpha^{n_1} \in F$. Then $x_2 = \beta\alpha^{n_1}\beta\alpha^\omega \in T_1$. Thus, there exists $n_2 \geq 1$ with $\xi_2 = \beta\alpha^{n_1}\beta\alpha^{n_2} \in F$. Proceeding in this way we get a sequence of natural numbers $n_k \geq 1$ such that

$$\xi_k = \beta\alpha^{n_1}\beta\alpha^{n_2}\ldots\beta\alpha^{n_k} \in F.$$

Let $x = \lim \xi_k$ (i.e. x is the unique infinite string where ξ_k are prefices of x). Then $x \in \mathcal{R}(F)$ (since (ξ_k) is an x-path in F), but $x \notin T_1$. Contradiction.

The argument for T_2 is similar. □

Part (a) of the following lemma shows that our definition of safety properties is a generalization of the definition of safety properties in the sense of [**3**].

LEMMA 3.5. *Let $T \subseteq A$. Then:*

(a) *T is safety property iff for each $x \in A \setminus T$ there exists some $\xi \in \mathcal{K}(x)$ such that whenever $y \in A$, $\xi \in \mathcal{K}(y)$ then $y \notin T$.*

(b) *T is a guarantee property iff for each $x \in T$ there exists some $\xi \in \mathcal{K}(x)$ such that whenever $y \in A$, $\xi \in \mathcal{K}(y)$, then $y \in T$.*

NOTATION 3.6. If $\xi \in \mathcal{K}(A)$ we put $U(\xi) = \{x \in \overline{A} : \xi \in \mathcal{K}(x) \}$.

It is easy to see that, because of condition (5), whenever $\zeta, \eta \in \mathcal{K}(x)$ for some $x \in \overline{A}$ then there exists $\xi \in \mathcal{K}(x)$ with $\zeta, \eta \in \mathcal{K}(\xi)$. In particular, whenever $x \in U(\zeta) \cap U(\eta)$ then $x \in U(\xi) \subseteq \mathcal{U}(\zeta) \cap U(\eta)$ for some $\xi \in \mathcal{K}(A)$. Hence, the sets $U(\xi)$, $\xi \in \mathcal{K}(A)$, form a topological basis. In what follows we assume $\overline{A}$ to be equipped with the topology induced by the basis $U(\xi)$, $\xi \in \mathcal{K}(A)$, and that A is endowed with the subspace topology. In part (b) of Lemma 3.7 we show that in order-enriched linear time models the topology induced by the basis $U(\xi)$, $\xi \in \mathcal{K}(A)$, is the Scott-topology on $\overline{A}$ considered as an algebraic dcpo. In general, the topology on $\overline{A}$ is not T_2. This is because whenever $\mathcal{K}(x) \subseteq \mathcal{K}(y)$ then each neighbourhood of x contains y. In particular, a converging sequence might have more than one limit. We write $x = \lim x_n$ to denote that x is one of the limits of the sequence (x_n). Since the topology on $\overline{A}$ is not T_2, we cannot expect that in the case where a linear time model $\overline{A}$ is equipped with a length function the metric on $\overline{A}$ induces the topology on $\overline{A}$. In part (c) of Lemma 3.7 we show that if $\overline{A}$ is metric-enriched the metric on A induces the (subspace-)topology on A. Part (c) of Lemma 3.7 can be applied to the metric-enriched linear time model Σ^∞ or the metric-enriched linear time model of pomsets $x \in Pom^\infty$ where $x[n]$ is finite for all n.

LEMMA 3.7. *Let $\overline{A}$ be a linear time model.*

(a) *Whenever $(x_n)_{n\geq 0}$ is a sequence in $\overline{A}$ such that there exists an x-path $(\xi_n)_{n\geq 0}$ with $\xi_n \in \mathcal{K}(x_n)$ for all $n \geq 0$ then $x = \lim x_n$. In particular, each x-path converges to x.*

(b) *If $\overline{A}$ is order-enriched then the topology on $\overline{A}$ agrees with the Scott topology on $\overline{A}$ as an algebraic dcpo.*

(c) *If $\overline{A}$ is metric-enriched in the sense of Definition 2.13 then the metric on A induces the topology on A.*

PROOF. (b) is an easy verification using the fact that $U(\xi) = \xi\uparrow$.

(a) Let (x_n) be a sequence in $\overline{A}$ and (ξ_n) an x-path with $\xi_n \in \mathcal{K}(x_n)$. Let U be an open neighbourhood of x. Then there exists $\eta_1, \dots, \eta_n \in \mathcal{K}(A)$ such that

$$x \in \bigcup_{1\leq j\leq n} U(\eta_j) \subseteq U.$$

Then $\eta_j \in \mathcal{K}(x) = \bigcup \mathcal{K}(\xi_i)$. Since $\mathcal{K}(\xi_i) \subset \mathcal{K}(\xi_{i+1})$ there exists $k \geq 0$ such that $\eta_j \in \mathcal{K}(\xi_k)$, $j = 1, \dots, n$. Then for all $i \geq k$ and $j = 1, \dots, n$: $\eta_j \in \mathcal{K}(\xi_k) \subseteq \mathcal{K}(\xi_i) \subseteq \mathcal{K}(x_i)$. Hence, for all $i \geq k$:

$$x_i \in \bigcup_{1\leq j\leq n} U(\eta_j) \subseteq U.$$

Thus, we conclude that $x = \lim x_i$.

(c) We first show that if $\xi \in \mathcal{K}(A)$, $|\xi| = n$, then $U(\xi) = B(\xi, 1/2^{n-1})$. Let $x \in U(\xi)$. Then $\xi \in \mathcal{K}(x)$ and, by Theorem 2.12(a) and (c), $\xi[n] = \xi = x[n]$ and hence $d(x,\xi) \leq 1/2^n$. Therefore: $x \in \overline{B}(\xi, 1/2^n) = B(\xi, 1/2^{n-1})$. If $x \in B(\xi, 1/2^{n-1})$ then $d(x,\xi) \leq 1/2^n$. Hence, $\xi \in \mathcal{K}(\xi) = \mathcal{K}_n(\xi) = \mathcal{K}_n(x) \subseteq \mathcal{K}(x)$. and $x \in U(\xi)$ follows as required.

Next we show that if $x \in A$ and $r > 0$ then $B(x,r) = U(x[n])$ where n is the natural number with $n = 0$ if $r \geq 1$ and $1/2^n < r \leq 1/2^{n-1}$ otherwise. If $y \in B(x,r)$ then $d(x,y) < r \leq 1/2^{n-1}$. Hence, $d(x,y) \leq 1/2^n$. Then $\mathcal{K}_n(x) = \mathcal{K}_n(y)$, and thus $x[n] = y[n] \in \mathcal{K}(y)$, from which we immediately obtain $y \in U(x[n])$.

If $y \in U(x[n])$ then $x[n] \in \mathcal{K}(y)$. Since $x \in A$ we have $|x[n]| = n$. Hence $x[n] = y[n]$ and therefore $d(x,y) \leq \frac{1}{2^n} < r$. Thus, $y \in B(x,r)$ as required.

□

COROLLARY 3.8.

- The topology on $\overline{A}$ is coarser than the topology on $\overline{A}$ induced by the metric. This is because every basis open $U(\xi)$ can be written as $B(x, 1/2^{n-1})$ where $\xi = x[n]$, $x \in A$. Note that $B(x[n], 1/2^{n-1}) = B(x, 1/2^{n-1})$ and that all elements $\xi \in \mathcal{K}(A)$ are of the form $\xi = x[n]$ for some $x \in A$ and $n \geq 0$.
- For order-enriched models the topology on A is the relative Scott topology. A is the subspace of maximal (and also non-compact) elements of $\overline{A}$.

The following theorem generalizes the topological characterizations of safety, guarantee and liveness properties as established in [**3, 9, 21**].

THEOREM 3.9. *Let $\overline{A}$ be a linear time model and $T \subseteq A$. Then:*

(a) *T is a safety property iff T is closed.*

(b) *T is a guarantee property iff T is open.*

(c) *T is a liveness property iff T is dense.*

PROOF. (a) Let T be closed. Then we show $T = \mathcal{A}(F)$ where $F = \bigcup_{x \in T} \mathcal{K}(x)$. If $x \in T$ then $\mathcal{K}(x) \subseteq F$. Hence, $x \in \mathcal{A}(F)$. Let $x \in \mathcal{A}(F)$, then $\mathcal{K}(x) \subseteq F$. Let (ξ_n) be a x-path. Since $\xi_n \in F$ and, by definition of F, there exists a sequence (x_n) in T with $\xi_n \in \mathcal{K}(x_n)$ we have by Lemma 3.7(a) that $x = \lim x_n$. Since T is closed and since $x_n \in T$ for all $n \geq 0$ we conclude $x \in T$.

Let $T = \mathcal{A}(F)$, $x \in A$ and (x_n) a sequence in T such that x is a limit of (x_n). We have to show that $x \in T$. Let $\xi \in \mathcal{K}(x)$. We have to show that $\xi \in F$. Since $U(\xi)$ is an open neighbourhood of x, and since (x_n) converges to x, there exists $m \geq 0$ such that $x_m \in U(\xi)$. Thus, $\xi \in \mathcal{K}(x_m)$, and since $x_m \in T = \mathcal{A}(F)$ we obtain $\xi \in F$.

(b) follows by (a) and the duality of $\mathcal{A}$ and $\mathcal{E}$.

(c) Let T be a liveness property. We have to show that whenever U is an open subset of $\overline{A}$ with $U \cap A \neq \emptyset$ then $U \cap T \neq \emptyset$. It is sufficient to consider the case that U is basic open, i.e. $U = \bigcap_{1 \leq i \leq n} U(\xi_i)$ for some $\xi, \ldots, \xi_n \in \mathcal{K}(A)$. Since $U \cap A \neq \emptyset$ there exists $x \in U \cap A$. Then there exists $\xi \in \mathcal{K}(x)$ with $\xi_i \in \mathcal{K}(\xi)$, $i = 1, \ldots, n$. Hence, $U(\xi) \subseteq U$. Since T is a liveness property there exists $y \in T$ with $\xi \in \mathcal{K}(y)$. Then $y \in U(\xi)$, and thus $y \in T \cap U$.

Let T be dense in A. If $\xi \in \mathcal{K}(A)$ then $U(\xi)$ is open, and because of condition (3) there exists $x \in A \cap U(\xi)$. Hence, $A \cap U(\xi) \neq \emptyset$. Since T is dense in A there is some $y \in T \cap U(\xi)$, from which it follows that $y \in T$ and $\xi \in \mathcal{K}(y)$.

□

Next we consider response and persistence properties. In general, we do not obtain the results of [**9, 21**] which characterize response and persistence properties as the G_δ, resp. F_σ-sets, unless the model satisfies stronger conditions (see Lemma 3.10); in the latter case the hierarchy as in [**9**] can be obtained. It is worth noting that the additional conditions are satisfied by the linear model of strings, but not by traces and pomsets. As a counter-example, consider $\alpha\ \iota\ \beta$ and the trace $x = [(\alpha\beta)^\infty]$, then there exists an infinite subset $[\alpha^*]$ of $\mathcal{K}(x)$ which does not contain an infinite x-path. The case for pomsets is similar, except that a partial solution can be obtained by modifying the definition of the map $\mathcal{K}(x)$ to assign to an infinite pomset x the set of its n-cuts $x[n]$, instead of assigning all finite prefices of x. The results of [**21**] are more problematic as the definitions of $\mathcal{R}(F)$ and $\mathcal{P}(F)$ differ from ours.

Recall that F_σ-sets are countable unions of closed sets, G_δ-sets countable intersection of open sets. The following lemma characterizing response and persistence can be stated. Its proof makes use of condition (i), which does not hold for traces and pomsets. It is an open question whether the lemma holds without (i).

LEMMA 3.10. *Let $\overline{A}$ be a linear time model such that:*

(i) *If $x \in A$ and X is an infinite subset of $\mathcal{K}(x)$ then X contains an x-path.*

(ii) *For each $\xi \in \mathcal{K}(A)$ the set $\mathcal{K}(\xi)$ is finite.*

Then for each subset T of A:

(a) *T is a response property iff T is a G_δ-set.*

(b) *T is a persistence property iff T is a F_σ-set.*

PROOF. (b) follows by (a) and the duality of $\mathcal{R}$ and $\mathcal{P}$. We show (a). Let $T = \mathcal{R}(F)$. We define F_k to be the set consisting of all $\xi \in F$ such that there exist $\xi_1, \ldots, \xi_k \in F \cap \mathcal{K}(\xi)$ with

$$\mathcal{K}(\xi_1) \subset \mathcal{K}(\xi_2) \subset \ldots \subset \mathcal{K}(\xi_k) \subseteq \mathcal{K}(\xi).$$

We prove that $T = \bigcap \mathcal{E}(F_k)$. Note that because of Theorem 3.9 (b) the sets $\mathcal{E}(F_k)$ are open, hence $\bigcap \mathcal{E}(F_k)$ is a G_δ-set.

- If $x \in T$ then there exists an x-path $(\xi_k)_{k\geq 1}$ such that $\xi_k \in F$ for all k. Then $\xi_k \in F_k$ and therefore $x \in \mathcal{E}(F_k)$.
- If $x \in \bigcap \mathcal{E}(F_k)$ then for each $k \geq 1$ there exists $\xi_k \in F_k \cap \mathcal{K}(x)$. By definition of F_k the cardinality of $\mathcal{K}(\xi_k)$ is at least k. Since by assumption (ii) the cardinality of $\mathcal{K}(\xi_k)$ is finite, the set $\{\xi_i : i \geq 1\}$ is infinite. By assumption (i) there exists an x-path in $\{\xi_i : i \geq 1\}$. Since $\xi_k \in F_k \subseteq F$ all elements of the x-path belong to F. Therefore, $x \in \mathcal{R}(F)$.

If $T = \bigcap G_k$, where G_k are open sets in A, we may assume that $G_1 \supseteq G_2 \supseteq \ldots$. Otherwise we deal with $G'_k = G_1 \cap \ldots \cap G_k$. Because of Theorem 3.9 (b) there exists subsets F_k of $\mathcal{K}(A)$ such that $G_k = \mathcal{E}(F_k)$. W.l.o.g. $F_1 \supseteq F_2 \supseteq \ldots$ (otherwise we deal with $F'_k = \bigcup_{i\geq k} F_i$). Let H_k be the set consisting of all $\xi \in F_k$ such that:

$$\text{whenever } \xi' \in \mathcal{K}(\xi),\ \xi' \neq \xi, \text{ then } \xi' \notin F_k.$$

Let $H = \bigcup H_k$ and $F = \bigcap F_k$. We show $T = \mathcal{E}(F) \cup \mathcal{R}(H)$ Note that $\mathcal{E}(F) = \mathcal{R}(\mathcal{E}_{\text{fin}}(F))$ and hence $\mathcal{E}(F) \cup \mathcal{R}(H) = \mathcal{R}(F')$ where $F' = \mathcal{E}_{\text{fin}}(F) \cup H$.

- If $x \in T$ then for each $k \geq 1$ there exists $\xi_k \in F_k$. Since $\mathcal{K}(\xi_k)$ is finite (by assumption (ii)) we may assume that ξ_k is minimal, i.e. whenever $\xi \in \mathcal{K}(\xi_k)$, $\xi_k \neq \xi'$, then $\xi' \notin F_k$.

  Case 1: The set $\{\xi_k : k \geq 1\}$ is finite.

  Then there exists $\xi \in \{\xi_k : k \geq 1\}$ with $\xi = \xi_k$ for infinitely many k. Hence, $\xi \in F_k$ for infinitely many k. Since $F_1 \supseteq F_2 \supseteq \ldots$ we get $\xi \in F_k$ for all k, i.e. $\xi \in F$ and $x \in \mathcal{E}(F)$.

  Case 2: The set $\{\xi_k : k \geq 1\}$ is infinite.

  Because of the minimality of ξ_k we have that $\xi_k \in H_k \subseteq H$. Let (η_k) be an x-path in $\{\xi_k : k \geq 1\}$ (which exists because of assumption (i)). Then $\eta_k \in H$ for all k, and thus $x \in \mathcal{R}(H)$.
- If $x \in \mathcal{E}(F)$ then $\xi \in F$ for some $\xi \in \mathcal{K}(x)$. Hence, $\xi \in F_k$ for all k and therefore $x \in \bigcup \mathcal{E}(F_k) = T$. If $x \in \mathcal{R}(H)$ then there exists an x-path (ξ_k) in H. Then $\xi_k \in H_{m_k}$ for some $m_k \geq 1$. Since $\mathcal{K}(\xi_k) \subset \mathcal{K}(\xi_{k+1})$ we have:

$$\xi_k \in \mathcal{K}(\xi_{k+1}) \quad \text{and} \quad \xi_k \neq \xi_{k+1}$$

  By definition of H_{m_k} we get:

$$\xi_{k+1} \in F_{m_{k+1}} \quad \text{and} \quad \xi_k \notin F_{m_{k+1}}$$

  Since $F_1 \supseteq F_2 \supseteq \ldots$ we get: $m_1 < m_2 < \ldots$ and therefore $m_k \geq k$. Hence, $\xi_k \in F_{m_k} \subseteq F_k$ for all k. Therefore, $x \in \bigcup \mathcal{E}(F_k) = T$.

$\square$

In [**21**] the respective definitions of $\mathcal{R}$ and $\mathcal{P}$ differ from ours, i.e.

$$\mathcal{R}(F) = \{x \in A : \exists(\xi_n) : \mathcal{K}(\xi_1) \subset \mathcal{K}(\xi_2) \subset \ldots \mathcal{K}(x) \text{ and } \xi_n \in F\ \}.$$

One can show that under the assumptions (ii) and

(i') Each infinite subset of $\mathcal{K}(x)$ contains an increasing sequence

the proof of Lemma 3.10 carries over to the modified definitions of $\mathcal{R}$ and $\mathcal{P}$ if we work with increasing sequences in $\mathcal{K}(x)$ instead of x-paths. Note that under the above conditions the domain of Mazurkiewicz traces becomes finitely concurrent.

Lemma 3.11. *Let F_1, F_2 be finitary properties. Then:*

(a) $\mathcal{A}(F_1) \cap \mathcal{A}(F_2) = \mathcal{A}(F_1 \cap F_2)$ *and* $\mathcal{A}(F_1) \cup \mathcal{A}(F_2) = \mathcal{A}\left(\mathcal{A}_{\text{fin}}(F_1) \cup \mathcal{A}_{\text{fin}}(F_2)\right)$

(b) $\mathcal{E}(F_1) \cup \mathcal{E}(F_2) = \mathcal{E}(F_1 \cup F_2)$ *and* $\mathcal{E}(F_1) \cap \mathcal{E}(F_2) = \mathcal{E}\left(\mathcal{E}_{\text{fin}}(F_1) \cap \mathcal{E}_{\text{fin}}(F_2)\right)$

(c) $\mathcal{R}(F_1 \cup F_2) = \mathcal{R}(F_1 \cup F_2)$

(d) $\mathcal{P}(F_1 \cap F_2) = \mathcal{P}(F_1) \cap \mathcal{P}(F_2)$

It is an open question whether $\mathcal{R}$ and $\mathcal{P}$ are closed under intersection and union respectively. However, under the assumptions (i) and (ii) of Lemma 3.10 we obtain that $\mathcal{R}(F_1 \cap F_2) = \mathcal{R}(F)$, where F is the set of $\eta \in F_2$ such that there exists $\xi \in F_1 \cap \mathcal{K}(\eta)$ satisfying: whenever $\eta' \in F_2 \cap \mathcal{K}(\eta)$ and $\xi \in \mathcal{K}(\eta')$ then $\eta' = \eta$. The duality of $\mathcal{R}$ and $\mathcal{P}$ then yields the closedness of $\mathcal{P}$ under union.

4. Temporal logic and linear time models

In this section we show how linear or branching time temporal formulas can be interpreted over arbitrary linear time models with an initial state $\perp$ and a next step relation $\to$. If $x \in A$ then we interpret the elements of $\mathcal{K}(x)$ as possible intermediate states which an execution of x may pass. If an execution of x reaches the intermediate state ξ then the possible next steps are those which lead to an intermediate state $\xi' \in \mathcal{K}(x)$ such that $\xi \to \xi'$. We associate $\to$ with a mapping which assigns to each step $\xi \to \xi'$ a multiset $act(\xi, \xi')$ of all those actions which are executed in the step from ξ to ξ'. If $act(\xi, \xi')$ contains more than one action then the actions in $act(\xi, \xi')$ are executed in parallel. An execution (called *observation*) of a (complete) computation is a sequence $(\xi_n)_{n \geq 0}$ which

- starts in the initial state $\xi_0 = \perp$
- successively performs $\to$-steps, i.e. $\xi_n \to \xi_{n+1}$
- approximates x, i.e. (ξ_n) is an x-path.

Observe that the next step relation allows the simultanous execution of independent actions; this should be compared with maximal progress.

In the case where the next step relation ensures the existence of a unique execution, i.e. where the next step of a computation x in an intermediate state ξ is uniquely determined, we consider the linear time logic LTL which is closely related to the linear time logic of [**23, 9**]. When the next step relation allows more than one possible next steps, we use a partial order logic $ISTL^*$.

In section 4.1 we formalize the conditions which a suitable next step relation on a linear time model has to fulfill. Section 4.2 introduces our interpretation of the linear time logic LTL over linear time models with a determinisitic next step relation. We show that our interpretation of LTL over Σ^∞ and a suitable next step relation coincides with those of [**23, 25**]. In section 4.3 we extend the interpretation of the logic $ISTL^*$ [**15, 29**] to arbitrary linear time models with a next step relation. The reader is cautioned to note that our intepretation of $ISTL^*$ is non-standard.

Our approach applied to the model $[\Sigma^\infty]$ of traces differs from that of [**21**] as we require an execution of a computation x to approximate x. This imposes fairness in the sense of maximality, see e.g. [**16, 20**]. If we consider the linear time model of partial order executions we get the interpretation of $ISTL$ à la [**15**].

4.1. Linear time models with a next step relation. Let Σ be a countable set of atomic actions. In what follows $\overline{A}$ is a linear time model with an initial state $\perp$. By a multiset of atomic actions we mean a function $\kappa : \Sigma \to I\!N_0$. If $\kappa(\alpha) = n \geq 1$ then n copies of α are contained in κ. If $\kappa(\alpha) = 0$ then α does not occur in κ. We write $\alpha \in \kappa$ to denote that α appears at least once in κ, i.e. $\kappa(\alpha) \geq 1$. Union of multisets is defined to be addition.

DEFINITION 4.1. A *next step relation* on $\overline{A}$ is a pair $(\rightarrow, act)$ consisting of a binary relation $\rightarrow$ on $\mathcal{K}(A)$ and a mapping act which assigns to each pair (ξ, ξ') of finite elements with $\xi \rightarrow \xi'$ a multiset $act(\xi, \xi')$ of atomic actions such that the following conditions (i) - (iv) are fulfilled:

(i) If $\xi \rightarrow \eta$ then $\xi \sqsubset \eta$.

(ii) If $\xi_1 \rightarrow \xi_2$ and $\xi_1 \sqsubset \eta \sqsubset \xi_2$ then $\xi_1 \rightarrow \eta$ and $\eta \rightarrow \xi_2$.

(iii) If $\xi \sqsubset \eta$ then there exists $k \geq 2$ and $\xi_1, \xi_2, \ldots, \xi_k \in \mathcal{K}(A)$ such that

$$\xi = \xi_1 \ \rightarrow \ \xi_2 \ \rightarrow \ \ldots \ \rightarrow \ \xi_{k-1} \ \rightarrow \ \xi_k = \eta.$$

(iv) Whenever $\xi = \xi_1 \rightarrow \xi_2 \rightarrow \ldots \rightarrow \xi_k = \eta$ and $\xi = \xi_1 \rightarrow \xi_2' \rightarrow \ldots \rightarrow \xi_n' = \eta$ then

$$\bigcup_{1 \leq i < k} act(\xi_i, \xi_{i+1}) \ = \ \bigcup_{1 \leq j < n} act(\xi_j, \xi_{j+1}).$$

Conditions (i) and (iii) assert that the next step relation is compatible with the natural order. By condition (i), whenever η is a possible next step of ξ then η respresents a partial computation of ξ, and by (iii), whenever ξ is a partial computation of η then the intermediate state η can be reached from ξ by performing finitely many steps. Condition (ii) states that whenever ξ_2 can be reached from ξ_1 in one step then each partial computation η which lies between ξ_1 and ξ_2 can be reached from ξ_1 in one step, and there is a step leading from η to ξ_2. Condition (ii) (together with (iv)) reflects the assumption that, whenever the parallel execution of a multiset κ of actions leads from a state ξ_1 to ξ_2 and η is a state between ξ_1 and ξ_2, then κ can be divided into multisets κ_1 and κ_2 such that first performing the actions in κ_1 in parallel, and then the actions in κ_2 leads from ξ_1 to ξ_2 via η. Note that it might be the case that $\xi_1 \rightarrow \xi_2$ is a step such that $act(\xi_1, \xi_2)$ consists of more than one action, and that $\xi_1 \rightarrow \xi_2$ cannot be broken down into a sequence of steps where in each step only a single action is performed. This is due to the fact that a step might stand for the synchronized execution of atomic steps which we represent by the multiset of all actions which participate in the synchronization. Condition (iv) asserts that each state η is associated with a unique multiset of actions which lead from a previous state ξ to η. In other words, we suppose each state to be associated with its 'history': the multiset of actions (more precisely, the partially ordered set of events) which must be performed to reach η from the initial state $\perp$.

DEFINITION 4.2. If $(\rightarrow, act)$ is a next step relation on $\overline{A}$ we say $(\overline{A}, \rightarrow, act)$ is a *linear time model with next step relation*. We say $(\overline{A}, \rightarrow, act)$ is an *interleaving model* iff for each $x \in A$ there exists an enumeration $\xi_0, \xi_1, \xi_2, \ldots$ of the elements of $\mathcal{K}(x)$ such that

$$\xi_0 = \perp \ \rightarrow \ \xi_1 \ \rightarrow \ \xi_2 \ \rightarrow \ \ldots$$

Otherwise we say $(\overline{A}, \rightarrow, act)$ is a *true concurrency model*.

In interleaving models the sets $\mathcal{K}(x)$ are totally ordered w.r.t. the natural order on $\overline{A}$ and the x-paths are exactly the subsequences of the unique sequence (ξ_n) in $\mathcal{K}(x)$ with $\xi_0 = \perp$ and $\xi_n \rightarrow \xi_{n+1}$ for all $n \geq 0$. We say (ξ_n) is the *full x-path*. We refer to the n-th element ξ_n of the (unique) full x-path as the *n-cut* of x and denote it by $x[n]$. In interleaving models, the set of n-cuts of a full x-path is totally ordered, and the next step of a computation x is uniquely determined. Because of this, for the case of interleaving models we choose a linear time logic. In contrast, in true concurrency models, where the partial computations do not specify the order in which concurrent events are executed, there might exist several predecessors for a given intermediate state ξ. For this reason, for the true concurrency approach we use a branching time logic, where the predecessors of an intermediate state arise from parallelism, and not from an explicit non-deterministic choice operator.

Each metric-enriched linear time model $\overline{A}$, together with a next step relation of the form $(\rightarrow, act)$ where

$$\xi \rightarrow \eta \quad \Longleftrightarrow \quad \exists x \in A \; \exists n \in I\!N_0 \quad (\; \xi = x[n] \quad \wedge \; \eta = x[n+1] \;)$$

is an interleaving model. Vice versa, if there is a next step relation $\rightarrow$ on A then a length function on $\overline{A}$ can be defined which turns $\overline{A}$ into a metric-enriched linear time model.

4.2. Linear time logic and interleaving models. We consider a linear time logic LTL which is essentially that of [**25, 9**]. The syntax of LTL is given by:

$$\phi \; ::= \; tt \;\mid\; a \;\mid\; \phi_1 \wedge \phi_2 \;\mid\; \neg\phi \;\mid\; X_\alpha \; \phi \;\mid\; Y_\alpha \; \phi \;\mid\; \phi_1 \; \mathcal{U} \; \phi_2 \;\mid\; \phi_1 \; \mathcal{S} \; \phi_2$$

where $a \in AP$ (AP denotes a set of atomic propositions) and $\alpha \in \Sigma$.

We interpret LTL over arbitrary interleaving models $\overline{A}$ as follows.

DEFINITION 4.3. A LTL *structure* is a 4-tuple $(\overline{A}, \rightarrow, act, L)$ consisting of an interleaving model $(\overline{A}, \rightarrow, act)$ and an interpretation L of the atomic propositions, i.e. L assigns to each atomic proposition a subset $L(a)$ of $\mathcal{K}(A)$.

Let $(\overline{A}, \rightarrow, act, L)$ be a LTL-structure. The elements of $L(a)$ fulfill the condition represented by the atomic proposition a. We identify each computation $x \in A$ with the execution which successively enters the states $x[0], x[1], x[2] \ldots$. In the n-th state $x[n]$, the unique step leading to $x[n+1]$ is performed. A formula ϕ is interpreted over the states of computations which are represented by pairs (x, n) where $x \in A$ is a computation and n a natural number. $(x, n) \models \phi$ means that in the n-th step of the computation x the condition specified by ϕ is fulfilled. An element $x \in A$ satisfies a formula ϕ (denoted by $x \models \phi$) iff ϕ is fulfilled in the initial state, i.e. $(x, 0) \models \phi$. The relation $(x, n) \models \phi$ is defined by structural induction.

$$
\begin{array}{lcl}
(x,n) \models tt & & \\[2ex]
(x,n) \models a & \iff & x[n] \in L(a) \\[2ex]
(x,n) \models \phi_1 \wedge \phi_2 & \iff & (x,n) \models \phi_i,\ i = 1,2 \\[2ex]
(x,n) \models \neg\phi & \iff & (x,n) \not\models \phi \\[2ex]
(x,n) \models X_\alpha\ \phi & \iff & (x,n+1) \models \phi \text{ and } \alpha \in act(x[n], x[n+1]) \\[2ex]
(x,n) \models Y_\alpha\ \phi & \iff & n \geq 1,\ (x,n-1) \models \phi,\ \alpha \in act(x[n-1], x[n]) \\[2ex]
(x,n) \models \phi_1\ \mathcal{U}\ \phi_2 & \iff & \text{there exists } k \geq n \text{ s.t. } (x,k) \models \phi_2 \text{ and} \\
 & & (x,j) \models \phi_1,\ j = n, n+1, \ldots, k-1 \\[2ex]
(x,n) \models \phi_1\ \mathcal{S}\ \phi_2 & \iff & \text{there exists } k \leq n \text{ s.t. } (x,k) \models \phi_2 \text{ and} \\
 & & (x,j) \models \phi_1,\ j = k+1, \ldots, n-1, n
\end{array}
$$

$Sat(\phi)$ denotes the elements $x \in A$ which satisfy ϕ. X_α and $\mathcal{U}$ are called future operators, Y_α and $\mathcal{S}$ past operators. A *past formula* is any formula which does not contain any occurrence of a future operator. A *future formula* is any formula which does not contain any occurrence of a past operator. For Φ to be a past formula and $\xi \in \mathcal{K}(A)$, there exists $x \in A$ and $n \geq 0$ with $x[n] = \xi$ and $(x,n) \models \Phi$ if and only if $(x,n) \models \Phi$ for all $x \in A$ and $n \geq 0$ with $x[n] = \xi$. We put:

$$F_\Phi \ = \ \{\ x[n] \ : \ x \in A,\ n \geq 0,\ (x,n) \models \Phi \ \}$$

We use the following abbreviations. We put:

$$ff \ = \ \neg tt, \quad \phi_1 \vee \phi_2 \ = \ \neg(\neg\phi_1 \wedge \neg\phi_2), \quad \phi_1 \to \phi_2 \ = \ \neg\phi_1 \vee \phi_2$$

and

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
X\phi & = & \bigvee_{\alpha\in\Sigma} X_\alpha\phi, & Y\phi & = & \bigvee_{\alpha\in\Sigma} Y_\alpha\phi, \\[3ex]
\Diamond\phi & = & tt\,\mathcal{U}\,\phi, & \Box\phi & = & \neg\Diamond\neg\phi
\end{array}
$$

As in [**9**], we define safety, guarantee, response and persistence formulas to be formulas of the form $\Box\Phi$, $\Diamond\Phi$, $\Box\Diamond\Phi$ and $\Diamond\Box\Phi$ respectively, where Φ is a past formula. A *liveness* formula is an LTL formula of the form

$$\Diamond \left(\bigvee_{i=1}^{n} \ (\ \Phi_i \ \wedge \ \Diamond\Lambda_i\ ) \right)$$

where Φ_i are past formulas and Λ_i are future formulas such that:

- $\Box(\ \bigvee_{i=1}^{n}\ \Phi_i\ )$ is valid.
- The formulas Λ_i are everywhere eventually satisfiable, i.e. for all $1\ \leq i \leq n$, $y \in A$ and $N \geq 0$ there exists $x \in A$ and $k \geq N$ such that $x[N] = y[N]$ and $(x, k) \models \Lambda_i$.

Instead of the second condition [**9**] require that the future formulas Λ_i are satisfiable. In the case of the linear time model Σ^∞ satisfiability is equivalent to our second condition, which can be seen as follows. Let Λ is a satisfiable future formula (satisfiability w.r.t. Σ^∞) and $s \in \Sigma^\infty$, $l \geq 0$ such that $(s,l)\ \models\ \Lambda$. Then, for each $t \in \Lambda^\infty$ and $N \geq 0$, let u be the string $s[N]t$. Then $u[N] = t[N]$ and $(u,k)\ \models\ \Lambda$ where $k = N + l$. In order to see that $(u,k)\ \models\ \Lambda$ it is essential that Λ does not contain past operators.

LEMMA 4.4. *Let Φ be a past formula.*

(a) *$Sat(\ \Box\Phi\ )\ =\ \mathcal{A}(F_\Phi)$ (a safety property)*

(b) *$Sat(\ \Diamond\Phi\ )\ =\ \mathcal{E}(F_\Phi)$ (a guarantee property)*

(c) *$Sat(\ \Box\Diamond\Phi\ ) \setminus \mathcal{K}(M)\ =\ \mathcal{R}(F_\Phi)$ (a response property)*

(d) *$Sat(\ \Diamond\Box\Phi\ ) \setminus \mathcal{K}(M)\ =\ \mathcal{P}(F_\Phi)$ (a persistence property)*

(e) *If Λ is a liveness formula then $Sat(\Lambda)$ is a liveness property.*

PROOF. (a) $x \in Sat(\ \Box\Phi\ )$ iff $(x, \perp) \models \Box\Phi$ iff $(x,n) \models \Phi$ for all $n \geq 0$ iff $x[n] \in F_\Phi$ for all $n \geq 0$ iff $x \in \mathcal{A}(F_\Phi)$.

(b) $x \in Sat(\ \Diamond\Phi\ )$ iff $(x, \perp) \models \Diamond\Phi$ iff $(x,n) \models \Phi$ for some $n \geq 0$ iff $x[n] \in F_\Phi$ for some $n \geq 0$ iff $x \in \mathcal{E}(F_\Phi)$.

(c) $x \in Sat(\Box\Diamond\Phi)$ iff $(x,n) \models \Phi$ for infinitely many n iff $x[n] \in F_\Phi$ for infinitely many n iff $x \in \mathcal{R}(F_\Phi)$.

(d) $x \in Sat(\Diamond\Box\Phi)$ iff $(x,n) \models \Phi$ for almost all n iff $x[n] \in F_\Phi$ for almost all n iff $x \in \mathcal{P}(F_\Phi)$.

(e) Let $\xi \in \mathcal{K}(A)$. Then $\xi = y[N]$ for some $y \in A$ and $N \geq 0$. We have to show that there exists $x \in Sat(\Lambda)$ with $x[N] = \xi$. Let

$$\Lambda\ =\ \Diamond \left(\bigvee_{i=1}^{n}\ (\ \Phi_i\ \wedge\ \Diamond\Lambda_i\ ) \right)$$

Since $\Box(\bigvee \Phi_i)$ is valid, $(y, N)\ \models\ \Phi_i$ for some i. Because Λ_i is everywhere eventually satisfiable, there exists $x \in A$ and $k \geq N$ such that $x[N] = \xi$ and $(x,k) \models \Lambda_i$. Since Φ_i is a past formula we get $(x,N)\ \models\ \Phi_i$. Since $k \geq N$ we have $(x,N)\ \models\ \Diamond\Lambda_i$, and hence

$$(x,N)\ \models\ \Phi_i\ \wedge\ \Diamond\Lambda_i$$

and therefore $x\ \models\ \Lambda$.

$\Box$

4.3. Partial order logic and true concurrency models. In this section we briefly introduce the logic $ISTL^*$ [**15, 29**] and show how its formulas can be interpreted over order-enriched linear time models. The reader is cautioned to note that our interpretation of $ISTL^*$ is over more general, non-standard models, but coincides with that of [**29**] for a suitably chosen next step relation. In [**15**] and [**29**]

$ISTL^*$ formulas are interpreted over *interleaving sequences* of partial order executions (i.e. linearizations of pomsets of a certain kind), and Mazurkiewicz traces respectively, whereas we give semantics (for syntactically the same formulas) in arbitrary order-enriched linear time models.

A *state formula* of $ISTL^*$ is a formula ϕ given by the grammar:

$$\phi \ ::= \ tt \ | \ a \ | \ \phi_1 \wedge \phi_2 \ | \ \neg\phi \ | \ A\psi$$

where $a \in AP$ is an atomic formula and ψ is a *path formula* built from the following production system:

$$\psi \ := \ \phi \ | \ \psi_1 \wedge \psi_2 \ | \ \neg\psi \ | \ X_\alpha\ \psi \ | \ Y_\alpha\ \psi \ | \ [\psi_1\ \mathcal{U}\ \psi_2] \ | \ [\psi_2\ \mathcal{S}\ \psi_2]$$

where ϕ is a state formula and $\alpha \in \Sigma$.

We use the following abbreviations:

$$ff \ = \ \neg tt, \quad f_1 \vee f_2 \ = \ \neg(\neg f_1 \wedge \neg f_2), \quad f_1 \rightarrow f_2 \ = \ \neg f_1 \vee f_2$$

for all state or path formulas f_1, f_2. If ψ, ψ' are path formulas then

$$E\psi \ = \ \neg A(\neg\psi), \qquad F\psi \ = \ [tt\ \mathcal{U}\ \psi], \qquad G\psi \ = \ \neg[tt\ \mathcal{U}\ (\neg\psi)],$$

$$P\psi \ = \ [tt\ \mathcal{S}\ \psi], \qquad H\psi \ = \ \neg[tt\ \mathcal{S}\ (\neg\psi)],$$

$$X\psi \ = \ \bigvee_{p\in\Gamma}\ X_p\psi, \qquad Y\psi \ = \ \bigvee_{p\in\Gamma}\ Y_p\psi.$$

DEFINITION 4.5. A $ISTL^*$ *structure* is a 4-tuple $(\overline{A}, \rightarrow, act, L)$ where $(\overline{A}, \rightarrow, act)$ is a linear time model with next step relation and L an interpretation of the atomic propositions, i.e. a function which assigns to each atomic proposition a subset $L(a)$ of $\mathcal{K}(A)$ consisting of those states ξ which are supposed to satisfy the condition a.

Let $(\overline{A}, \rightarrow, act, L)$ be a $ISTL^*$ structure. State formulas are interpreted over intermediate states of computations which we represent by pairs (x, ξ) where $x \in A$ and $\xi \in \mathcal{K}(x)$. Path formulas are interpreted over states of observations.

DEFINITION 4.6. Let $(\overline{A}, \rightarrow, act)$ be a linear time model with next step relation. An *observation* on $\overline{A}$ is a sequence $\pi = (\xi_n)_{n\geq 0}$ in $\mathcal{K}(A)$ such that:

- either $\xi_i \rightarrow \xi_{i+1}$ for all $i \geq 0$, or
- there is some $k \geq 0$ such that

$$\xi_0 \ \rightarrow \ \xi_1 \ \rightarrow \ \xi_2 \ \rightarrow \ \dots \ \rightarrow \ \xi_k \ = \ \xi_{k+1} \ = \ \xi_{k+2} \ = \ \dots$$

We write $\pi(i)$ to denote the i-th element of π, i.e. if $\pi = (\xi_0, \xi_1, \dots)$ then $\pi(i) = \xi_i$. π is called a *x-observation* iff in addition $\bigsqcup \xi_i = x$. An *initial x-observation* is an x-observation $\pi = (\xi_0, \xi_1, \dots)$ with $\xi_0 = \perp$.

The path quantifiers A and E of $ISTL^*$ range over x-observations. The set of all such observations is an 'Abramhamson structure', i.e. suffix-closed and fusion-closed (cf. [**1, 11, 15**]). Suffix-closedness means that if $(\xi_n)_{n\geq 0}$ is an x-observation then also $(\xi_n)_{n\geq k}$ is an x-observation for arbitrary $k \geq 0$. Fusion-closedness means that if $(\xi_n)_{n\geq 0}$ and $(\eta_n)_{n\geq 0}$ are x-observations such that $\xi_n = \eta_k$ for some $n \geq 0$ and $k \geq 0$ then the sequence

$$\xi_0,\ \xi_1,\ \ldots,\ \xi_n = \eta_k,\ \eta_{k+1},\ \eta_{k+2},\ \ldots$$

is an x-observation.

A computation x is said to satisfy a state formula ϕ (denoted by $x \models \phi$) iff x satisfies ϕ in its initial state $\perp$, i.e. iff $(x, \perp) \models \phi$. Here $(x,\xi) \models \phi$ where $x \in A$, $\xi \in \mathcal{K}(x)$, is defined by structural induction:

$$
\begin{array}{lll}
(x,\xi) \models tt & & \\[2ex]
(x,\xi) \models a & \iff & \xi \in L(a) \\[2ex]
(x,\xi) \models \phi_1 \wedge \phi_2 & \iff & (x,\xi) \models \phi_i,\ i = 1,2 \\[2ex]
(x,\xi) \models \neg\phi & \iff & (x,\xi) \not\models \phi \\[2ex]
(x,\xi) \models A\psi & \iff & (\pi, i) \models \psi \text{ for each } x\text{-observation } \pi \text{ with } \pi(i) = \xi
\end{array}
$$

and for each observation $\pi = (\xi_0, \xi_1, \xi_2, \ldots)$ and $i \geq 0$:

$$
\begin{array}{lll}
(\pi,i) \models \phi & \iff & (x,\xi_i) \models \phi \ \text{ where } x = \bigsqcup \xi_n \\[2ex]
(\pi,i) \models \psi_1 \wedge \psi_2 & \iff & (\pi,i) \models \psi_i,\ i = 1,2 \\[2ex]
(\pi,i) \models \neg\psi & \iff & (\pi,i) \not\models \psi \\[2ex]
(\pi,i) \models X_\alpha\, \psi & \iff & (\pi,i+1) \models \psi,\ \ \alpha \in act(\xi_i, \xi_{i+1}) \\[2ex]
(\pi,i) \models Y_\alpha\, \psi & \iff & i \geq 1 \text{ and } (\pi,i-1) \models \psi,\ \ \alpha \in act(\xi_{i-1}, \xi_i) \\[2ex]
(\pi,i) \models [\psi_1\ \mathcal{U}\ \psi_2] & \iff & \text{there exists } k \geq i \text{ s.t. } (\pi,k) \models \psi_2 \\
 & & \text{and } (\pi,j) \models \psi_1,\ j = i, i+1, \ldots, k-1 \\[2ex]
(\pi,i) \models [\psi_1\ \mathcal{S}\ \psi_2] & \iff & \text{there exists } k \leq i \text{ with } (\pi,k) \models \psi_2 \\
 & & \text{and } (\pi,j) \models \psi_1,\ j = k+1, \ldots, i-1, i
\end{array}
$$

REMARK 4.7. If $ISTL^*$ formulas are interpreted over a LTL structure $(\overline{A}, \rightarrow, act, L)$ then the quantifiers E and A have the same interpretation. This is because $x[0], x[1], \ldots$ is the unique x-observation. In this case the logic $ISTL^*$ reduces to the linear time logic. Let $\widehat{\phi}$ be the LTL formula which arises from a state formula ϕ by removing the quantifiers A and E. Then $x \models_{IST} \phi$ if and only if $x \models_{LT} \widehat{\phi}$. (The index LT, resp. IST, denotes whether $(\overline{A}, \rightarrow, act, L)$ is assumed to be a LTL structure or a $ISTL^*$ structure.)

Let $Sat(\phi)$ be the set of all $x \in D$ which satisfies ϕ:

$$Sat(\phi) \ = \ \{ \ x \in D \ : \ x \models \phi \ \}$$

The operators $\mathcal{U}$ and X are called future operators, $\mathcal{S}$ and Y past opertors. A *past formula* is a formula which does not contain any future operators. A future formula is a formula without past operators. Let Φ be a past state formula and $\xi \in \mathcal{K}(A)$. Then $(x, \xi) \models \Phi$ for some $x \in A$ with $\xi \sqsubseteq x$ if and only if $(x, \xi) \models \Phi$ for all $x \in A$ with $\xi \sqsubseteq x$. We define:

$$F_\Phi \ = \ \{ \xi \in \mathcal{K}(A) \ : \ (x, \xi) \models \Phi \text{ for some } x \in A \ \}$$

Safety, guarantee, response and persistence properties are given by the forms $AG\Phi$, $EF\Phi$, $EGF\Phi$ and $AFG\Phi$ respectively, where Φ is a past state formula. A *liveness* formula is a state formula of the form

$$EF \left(\bigvee_{i=1}^{n} (\Phi_i \ \wedge \ F\Lambda_i) \right)$$

where Φ_i are past state formulas and Λ_i future state formulas such that

- $AG(\bigvee \ \Phi_i)$ is valid
- Λ_i is everywhere eventually satisfiable, i.e. for each $\xi \in \mathcal{K}(A)$ there exists $x \in A$ and $\eta \in \mathcal{K}(x)$ with:

$$\xi \ \sqsubseteq \ \eta \ \sqsubset \ x, \quad (x, \eta) \ \models \ \Lambda_i$$

LEMMA 4.8. *Let Φ be a past formula. Then:*

(a) $Sat(\ AG\Phi\ ) \ = \ \mathcal{A}(\ F_\Phi\ )$
(b) $Sat(\ EF\Phi\ ) \ = \ \mathcal{E}(\ F_\Phi\ )$
(c) $Sat(\ EGF\Phi\ ) \setminus \mathcal{K}(D) \ = \ \mathcal{R}(\ F_\Phi\ )$
(d) $Sat(\ AFG\Phi\ ) \setminus \mathcal{K}(D) \ = \ \mathcal{P}(\ F_\Phi\ )$
(e) *If Λ is a liveness formula then $Sat(\Lambda)$ is a liveness property.*

PROOF. (a)-(d) The proof is similar to Lemma 4.4. $\mathcal{K}(D)$ should be clear from the context.

(e) Let $\Lambda \ = \ EF\ (\ \bigvee(\Phi_i \wedge F\Lambda_i)\ )$ be a liveness formula. Let $\xi \in \mathcal{K}(A)$. Since Φ_i are past formulas and since $AG(\bigvee \Phi_i)$ is valid, there is some i with $\xi \in F_{\Phi_i}$. Since Λ_i is everywhere eventually satisfiable there exists $x \in A$ and $\eta \in \mathcal{K}(x)$ such that $\xi \sqsubseteq \eta \sqsubset x$ and $(x, \eta) \models \Lambda_i$. Let

π be an initial x-observation such that $\pi(j) = \xi$ and $\pi(k) = \eta$ for some $0 \leq j \leq k$. Then $(\pi, k) \models \Lambda_i$ and $(\pi, j) \models \Phi_i$. Hence, $(\pi, j) \models \Phi_i \wedge F\Lambda_i$ and therefore

$$(\pi, 0) \models F(\bigvee(\Phi_i \wedge F\Lambda_i)).$$

Thus, $x \models \Lambda$, i.e. $x \in Sat(\Lambda)$.

□

We have not been able to find syntactic descriptions of obligation and reactivity properties, and also progress properties in the sense of [**9**]. It is an open problem whether it can be shown that each of these classes of formulas is characteristic in the sense that each extensional property corresponds to a syntactic property.

The following lemma shows that our requirement that an x-observation approximates x ensures that each action α which is enabled in some state ξ during some execution of x is actually performed in every linearization of x at a state subsuming (above) ξ, cf maximality [**16**]. If $\alpha \in \Sigma$ then we put:

$$en_\alpha \ = \ EX_\alpha tt, \quad \mathrm{ex}_\alpha \ = \ AFX_\alpha tt$$

Then $(x, \xi) \models en(\alpha)$ iff the action α is enabled (i.e. can be performed) in the state ξ. $(x, \xi) \models \mathrm{ex}_\alpha$ iff in each execution of x the action α will be performed at some state subsuming ξ.

LEMMA 4.9. *The formula $AG(\ en_\alpha \ \rightarrow \ \mathrm{ex}_\alpha)$ holds for all $x \in A$.*

4.4. Examples.

4.4.1. *Interpreting LTL over strings.* The temporal logic used in [**30, 25, 9**] is essentially the same as our logic LTL, the only difference being that our next/previous step operators X_α, Y_α are labelled with actions α. Our interpretation of LTL formulas (using X, Y instead of X_α, Y_α) over the metric-enriched linear time model Σ^∞ coincides with that of [**9**].

The language LTL also includes Lamport's linear time logic (called TL) [**23, 28**]. TL formulas are built from the atomic propositions using the ordinary logical operators $\vee$, $\wedge$ and $\neg$, and the temporal operators $\Box$ and $\Diamond$. The interpretation of [**23**] of TL-formulas over sequences of system states corresponds to our interpretation of TL for the case of the interleaving model $\Theta^\infty \ = \ \Theta^* \cup \Theta^\omega$. Here Θ denotes a set of (possible) system states, typically mappings from program and control variables to values. Θ^* denotes the set of finite sequences over Θ and Θ^ω the set of infinite sequences. Terminating computations are represented by infinite strings where the final state is repeated infinitely often.

[**28**] defines safety properties as those which are induced by formulas of the form $a \rightarrow \Box b$ where a and b are atomic propositions. Liveness formulas in the sense of [**28**] have the form $\Box(a \rightarrow \Diamond b)$.

4.4.2. *Interpreting ISTL over traces.* The logic $ISTL^*$ of [**29**] is interpreted over Mazurkiewicz traces. We now consider its relationship with our linear model framework.

In [**29**], the starting point is a program described by a tuple $(\Sigma, \iota, \Theta, \overline{y})$ where (Σ, ι) is a concurrent alphabet, Θ a satisfiable predicate (the initial condition) and $\overline{y}$ a finite sequence of program variables.

An assignment for $\overline{y}$ is a function J which assigns to each program variable y a value $J(y)$ of the domain of y. The assignments can be viewed as states of the program. Each $\alpha \in \Sigma$ is associated with a pair $< en_\alpha, f_\alpha >$ where en_α is an enabling condition and f_α a transformation that describes the effect of α applied in a state where en_α holds, i.e. f_α is a function which assigns to each assignment J for $\overline{y}$ with $J \models en_\alpha$ an assignment $f_\alpha(J)$. (Here a satisfaction relation $\models$ for the enabling conditions and the assignments for $\overline{y}$ is supposed such that $J \models en_\alpha$ iff α is enabled in J.) Moreover, the commutativity of independent actions and the fact that independent actions can neither disable nor enable each other is required. Formally, for all actions α, β with $\alpha\iota\beta$ and all assignments J for $\overline{y}$:

- If $J \models en_\alpha \wedge en_\beta$ then $f_\alpha(f_\beta(J)) = f_\beta(f_\alpha(J))$.
- If $J \models en_\alpha$ then $J \models en_\beta$ if and only if $f_\alpha(J) \models en_\beta$.

For simplicity, we assume a fixed initial state (an assignment J_{init} for $\overline{y}$). (J_{init} might be either an assignment where the initial condition Θ holds or an 'accessible' assignment, i.e. an assignment J which is reachable from an assignment where the initial condition holds.) We define Σ^*_{init} to be the set of finite strings $s = \alpha_0\alpha_1 \ldots \alpha_n$ over Σ such that $J_i \models en_{\alpha_i}$, $i = 0, 1, \ldots, n$ where $J_0 = J_{init}$ and $J_{i+1} = f_{\alpha_i}(J_i)$. The interpretation J_{n+1} is called the 'final interpretation' of s and is denoted by fin_s. The commutativity of independent actions implies that if $s \equiv t$ then $fin_s = fin_t$. Hence, we may define $fin_\xi = fin_s$ for each finite trace $\xi = [s]$ where $s \in \Sigma^*_{init}$. Let x be an infinite trace such that $x = [s]$ for some infinite string s over Σ where all prefices of s belong to Σ^*_{init}. x can be viewed as a 'run' in the sense of [**29**] (which is defined as a maximal subset of $[\Sigma^*_{init}]$ consisting of pairwise consistent traces where the consistency of two finite traces ξ_1, ξ_2 means that ξ_1, $\xi_2 \sqsubseteq \xi$ for some finite trace ξ). An 'observation' of x in the sence of [**29**] is a sequence of traces $\xi_0, \xi_1, \ldots$ such that ξ_0 is the empty trace, $\xi_{i+1} = \xi_i[\alpha_i]$ for some $\alpha_i \in \Sigma$, and whenever $\xi \sqsubseteq x$ then $\xi \sqsubseteq \xi_i$ for some i. Hence, the observations of x in the sense of [**29**] are exactly the x-observations in the linear time model $[\Sigma^\infty]$, together with the next step relation $\rightarrow$ defined by:

$$x \rightarrow y \iff \exists s \in \Sigma^*, \alpha \in \Sigma x = [s] \wedge y = [s\alpha]$$

where $act([s], [s\alpha]) = \{\alpha\}$ is the multiset containing α.

We assume that there is a satisfaction relation $\models$ for the atomic propositions and the interpretations J for the program variables $\overline{y}$ such that $J \models a$ iff a is true in the state J. This yields an interpretation L for the atomic propositions which assigns to each atomic proposition a a set $L(a)$ of finite traces $\xi \in [\Sigma^*_{init}]$:

$$\xi \in L(a) \text{ iff } fin_\xi \models a.$$

[**29**] associates each run x with an $ISTL^*$-structure and obtains a satisfaction relation $\models_x$ for each run x. This satisfaction relation agrees with ours (in the sense that $\perp \models_x \varphi$ iff $x \models \varphi$) when we deal with the linear time model of traces, the next step relation $\rightarrow$ and the interpretation L as above. Here we replace the next step operator X in [**29**] by the labelled next step operators X_α, and similarly their first order (state) formulas are substituted by atomic propositions.

Instead of x-observations, which are maximal in the order-theoretic sense, [**21**] use arbitrary observations (ξ_n) in $\mathcal{K}(x)$ as executions of x; for example, the latter admits non-maximal Mazurkiewicz traces. In [**21**] a computation x in a state ξ satisfies a formula of the form $E\phi$ iff there exists an observation π in $\mathcal{K}(x)$ starting in ξ with $(\pi, 0) \models \phi$. Notice that it is not required that π approximates x, i.e. the case $\bigsqcup \pi(i) \sqsubset x$ is allowed. For instance, the formula

$$\Phi = EGX_\alpha tt$$

is satisfied in the approach of [**21**] by the trace $[s]$, $s = \beta\alpha\alpha\alpha\ldots$, where $\alpha \iota \beta$, but not in our framework. This is because

$$[\alpha] \to [\alpha\alpha] \to [\alpha\alpha\alpha] \to \ldots$$

is considered an execution of $[s]$ in [**21**], but not in this paper.

Another useful next step relation on $[\Sigma^\infty]$ is given by:

$$x \Rightarrow y$$

iff there exists pairwise independent actions $\alpha_1, \ldots, \alpha_n \in \Sigma$ such that $x[\alpha_1 \ldots \alpha_n] = y$. The associated multiset of actions is

$$act(\ x,\ x[\alpha_1, \ldots, \alpha_n]\ ) = \text{multiset consisting of } \alpha_1, \ldots, \alpha_n.$$

This next step relation allows the parallel execution of pairwise independent actions in one step. The interpretation of $ISTL^*$ formulas over the $ISTL^*$ structure

$$([\Sigma^\infty], \Rightarrow, act, L)$$

differs from the interpretation over $([\Sigma^\infty], \to, act, L)$ in the next (resp. previous) step operators X_α (resp. Y_α).

LEMMA 4.10. *If ϕ is a formula which does not contain the operators X_α and Y_α then an infinite trace x satisfies ϕ w.r.t. the next step relation $\to$ if and only if ϕ is satisfied by x using the interpretation based on the next step relation $\Rightarrow$.*

If $ISTL^*$ is used to formulate real-time constraints such as 'a process responds to a request within 3 time units', and if we suppose that each atomic action can be executed in a single time unit, the next step relation $\to$ is not helpful since it ignores the fact that the parallel execution of pairwise independent actions $\alpha_1, \ldots, \alpha_n$ can be performed within a single time unit. Consider the formula

$$\phi = EG(Y_\beta tt \to XX_\alpha tt)$$

where β stands for an (input-)action which is performed by a handshake mechanism and where α is an (output-)action representing the acknowledge for the receipt of the message transmitted by β. Then ϕ ensures the existence of an execution which satisfies the following: whenever the system receives a message it acknowledges the receipt after two time units. Let $s = \beta\gamma\alpha t$ where $\beta \iota \gamma$, $\neg(\ \alpha \iota \beta\ )$ and

$\neg(\ \alpha\ \iota\ \gamma\ )$ and where $t\ =\ \surd\surd\surd\ldots$. One might think of γ as an input-action where the message is transmitted on a channel different from that which is used for β (hence β and γ can be performed in parallel) and the acknowledge sent by α consists of a message that uses an information which is given by γ (hence α and γ are dependent). Using the next step relation $\Rightarrow$ we get that the trace $[s]$ satisfies ϕ. By means of $\rightarrow$ the trace $[s]$ does not satisfy ϕ.

4.4.3. *Interpreting $ISTL$ over pomsets.* When considering the order-enriched linear time model Pom^{∞} there are two natural ways to define the next step relation.

The first possibility is to define the step relation $x \rightarrow y$ iff $x \sqsubset y$ and whenever $x \sqsubseteq z \sqsubseteq y$ then either $x = z$ or $z = y$. Then $x \rightarrow y$ iff $x\ =\ y\lceil S$ where S arises from the event set of y by removing a single event e of maximal depth. If α is the label of this event e in y then we put $act(x, y) = \{\alpha\}$.

[**15**] proposes an interpretation of $ISTL^*$ over pomsets of a certain kind, called 'partial order executions'. In the approach of [**15**] the actions α are associated with an operation which explains how the variables of a system are modified when α is executed. A partial order execution is then a pomset together with an initial 'snapshot' (i.e. a partial function from variables to values) such that each pair of events e, e' which affect the same variables are ordered, i.e. either $e \leq e'$ or $e' \leq e$. In the approach of [**15**] intermediate states of a computation represented by a partial order execution x are 'slices', i.e. a left-closed finite set S' of the event set of x. Hence, a slice of a pomset x can be identified with a finite pomset $\xi \in \mathcal{K}(x)$ which is an intermediate state in our approach. [**15**] interpret path formulas over 'acceptable paths': if x is a partial order execution then an acceptable path of x is a sequence (S_n) of x-slices such that $S_n = S_{n+1} \setminus \{e\}$ for some maximal event e in S_{n+1} and such that each event e of x is contained in some slice S_n. Hence, an acceptable path is an x-observation w.r.t. the next step relation $\rightarrow$. Identifying partial order executions and pomsets we obtain that the interpretation of $ISTL$ in the sense of [**15**] agrees with our interpretation using the linear time model of pomsets and the next step relation $\rightarrow$.

Secondly, we consider the next step relation $\Rightarrow$ defined as follows. Let $y = (S, \leq, l)$ and $x = y\lceil S'$ where $S' \subset S$ is left-closed. Then $x \Rightarrow y$ iff, for all $e, e'\ \in\ S \in S'$, either $e = e'$ or $\neg(e \leq e') \wedge (e' \leq e)$. I.e. $x \Rightarrow y$ iff the events in $S \setminus S'$ are pairwise independent. In this case the step from x to y stands for the parallel execution of the events $S \setminus S'$. We define $act(x, y)$ to be multiset of all actions $l(e)$, $e \in S \setminus S'$.

5. Conclusion and Further Work

We have formulated an abstract, axiomatically given notion of a *linear time* model, and considered classes of behavioural properties (safety, guarantee, liveness, response and persistence) in such models. Our framework admits the interleaving models, as well as some 'true concurrency' models such as Mazurkiewicz traces and pomsets as special cases, but it does not handle full non-determinism. In this general framework we have been able to obtain extensional, topological and temporal characterizations of classes of properties including safety and liveness, generalising many of the results of [**3, 9, 21**]. As yet, we do not know how to admit the automata-theoretic characterization of [**9**] into our framework, and how to syntactically characterize properties such as reactivity. This is the subject of future study.

References

[1] K. Abrahamson, *Decidability and expressiveness of logics of programs*, Ph.D. Thesis, University of Washington at Seattle, 1980.

[2] S. Abramsky, A. Jung, *Domain Theory*, In S. Abramsky, D.M. Gabbay and T.S.E. Maibaum, editors, Handbook of Logic in Computer Science, vol. 3, Clarendon Press, 1994.

[3] B. Alpern, F. Schneider, *Defining liveness*, Information Processing Letters **21**, 1985, 181–185.

[4] B. Alpern, F. Schneider, *Recognizing safety and liveness*, Distributed Computing **2**, 1987, 117–126.

[5] C. Baier, M. Majster-Cederbaum: Denotational Linear Time Semantics and Sequential Composition, Information Processing Letters, Vol. 59, pp 135-143, 1996.

[6] C. Baier, M. Majster-Cederbaum: Metric Semantics from Partial Order Semantics, Acta Informatica, Vol. 34, pp 701-735, 1997.

[7] J.W. de Bakker, J.H.A. Warmerdam, *Metric pomset semantics for a concurrent language with recursion*, In I. Guessarian, editor, Semantics of Systems of Concurrent Processes, LNCS vol. 469, Springer-Verlag, 1990, 21–49.

[8] E.M. Clarke, E.A. Emerson, *Design and Synthesis of Synchronization Skeletons using Branching Time Temporal Logic*, In Proc. Workshop on Logics of Programs, LNCS vol. 131, Springer-Verlag, 1981.

[9] E. Chang, Z. Manna, A. Pnueli, *The Safety-Progress Classification*, In Proc. Computer and System Science, NATO Advanced Science Institute Series, Springer-Verlag, 1992.

[10] E. Chang, Z. Manna, A. Pnueli, *Characterization of Temporal Property Classes*, In W. Kuich, editor, Proc. ICALP92, LNCS vol. 623, Springer-Verlag, 1992, 474–486.

[11] C. Courcoubetis, M. Vardi, P. Wolper, *Reasoning about fair concurrent programs*, In Proc. 18th ACM Symposium on Theory of Computing, Berkeley, ACM Press, 1986.

[12] R. Engelking, *General Topology*, Sigma Series in Pure Mathematics, vol. 6, Heldermann Verlag, Berlin, 1989.

[13] G. Gierz, H. Hofmann, K. Keimel, J. Lawson, M. Mislove, D. Scott, *A Compendium of Continuous Lattices*, Springer-Verlag, 1980.

[14] U. Goltz, R. Loogen, *Modelling Nondeterministic Concurrent Processes with Event Structures*, Fundamenta Informaticae **14** 1 (1991) 39–74.

[15] S. Katz, D. Peled, *Interleaving Set Temporal Logic*, Theoretical Computer Science **75** 3 (1990) 21–43.

[16] M. Kwiatkowska, *Event Fairness and Non-Interleaving Concurrency*, Formal Aspects of Computing **1** 3 (1989) 213–228.

[17] M. Kwiatkowska, *Defining Process Fairness for Non-Interleaving Concurrency.* In Proc. Foundations of Software Technology and Theoretical Computer Science, LNCS vol. 472, Springer-Verlag, 1990, 286–300.

[18] M. Kwiatkowska, *A metric for traces*, Information Processing Letters **35** (1990) 129–135.

[19] M. Kwiatkowska. *On the domain of traces and sequential composition.* In S. Abramsky and T.S.E.. Maibaum, editors, Proc. 16th Coll. on Trees in Algebra and Programming (CAAP'91), LNCS vol. 493, Springer-Verlag, 1991, 42–56.

[20] M. Kwiatkowska. *On topological characterization of behavioural properties.* In G. Reed, A. Roscoe, and R. Wachter, editors, Topology and Category Theory in Computer Science, Oxford University Press, 1991, 153–177.

[21] M. Kwiatkowska, D. Peled, W. Penczek, *A Hierarchy of Partial Order Temporal Properties.* In Proc. Temporal Logic, LNCS vol. 827, Springer-Verlag, 1994, 398–414.

[22] L. Lamport, *Proving the Correctness of Multiprocess Programs*, IEEE Trans. Software Engineering **SE-3** 2, (1977) 125–143.

[23] L. Lamport, *Specifying Concurrent Program Modules*, ACM Transactions on Programming Languages and Systems **5** 2 (1983).

[24] M. Majster-Cederbaum, C. Baier, *Metric Completion versus Ideal Completion.* Theoretical Computer Science, Vol. 170, pp 145-171, 1996.

[25] Z. Manna, A. Pnueli, *A Hierarchy of Temporal Properties.* In Proc. 9th ACM Symposium on Principles of Distributed Computing, ACM Press, 1990, 377–408.

[26] A. Mazurkiewicz, *Basic notions of trace theory.* In Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, LNCS vol. 354, Springer-Verlag, 1988, 25–34.

[27] R. Milner, *Communication and Concurrency*, Prentice Hall, 1989.

[28] S. Owicki, L. Lamport, *Proving Liveness Properties of Concurrent Programs*, ACM Transactions on Programming Languages and Systems, **4** 3 (1982) 455–495.

[29] D. Peled, A. Pnueli, *Proving Partial Order Liveness Properties.* In Proc. ICALP'90, Warwick, LNCS vol. 443, Springer-Verlag, 553–571.

[30] A. Pnueli, *The Temporal Logic of Programs*, Proc. 18th Ann. Symp. on Foundations of Computer Science, Providence, IEEE Press, 1977.

[31] V. Pratt, *The Pomset Model of Parallel Processes: Unifying the Temporal and the Spatial.* In Proc. Seminar on Concurrency, LNCS vol. 197, Springer-Verlag, 1984.

[32] W. Reisig, *Partial Order Semantics versus Interleaving Semantics for CSP-like Languages and its Impact on Fairness.* In Proc. ICALP'84, LNCS vol. 172, Springer-Verlag, 1984, 403–413.

[33] W. Reisig, *Elements of a Temporal Logic Coping with Concurrency*, SFB-Bericht 342/23, 92A, Techn. Universität München, 1992.

[34] W. Thomas, *Automata on Infinite Objects.* In J. van Leeuwen, editor, Handbook of Theoretical Computer Science vol. B, North-Holland, 1990, 135–191.

[35] G. Winskel, *An introduction to event structures.* In Linear Time, Branching Time and Partial Order in Models and Logics for Concurrency, LNCS vol. 354, Springer-Verlag, 1988, 364–397.

Fakultät für Mathematik & Informatik, Universität Mannheim, 68131 Mannheim, Germany
E-mail address: baier@informatik.uni-mannheim.de

School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK
E-mail address: M.Z.Kwiatkowska@cs.bham.ac.uk