Substructure Temporal Logic

Massimo Benerecetti, Fabio Mogavero, and Aniello Murano
Università degli Studi di Napoli Federico II

***Abstract*—In formal verification and design, reasoning about substructures is a crucial aspect for several fundamental problems, whose solution often requires to select a portion of the model of interest on which to verify a specific property.**

In this paper, we present a new branching-time temporal logic, called Substructure Temporal Logic (STL*, for short), whose distinctive feature is to allow for quantifying over the possible substructure of a given structure. This logic is obtained by adding two new operators to CTL*, whose interpretation is given relative to the partial order induced by a suitable substructure relation. STL* turns out to be very expressive and allows to capture in a very natural way many well known problems, such as module checking, reactive synthesis and reasoning about games. A formal account of the model theoretic properties of the new logic and results about (un)decidability and complexity of related decision problems are also provided.

I. Introduction

Since the seminal paper by Pnueli [20], *temporal logic*, a special kind of *modal logic* geared towards the description of the temporal ordering of events, has been established as the de facto specification language for system verification and design. Depending on the possible views of the underlying nature of time, two varieties of temporal logics are mainly considered in the literature. In *linear-time temporal logics*, such as LTL [20], time is considered as an infinite chain of different time instants, each one having a unique immediate future moment. Under this view, formulas are interpreted over linear sequences describing the ongoing behavior of system computations. Conversely, in *branching-time temporal logics*, such as CTL [4], CTL$^+$ [7], and CTL* [8], each time instant may split into several possible immediate future moments and a suitable pair of operators, the *existential* and *universal path quantifiers*, are used to express properties along some or all possible temporal branches. Accordingly, formulas of these logics are interpreted over branching structures, such as infinite trees, which better characterize nondeterministic behaviors of incompletely specified deterministic systems.

The success of such a specification framework is due to a multiplicity of factors, most notably, the ability to express relevant properties of computational systems and the discovery of algorithmic methods to solve the principal decision problems related to system verification and design. From the standpoint of verification, *model checking* [4], [5], [6] is a well-established formal method that allows to automatically check for global system correctness. In order to check whether a system satisfies a required property, we describe its structure through mathematical models like *Kripke structures* or *labeled transition systems*. A more challenging problem, from the standpoint of design, is *synthesis* [3], which is based on the appealing idea of building a system directly from its specification, instead of first developing it and then verifying its correctness. The modern approach to this problem was initiated by Pnueli and Rosner in [21], who introduced LTL *reactive synthesis*.

Over the years, an enormous body of work has been devoted to increase the expressive power of temporal logics, so as to capture more and more complex system behaviors. To this aim, two main directions have been followed. The first one is to extend the semantics of already defined logics, by changing the interpretation of their syntactic operators. The second one, instead, is to extend the syntax, by replacing or introducing new operators. The success of the resulting extensions often depends upon the ratio between the achieved gain in expressiveness and the consequent increase in the complexity of the related decision problems.

One of the most important semantic extensions, which has proved to be fundamental in practice for the verification of liveness properties, was the introduction of *fairness constraints* into CTL [9]. The resulting semantics restricts the interpretation of the path quantifiers to range over fair paths only, in order to rule out unrealistic executions. Another classic semantic extension was the introduction of *module checking* for branching time formulas [15], which corresponds to model checking in the context of *open system* analysis. An open system is modeled as a module interacting with the environment and its correctness requires that the desired property holds with respect to all such interactions. In this case, the entire definition of the modeling relation changes. Similarly, the reactive synthesis problem can be formulated as a semantic extension of the concept of synthesis of a model for a logic formula. While classic synthesis corresponds to the construction of a witness for the satisfiability, reactive synthesis further requires that such witness belongs to the restricted class of models that are coherent with the possible interactions with the environment.

On the side of syntactic extensions, instead, a first line of research focuses on logics for the analysis of *strategic ability*, in the setting of *multi-agent games*, such as ATL [1] and SL [17], [16]. These logics syntactically extend classic temporal logics, by means of suitable modal operators which quantify over agent strategies, in order to express properties about cooperation and competition among agents. In particular, these modalities allow for a selective quantifications over those computations that are precisely the result of an infinite play among the agents. A different line of syntactic extensions focuses on epistemic and dynamic logics, whose concern is reasoning about knowledge and its evolution. Knowledge is usually modeled by a set of modal relations between information states. These relations are referred to in the syntax of the logics by means of corresponding modal operators. Two very interesting examples of this research vein are represented by the *logic of public announcement* [10], [19] and *sabotage logic* [24], both of which contain operators able to select and predicate on parts of the model under exam. These two languages can be also seen as logics about dynamically changing structures.

Although all described extensions have been introduced for quite different purposes, they all share a characterizing common factor: they extend the underlying temporal logic by means of specific features, which allow to extract and analyze portions of the model of interest. In other words, these logics permit to verify specific requirements over particular substructures either of the original model or of its unwinding.

For example, CTL with fairness allows to predicate on the substructure of the model unwinding containing only those paths that are fair w.r.t. a given constraint. Module checking requires the verification of a given branching-time temporal formula on all the substructures obtained by a pruning of possible actions executable by the environment from the whole interaction module between the system and the environment. Reactive synthesis deals with the extraction of a deterministic program as a suitable substructure of the computation tree modeling the possible dependences between input and output signals, which satisfies a given specification. The strategy quantifiers available in almost all logics to reasoning about multi-agent games essentially extract and analyze substructures of the game structure that are coherent with the chosen strategy. Epistemic and dynamic logics, instead, usually deal with substructures of the multi-modal model, each containing a subset of the knowledge relations. In particular, the concept of substructure is a crucial element in the semantics of the logic of public announcement and sabotage logic, and it is explicit in the definition of the interpretations of their characterizing modal operators.

In this paper, we propose and study a new logic, called *substructure temporal logic* (STL*, for short), in which it is possible to predicate directly over substructures of a model. In particular, the underlying semantics is defined by means of a two-layer interpretation, in which a classic temporal structure $\mathcal{K}$ is coupled with a higher-level modal layer. The elements of the higher-level layer are the substructures of $\mathcal{K}$ and its modal relation coincides with the partial order on these substructures. The syntactic counterpart is represented by two new syntactic constructs, called *semilattice operators*, provided to switch reasoning between the two different levels. The semantics of the semilattice operators resembles the semantics of the classic until and release temporal operators, except for the fact that it is defined on the lattice induced by the substructure relation. With more details, each operator first selects one of the substructures of the original model and then proceeds by verifying a specified temporal property on that substructure. In other words, the selection process performs the shift from the lower semantic layer to the higher one, while the verification process performs the inverse shift. In order to have a finer control on what and how much information of the original structure must be preserved by the substructures of interest, an additional parameter of the semilattice operators, called *selector parameter*, is provided. This parameter allows to select as elements of the semilattice precisely those substructures preserving the desired information.

The resulting logic turns out to be very expressive, allowing to encode in a uniform way most of the additional features proposed in the literature to reason about portions of the original model. In this perspective, the logic can be viewed as a first step towards providing a unifying framework, encompassing those previous approaches. Depending upon the class of structures on which the logic is interpreted, decision problems for the logic differ in complexity. While the satisfiability problem for the logic is undecidable when interpreted over Kripke structures, it becomes decidable in non-elementary time when interpreted over infinite regular trees. On the other hand, the model checking problem is decidable under both interpretations, being decidable in PSPACE and in non-elementary time, respectively.

Organization: The paper is organized as follows. Section II provides some basic definitions and the underlying semantic framework for the logic. The syntax and the semantics are presented in Section III, where some basic properties are discussed as well. Section IV focuses on some concrete applications, by showing that module checking, turn-based and concurrent games, and reactive synthesis can all be captured very naturally within the logic. Sections V and VI are devoted, instead, to a theoretical account of the formal properties of the logic. In particular, expressiveness, succinctness and (un)decidability results for STL* and for some of its fragments are reported and discussed. Finally, some conclusions and future work are proposed.

II. PRELIMINARIES

***Basic definitions**:* A *Kripke structure* (KS, for short) over a finite non-empty set of *atomic propositions* AP is a tuple $\mathcal{K} \triangleq \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_0 \rangle \in \mathrm{KS}(\mathrm{AP})$, where W is an enumerable non-empty set of *worlds*, $w_0 \in \mathrm{W}$ is a designated *initial world*, $R \subseteq \mathrm{W} \times \mathrm{W}$ is a left-total *transition relation* such that $R^*(w_0) = \mathrm{W}$, i.e., each world is reachable from the initial one, and $\mathsf{L} : \mathrm{W} \mapsto 2^{\mathrm{AP}}$ is a *labeling function* mapping each world to the set of atomic propositions true in that world. By $\mathcal{K}_w \triangleq \langle \mathrm{AP}, \mathrm{W}', R \cap (\mathrm{W}' \times \mathrm{W}'), \mathsf{L}_{\restriction \mathrm{W}'}, w \rangle$ we denote the KS obtained from $\mathcal{K}$ by substituting its initial world with the given one $w \in \mathrm{W}$, its set of states with $\mathrm{W}' \triangleq R^*(w)$, and its labeling function with the related restriction to W'. Observe that there is no loss of generality in requiring the reachability constraint on the transition relation, due to the fact that all parts that are not reachable from the initial world do not affect the satisfiability of a temporal formula.

A *track* (resp., *path*) in $\mathcal{K}$ is a finite (resp., infinite) sequence of worlds $\rho \in \mathrm{Trk} \subseteq \mathrm{W}^{+}$ (resp., $\pi \in \mathrm{Pth} \subseteq \mathrm{W}^{\omega}$) such that *(i)* $\mathsf{fst}(\rho) = w_0$ (resp., $\mathsf{fst}(\pi) = w_0$) and *(ii)*, for all $i \in [0, |\rho| - 1[$ (resp., $i \in \mathbb{N}$), it holds that $((\rho)_i, (\rho)_{i+1}) \in R$ (resp., $((\pi)_i, (\pi)_{i+1}) \in R$). Intuitively, tracks (resp., paths) of a Ks $\mathcal{K}$ are legal sequences of reachable worlds that can be seen as partial (resp., complete) descriptions of possible *computations* of the system modeled by $\mathcal{K}$. Given a track ρ (resp., path π), we denote by $(\rho)_{\leq j}$ and $(\rho)_{\geq j}$ (resp., $(\pi)_{\leq j}$ and $(\pi)_{\geq j}$) the prefix up to and the suffix from position $j \in [0, |\rho|[$ (resp., $j \in \mathbb{N}$).

In the following, we use the name of a Ks as subscript to extract the components from its tuple-structure, i.e., if $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_0 \rangle$, we have $\mathrm{W}_{\mathcal{K}} \triangleq \mathrm{W}$, $R_{\mathcal{K}} \triangleq R$, $\mathsf{L}_{\mathcal{K}} \triangleq \mathsf{L}$, and $w_{0\mathcal{K}} \triangleq w_0$. Also, we use the same notational concept to make explicit to which Ks the sets Trk and Pth are related to. Note that, we may omit the subscripts, if the Ks can be identified from the context.

A *Kripke tree* (Kt, for short) over AP is just a Ks $\mathcal{T} \in \mathrm{KT}(\mathrm{AP}) \subset \mathrm{KS}(\mathrm{AP})$, where *(i)* $\mathrm{W}_{\mathcal{T}} \subseteq \Delta^{*}$ is a Δ-tree for a set Δ of directions, *(ii)* $w_{0\mathcal{T}} = \varepsilon$, and *(iii)*, for all $t \in \mathrm{W}_{\mathcal{T}}$ and $d \in \Delta$, it holds that $t \cdot d \in \mathrm{W}_{\mathcal{T}}$ iff $(t, t \cdot d) \in R_{\mathcal{T}}$.

The *unwinding* of a Ks $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$ is the unique Kt $\mathcal{K}^{U} \in \mathrm{KT}(\mathrm{AP})$, where *(i)* $\mathrm{W}_{\mathcal{K}}$ is the set of its directions, *(ii)* its worlds in $\mathrm{W}_{\mathcal{K}^U} \triangleq \{(\rho)_{\geq 1} : \rho \in \mathrm{Trk}_{\mathcal{K}}(w_{0\mathcal{K}})\}$ are the suffixes of the tracks of $\mathcal{K}$ starting in the successors of $w_{0\mathcal{K}}$, *(iii)* $(\rho, \rho \cdot w) \in R_{\mathcal{K}^U}$ iff $(\mathsf{lst}(w_{0\mathcal{K}} \cdot \rho), w) \in R_{\mathcal{K}}$, and *(iv)* there is a surjective function $\mathsf{unw} : \mathrm{W}_{\mathcal{K}^U} \to \mathrm{W}_{\mathcal{K}}$, called *unwinding function*, such that *(v.i)* $\mathsf{unw}(\rho) = \mathsf{lst}(w_{0\mathcal{K}} \cdot \rho)$ and *(v.ii)* $\mathsf{L}'(\rho) = \mathsf{L}(\mathsf{unw}(\rho))$, for all $\rho \in \mathrm{W}_{\mathcal{K}^U}$ and $w \in \mathrm{W}_{\mathcal{K}}$.

In Figure 1, we depict a Ks $\mathcal{K}$ over $\mathrm{AP} \triangleq \{$●, ■, ◆$\}$ and its unwinding $\mathcal{K}^{U}$, which we use as running example in the whole paper. Note that we assume all worlds in $\mathcal{K}$ to be labeled by their own shapes. Therefore, AP is the set of $\mathcal{K}^{U}$ directions too. Also, the labeling of all worlds in $\mathcal{K}^{U}$ is the last symbol appearing in their names, except for the root, whose labeling coincides with that of the initial world of $\mathcal{K}$.

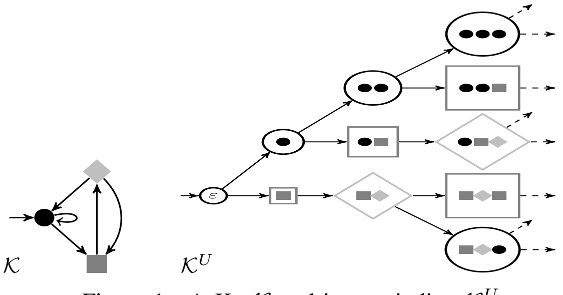

Figure 1: A Ks $\mathcal{K}$ and its unwinding $\mathcal{K}^{U}$.

Substructure semilattice: At the base of the semantics definition of the logic is the concept of ordering between Kss. Let $\mathcal{K}, \mathcal{K}' \in \mathrm{KS}(\mathrm{AP})$ be two Kss. We say that $\mathcal{K}$ is a *superstructure* of $\mathcal{K}'$ and $\mathcal{K}'$ is a *substructure* of $\mathcal{K}$, in symbols $\mathcal{K}' \sqsubseteq \mathcal{K}$, if *(i)* $\mathrm{W}_{\mathcal{K}'} \subseteq \mathrm{W}_{\mathcal{K}}$, *(ii)* $R_{\mathcal{K}'} \subseteq R_{\mathcal{K}} \cap (\mathrm{W}_{\mathcal{K}'} \times \mathrm{W}_{\mathcal{K}'})$, *(iii)* $\mathsf{L}_{\mathcal{K}'} = (\mathsf{L}_{\mathcal{K}})_{\restriction \mathrm{W}_{\mathcal{K}'}}$, and *(iv)* $\mathsf{w}_{0\mathcal{K}'} = \mathsf{w}_{0\mathcal{K}}$. Moreover, $\mathcal{K}$ and $\mathcal{K}'$ are *comparable* if *(i)* $\mathcal{K} \sqsubseteq \mathcal{K}'$ or *(ii)* $\mathcal{K}' \sqsubseteq \mathcal{K}$ holds, otherwise they are *incomparable*. Observe that $\sqsubseteq$ represents a *partial order* on Kss, whose *strict version*, denoted by $\sqsubset$, is such that $\mathcal{K}' \sqsubset \mathcal{K}$ if $\mathcal{K}' \sqsubseteq \mathcal{K}$ and $\mathcal{K}' \neq \mathcal{K}$.

For a given set of Kss $\aleph \subseteq \mathrm{KS}(\mathrm{AP})$ and a Ks $\mathcal{K} \in \aleph$, we say that $\mathcal{K}$ is *minimal* in $\aleph$, or simply *minimal* in case $\aleph$ equals to $\mathrm{KS}(\mathrm{AP})$, if there is no Ks $\mathcal{K}' \in \aleph$ such that $\mathcal{K}' \sqsubset \mathcal{K}$. Observe that minimal elements w.r.t. $\sqsubseteq$ are just those Kss for which the only part reachable from the initial state is either a single lasso or an infinite chain. This implies that $\mathcal{K}$ is minimal iff $|\mathrm{Pth}_{\mathcal{K}}| = 1$.

In order to identify the particular set of substructures of interest on which we predicate in the logic, we introduce the notion of filtering of a Ks. Let $\mathrm{X} \subseteq \mathrm{W}_{\mathcal{K}}$ be a subset of worlds of a given Ks $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$. Then, by $\mathfrak{F}_{\mathcal{K}}(\mathrm{X}) \triangleq \{\mathcal{K}' \in \mathrm{KS}(\mathrm{AP}) : \mathcal{K}' \sqsubseteq \mathcal{K} \wedge \forall w \in \mathrm{W}_{\mathcal{K}'} \cap \mathrm{X} \,.R_{\mathcal{K}'}(w) = R_{\mathcal{K}}(w)\}$ we denote the *filtering* of $\mathcal{K}$ w.r.t. X, i.e., the set of substructures of $\mathcal{K}$ that preserve all edges exiting from worlds in X. The ordering $\sqsubseteq$ on $\mathfrak{F}_{\mathcal{K}}(\mathrm{X})$ induces an *upper semilattice* satisfying the following properties: *(i)* the *maximal element* is $\mathcal{K}$; *(ii)* the *minimal elements* are exactly those Kss having a unique edge outgoing from states not in X; *(iii)* the *join* $\mathcal{K}_1 \sqcup \mathcal{K}_2$ of two elements $\mathcal{K}_1, \mathcal{K}_2 \in \mathfrak{F}_{\mathcal{K}}(\mathrm{X})$ is the Ks having set of worlds $\mathrm{W}_{\mathcal{K}_1 \sqcup \mathcal{K}_2} \triangleq \mathrm{W}_{\mathcal{K}_1} \cup \mathrm{W}_{\mathcal{K}_2}$, transition relation $R_{\mathcal{K}_1 \sqcup \mathcal{K}_2} \triangleq R_{\mathcal{K}_1} \cup R_{\mathcal{K}_2}$, and labeling function $\mathsf{L}_{\mathcal{K}_1 \sqcup \mathcal{K}_2} \triangleq (\mathsf{L}_{\mathcal{K}})_{\restriction \mathrm{W}_{\mathcal{K}_1 \sqcup \mathcal{K}_2}}$. Also, observe that $|\mathfrak{F}_{\mathcal{K}}(\mathrm{X})| = \infty$ iff one of the following conditions hold: *(i)* there is a world $w \in \mathrm{W}_{\mathcal{K}}$ having an infinite number of outgoing edges, i.e., $|R_{\mathcal{K}}(w)| = \omega$ or *(ii)* there are infinitely many worlds with at least two outgoing edges, i.e., $|\{w \in \mathrm{W}_{\mathcal{K}} : |R_{\mathcal{K}}(w)| \geq 2\}| = \omega$.

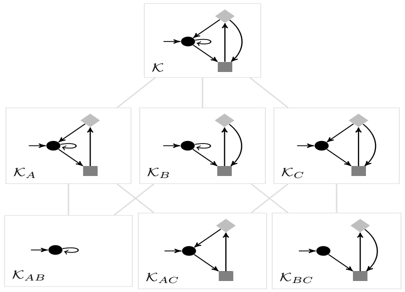

Figure 2: The substructure semilattice $\mathfrak{F}_{\mathcal{K}}(\emptyset)$.

In Figure 2, we depict the Hasse diagram of the finite semilattice of the substructures in the filtering of $\mathcal{K}$ w.r.t. $\emptyset$. In addition, in Figure 3, we report the diagrams of the sub-semilattices obtained by restricting the order to two smaller filterings. Note that no edge that is the unique outgoing one from a state (e.g., ■ in $\mathcal{K}$ or ● in $\mathcal{K}_{BC}$) can be pruned, otherwise the left-totality constraint of the transition relation would be violated. Moreover, by removing only the edge from ● to ■ in $\mathcal{K}$, we obtain a structure that is not a Ks, as the reachability constraint is violated. Note that $\mathcal{K}_{AB}$

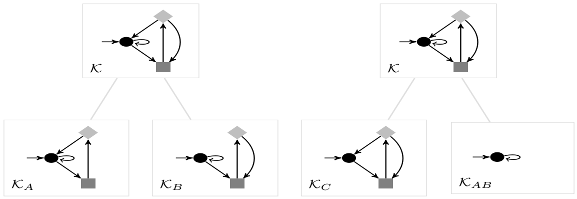

Figure 3: The two different filterings of $\mathcal{K}$ w.r.t. $\{\bullet\}$ and $\{\blacklozenge\}$.

belongs to the filtering $\mathfrak{F}_{\mathcal{K}}(\{\blacklozenge\})$, since it does not contain the state $\blacklozenge$, thus, the defining constraint of $\mathfrak{F}_{\mathcal{K}}(\{\blacklozenge\})$ is trivially satisfied.

III. Substructure Temporal Logics

The *substructure temporal logic* (STL*, for short) extends CTL* [8] by using two special ternary constructs, $\varphi_1 \mathbb{U}[\phi]\varphi_2$ and $\varphi_1 \mathbb{R}[\phi]\varphi_2$, called *semilattice operators*. These constructs can be informally read, respectively, as "there is a strict substructure satisfying φ_2 such that all its strict superstructures satisfy φ_1" and "all strict substructures satisfy φ_2 unless one of their strict superstructures satisfies φ_1", where the formula ϕ, called *selector parameter*, specifies the particular semilattice of substructures on which the quantifications act. Specifically, this parameter is used to identify on which worlds of the original model the pruning is forbidden. From an high level point of view, we can consider these new operators as a strict version of the until and release temporal operators acting on substructures and their partial order instead of linear points in time. As in CTL*, in STL* the path quantifiers E and A can prefix a linear-time formula composed by an arbitrary Boolean combination and nesting of temporal operators X, U, and R.

Syntax: The formal syntax of STL* is defined as follows.

Definition III.1 (STL* Syntax)**.** STL* state *(φ) and* path *(ψ)* formulas *are built inductively from the set* AP *according to the following grammar, where $p \in$ AP:*

1) $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi\mathbb{U}[\varphi]\varphi \mid \varphi\mathbb{R}[\varphi]\varphi \mid \mathsf{E}\psi \mid \mathsf{A}\psi$*;*
2) $\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathsf{X}\psi \mid \psi\mathsf{U}\psi \mid \psi\mathsf{R}\psi$*.*

Simpler STL$^+$ *and* STL *formulas are obtained by forbidding, respectively, nesting and both nesting and Boolean combinations of temporal operators, as in* CTL$^+$ *and* CTL*.*

In the following, as syntactical abbreviations, we use the Boolean values true $\mathfrak{t}$ and false $\mathfrak{f}$ and the simpler temporal operators eventually $\mathsf{F}\varphi \triangleq \mathfrak{t}\,\mathsf{U}\varphi$ and globally $\mathsf{G}\varphi \triangleq \mathfrak{f}\,\mathsf{R}\varphi$. We shall define the restricted constructs $\mathbb{EX}[\phi]\varphi \triangleq \mathfrak{f}\,\mathbb{U}[\phi]\varphi$, $\mathbb{AX}[\phi]\varphi \triangleq \mathfrak{t}\,\mathbb{R}[\phi]\varphi$, called *immediate substructure operators*, and the operators $\mathbb{F}[\phi]\varphi \triangleq \mathfrak{t}\,\mathbb{U}[\phi]\varphi$ and $\mathbb{G}[\phi]\varphi \triangleq \mathfrak{f}\,\mathbb{R}[\phi]\varphi$. In addition, we can derive the reflexive versions of the operators as follows: $\varphi_1\overline{\mathbb{U}}[\phi]\varphi_2 \triangleq \varphi_2 \vee (\varphi_1 \wedge \varphi_1\mathbb{U}[\phi]\varphi_2)$, $\varphi_1\overline{\mathbb{R}}[\phi]\varphi_2 \triangleq \varphi_2 \wedge (\varphi_1 \vee \varphi_1\mathbb{R}[\phi]\varphi_2)$, $\overline{\mathbb{F}}[\phi]\varphi \triangleq \varphi \vee \mathbb{F}[\phi]\varphi$, and $\overline{\mathbb{G}}[\phi]\varphi \triangleq \varphi \wedge \mathbb{G}[\phi]\varphi$. Sometimes, we omit the selector parameter ϕ, whenever it equals to $\mathfrak{f}$, in all semilattice operators, as well as in the derived ones later introduced.

By replacing the two constructs $\varphi\mathbb{U}[\phi]\varphi$ and $\varphi\mathbb{R}[\phi]\varphi$ with the simpler operators $\mathbb{F}[\phi]\varphi$ and $\mathbb{G}[\phi]\varphi$, in Rule 1 of Definition III.1, we obtain a family of sublogics of STL* called *weak substructure temporal logics* (WSTL*, WSTL$^+$, and WSTL, for short).

Semantics: We shall write $\mathcal{K} \models \varphi$ to denote that a state formula φ holds in $\mathcal{K}$ or, equivalently, $\mathcal{K}$ is a *model* of φ. Moreover, for a path $\pi \in \mathrm{Pth}_{\mathcal{K}}$ and a number $k \in \mathbb{N}$, we write $\mathcal{K}, \pi, k \models \psi$ to indicate that a path formula ψ holds on π at position k. The semantics of STL* formulas, except for the new lattice operators, is defined as usual for CTL* and, for sake of space, is omitted here. The formal semantics of $\varphi_1\mathbb{U}[\phi]\varphi_2$ and $\varphi_1\mathbb{R}[\phi]\varphi_2$ follows.

Definition III.2 (STL* Semantics)**.** *Given a* KS $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$*, for all* STL* *state formulas φ_1, φ_2, and ϕ, it holds that:*

1) $\mathcal{K} \models \varphi_1\mathbb{U}[\phi]\varphi_2$ *if there exists a $\mathcal{K}' \in \mathfrak{S}_{\mathcal{K}}(\phi)$ such that $\mathcal{K}' \models \varphi_2$ and, for all strict superstructures $\mathcal{K}'' \in \mathfrak{S}_{\mathcal{K}}(\phi)$ of $\mathcal{K}'$, it holds that $\mathcal{K}'' \models \varphi_1$;*
2) $\mathcal{K} \models \varphi_1\mathbb{R}[\phi]\varphi_2$ *if, for all $\mathcal{K}' \in \mathfrak{S}_{\mathcal{K}}(\phi)$, it holds that $\mathcal{K}' \models \varphi_2$ or there exists a strict superstructure $\mathcal{K}'' \in \mathfrak{S}_{\mathcal{K}}(\phi)$ of $\mathcal{K}'$ such that $\mathcal{K}'' \models \varphi_1$;*

where $\mathfrak{S}_{\mathcal{K}}(\phi) \triangleq \mathfrak{F}_{\mathcal{K}}(\phi) \setminus \{\mathcal{K}\}$ and $\mathfrak{F}_{\mathcal{K}}(\phi) \triangleq \mathfrak{F}_{\mathcal{K}}(\{w \in \mathrm{W}_{\mathcal{K}} : \mathcal{K}_w \models \phi\})$ is the set of all $\mathcal{K}$ substructures preserving edges exiting from those worlds on which the formula ϕ is satisfied.

Observe that, by replacing the set $\mathfrak{S}_{\mathcal{K}}(\phi)$ with $\mathfrak{F}_{\mathcal{K}}(\phi)$, in Items 1 and 2 of the previous definition, we obtain the semantics of reflexive operators $\varphi_1\overline{\mathbb{U}}[\phi]\varphi_2$ and $\varphi_1\overline{\mathbb{R}}[\phi]\varphi_2$.

To better understand the intuition behind the introduced semilattice operators, we present two examples based on the KS $\mathcal{K}$ of Figure 2 and its filtering $\mathfrak{F}_{\mathcal{K}}(\emptyset)$.

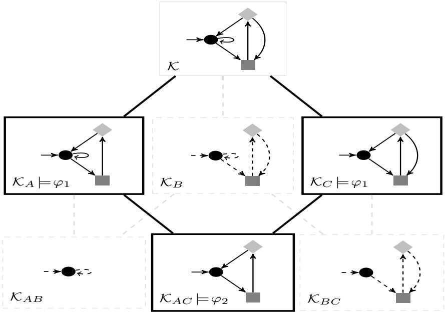
$\mathcal{K}$
$\mathcal{K}_A \models \varphi_1$
$\mathcal{K}_B$
$\mathcal{K}_C \models \varphi_1$
$\mathcal{K}_{AB}$
$\mathcal{K}_{AC} \models \varphi_2$
$\mathcal{K}_{BC}$

Figure 4: $\mathbb{U}$ semantics.

Consider the formula $\varphi_1\mathbb{U}\varphi_2$, where $\varphi_1 \triangleq \mathsf{AGEF}\bullet$, and $\varphi_2 \triangleq (\mathsf{AGF}\blacklozenge) \wedge ((\mathsf{AGF}\bullet) \vee (\mathsf{AFG}\neg\bullet))$. Intuitively, φ_1 is true on all KSs containing only paths from whose states it is possible to reach eventually $\bullet$, while φ_2 is verified on all KSs for which all paths contain infinitely often $\blacklozenge$ and either all of them also contain infinitely often $\bullet$ or they all do not. It is easy to see that $\mathcal{K}_{AC}$ satisfies φ_2 and both $\mathcal{K}_A$ and $\mathcal{K}_C$ satisfy φ_1. Thus, as depicted in Figure 4 (we highlight the witness by using solid bold lines), we have that $\mathcal{K} \models \varphi_1\mathbb{U}\varphi_2$. Indeed, there exists a strict substructure ($\mathcal{K}_{AC}$) of $\mathcal{K}$ satisfying φ_2 such that all its strict superstructures

($\mathcal{K}_A$ and $\mathcal{K}_C$) satisfy φ_1. Observe that this is the unique witness for the required property on $\mathcal{K}$, since the only other substructure ($\mathcal{K}_{BC}$) satisfying φ_2 has a strict superstructure ($\mathcal{K}_B$) that does not satisfy φ_1. Also, note that $\mathcal{K} \not\models \varphi_1 \mathbb{U}[\bullet] \varphi_2$ and $\mathcal{K} \not\models \varphi_1 \mathbb{U}[\blacklozenge] \varphi_2$, since in the corresponding filterings $\mathfrak{F}_{\mathcal{K}}(\bullet)$ and $\mathfrak{F}_{\mathcal{K}}(\blacklozenge)$ there is no Ks satisfying φ_2.

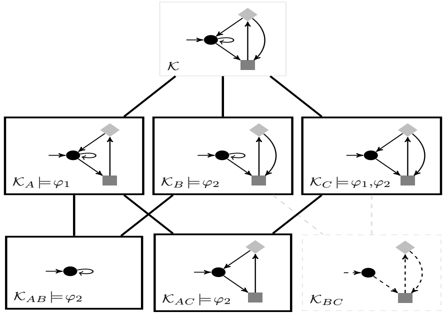
$\mathcal{K}$
$\mathcal{K}_A \models \varphi_1$
$\mathcal{K}_B \models \varphi_2$
$\mathcal{K}_C \models \varphi_1, \varphi_2$
$\mathcal{K}_{AB} \models \varphi_2$
$\mathcal{K}_{AC} \models \varphi_2$
$\mathcal{K}_{BC}$

Figure 5: $\mathbb{R}$ semantics.

Consider the formula $\varphi_1 \mathbb{R} \varphi_2$, where $\varphi_1 \triangleq \mathsf{AF}\blacklozenge$ and $\varphi_2 \triangleq \mathsf{EGF}\bullet$. Intuitively, φ_1 is true on all Kss in which every path reaches eventually $\blacklozenge$, while φ_2 is verified on all Kss containing a path visiting infinitely often $\bullet$. It is easy to see that all Kss in $\mathfrak{F}_{\mathcal{K}}(\emptyset)$ but $\mathcal{K}_{BC}$ satisfy φ_2 and $\mathcal{K}_C$ also satisfies φ_1. Thus, as depicted in Figure 5, we have that $\mathcal{K} \models \varphi_1 \mathbb{R} \varphi_2$. Indeed, the only strict substructure ($\mathcal{K}_{BC}$) of $\mathcal{K}$ not satisfying φ_2 has a strict superstructure ($\mathcal{K}_C$) satisfying φ_1.

Basic concepts: We say that a formula φ is an *invariant* for two Kss $\mathcal{K}_1$ and $\mathcal{K}_2$ whenever $\mathcal{K}_1 \models \varphi$ iff $\mathcal{K}_2 \models \varphi$. For a given set of Kss $\aleph \subseteq \mathrm{KS}(\mathrm{AP})$, we say that φ is $\aleph$-*satisfiable* if there is a Ks $\mathcal{K} \in \aleph$ such that $\mathcal{K} \models \varphi$. Furthermore, for all state formulas φ_1 and φ_2, we say that φ_1 $\aleph$-*implies* φ_2, in symbols $\varphi_1 \Rightarrow_{\aleph} \varphi_2$, if, for all Kss $\mathcal{K} \in \aleph$, it holds that if $\mathcal{K} \models \varphi_1$ then $\mathcal{K} \models \varphi_2$, i.e., φ_2 is an $\aleph$-*consequence* of φ_1. Also, we say that φ_1 is $\aleph$-*equivalent* to φ_2, in symbols $\varphi_1 \equiv_{\aleph} \varphi_2$, if $\varphi_1 \Rightarrow_{\aleph} \varphi_2$ and $\varphi_2 \Rightarrow_{\aleph} \varphi_1$.

In the remaining part of the work, we use the symbol STL*[$\aleph$] to denote to which set of Kss $\aleph \subseteq \mathrm{KS}(\mathrm{AP})$ the interpretation of formulas has to be restricted to. The notion of satisfiability and model checking relative to a given class $\aleph$ are defined in the obvious way. Whenever $\aleph$ coincides with KS(AP) (resp., KT(AP)) we shall use the corresponding symbol Ks (resp., Kt) instead.

Given the similarity between the semilattice operators and the linear temporal operators, it is worth discussing what equivalences from classic Ltl holds for the new operators. The first series of equivalences describes the fixpoint semantics of the reflexive operators: *(i)* $\varphi_1 \overline{\mathbb{U}}[\phi] \varphi_2 \equiv \varphi_2 \vee \varphi_1 \wedge \mathbb{EX}[\phi] \varphi_1 \overline{\mathbb{U}}[\phi] \varphi_2$; *(ii)* $\varphi_1 \overline{\mathbb{R}}[\phi] \varphi_2 \equiv \varphi_2 \wedge (\varphi_1 \vee \mathbb{AX}[\phi] \varphi_1 \overline{\mathbb{R}}[\phi] \varphi_2)$; *(iii)* $\overline{\mathbb{F}}[\phi] \varphi \equiv \varphi \vee \mathbb{EX}[\phi] \overline{\mathbb{F}}[\phi] \varphi$; *(iv)* $\overline{\mathbb{G}}[\phi] \varphi \equiv \varphi \wedge \mathbb{AX}[\phi] \overline{\mathbb{G}}[\phi] \varphi$. The second series describes the relation between the strict operators and the immediate substructure operators: *(i)* $\varphi_1 \mathbb{U}[\phi] \varphi_2 \equiv \mathbb{EX}[\phi] \varphi_1 \overline{\mathbb{U}}[\phi] \varphi_2$; *(ii)* $\varphi_1 \mathbb{R}[\phi] \varphi_2 \equiv \mathbb{AX}[\phi] \varphi_1 \overline{\mathbb{R}}[\phi] \varphi_2$; *(iii)* $\mathbb{F}[\phi] \varphi \equiv \mathbb{EX}[\phi] \overline{\mathbb{F}}[\phi] \varphi$; *(iv)* $\mathbb{G}[\phi] \varphi \equiv \mathbb{AX}[\phi] \overline{\mathbb{G}}[\phi] \varphi$. Notice however that, due to the branching nature of semilattice operators, the classic equivalence $\psi_1 \mathsf{R} \psi_2 \equiv \mathsf{G} \psi_2 \vee \psi_2 \mathsf{U}(\psi_1 \wedge \psi_2)$, linking together the three Ltl temporal operators R, G, and U, lifts neither to $\mathbb{R}$, $\mathbb{G}$, and $\mathbb{U}$ nor to $\overline{\mathbb{R}}$, $\overline{\mathbb{G}}$, and $\overline{\mathbb{U}}$. For instance, consider the formulas φ_1 and φ_2 of the example for Figure 4. It is easy to see that $\mathcal{K} \models \varphi_1 \mathbb{U}(\varphi_1 \wedge \varphi_2)$, since $\mathcal{K}_{AD}$ satisfies φ_1 too. However, $\mathcal{K} \not\models \varphi_2 \mathbb{R} \varphi_1$, since $\mathcal{K}_B$ does not satisfy φ_1. The same holds for the reflexive version of the two operators. Therefore, we have that $\varphi_1 \mathbb{R}[\phi] \varphi_2 \not\equiv \mathbb{G}[\phi] \varphi_2 \vee \varphi_2 \mathbb{U}[\phi](\varphi_1 \wedge \varphi_2)$ and $\varphi_1 \overline{\mathbb{R}}[\phi] \varphi_2 \not\equiv \overline{\mathbb{G}}[\phi] \varphi_2 \vee \varphi_2 \overline{\mathbb{U}}[\phi](\varphi_1 \wedge \varphi_2)$.

Interesting properties: Before moving to discuss the applications, let us introduce some interesting properties expressible in STL* that cannot, as we shall see later on, be expressed in CTL*.

The simplest concept we can describe using STL* is the *absolute minimality* of a Ks $\mathcal{K}$ w.r.t. a given specification φ and an assigned selector parameter ϕ. More formally, we want to specify the property of $\mathcal{K}$ being minimal in the set $\{\mathcal{K}' \in \mathfrak{F}_{\mathcal{K}}(\phi) : \mathcal{K}' \models \varphi\}$, i.e., that $\mathcal{K}$ is the unique element of its filtering $\mathfrak{F}_{\mathcal{K}}(\phi)$ that satisfies φ. To express this concept, we introduce the following construct: $\mathtt{Min}_{\phi}(\varphi) \triangleq \varphi \wedge \mathbb{G}[\phi] \neg \varphi$. Then, $\mathcal{K} \models \mathtt{Min}_{\phi}(\varphi)$ iff $\mathcal{K}$ satisfies φ and none of its substructure does. So, $\mathcal{K}$ is minimal w.r.t. φ in the semilattice selected by ϕ. Note that, if both φ and ϕ belong to any of the weak sublogics of STL*, $\mathtt{Min}_{\phi}(\varphi)$ does as well.

By nesting the minimality construct within the simple semilattice operators $\overline{\mathbb{F}}$ and $\overline{\mathbb{G}}$, we can also predicate on minimal substructures of a given Ks. We call this property *relative minimality*. In particular, given two formulas φ_1 and φ_2, we can assert the existence of a minimal substructure w.r.t. φ_1 that satisfies φ_2, or that all minimal substructures w.r.t. φ_1 have to satisfy φ_2. These concepts can be expressed by the following constructs: $\mathtt{EMin}_{\phi}(\varphi_1, \varphi_2) \triangleq \overline{\mathbb{F}}[\phi](\mathtt{Min}_{\phi}(\varphi_1) \wedge \varphi_2)$ and $\mathtt{AMin}_{\phi}(\varphi_1, \varphi_2) \triangleq \overline{\mathbb{G}}[\phi](\mathtt{Min}_{\phi}(\varphi_1) \rightarrow \varphi_2)$. Intuitively, we have that $\mathcal{K} \models \mathtt{EMin}_{\phi}(\varphi_1, \varphi_2)$ iff there exists a substructure $\mathcal{K}'$ of $\mathcal{K}$, which is minimal w.r.t. φ_1 in the semilattice selected by ϕ, such that $\mathcal{K}' \models \varphi_2$. Similarly, we have that $\mathcal{K} \models \mathtt{AMin}_{\phi}(\varphi_1, \varphi_2)$ iff for all substructures $\mathcal{K}'$ of $\mathcal{K}$, which are minimal w.r.t. φ_1 in the semilattice selected by ϕ, it holds that $\mathcal{K}' \models \varphi_2$. Observe that these two constructs are dual of each other, i.e., $\neg \mathtt{EMin}_{\phi}(\varphi_1, \varphi_2) \equiv \mathtt{AMin}_{\phi}(\varphi_1, \neg \varphi_2)$. Once again, note that if φ_1, φ_2, and ϕ belong to any of the weak sublogics of STL*, the same holds of $\mathtt{EMin}_{\phi}(\varphi_1, \varphi_2)$ and $\mathtt{AMin}_{\phi}(\varphi_1, \varphi_2)$.

It is interesting to see that, while it makes sense to speak about the absolute minimality of a Ks in its filtering w.r.t. a given formula, the symmetric notion of absolute maximality of a Ks in its filtering is a trivial one, as it clearly boils down to verify the argument formula on the Ks itself.

By using the semilattice operators $\overline{\mathbb{U}}$ and $\overline{\mathbb{R}}$, we can express the symmetric notion of *relative maximality*, namely the existence of a maximal substructure w.r.t. φ_1 that satisfies φ_2, or that all maximal substructures w.r.t. φ_1 have to satisfy φ_2. Such concepts can be expressed by the following constructs: $\mathtt{EMax}_{\phi}(\varphi_1, \varphi_2) \triangleq (\neg\varphi_1)\overline{\mathbb{U}}[\phi](\varphi_1 \wedge \varphi_2)$ and $\mathtt{AMax}_{\phi}(\varphi_1, \varphi_2) \triangleq (\varphi_1)\overline{\mathbb{R}}[\phi](\varphi_1 \rightarrow \varphi_2)$. Intuitively, we have that $\mathcal{K} \models \mathtt{EMax}_{\phi}(\varphi_1, \varphi_2)$ iff there exists a substructure $\mathcal{K}'$ of $\mathcal{K}$ that is the maximal one satisfying both φ_1 and φ_2, since it does not have any superstructure satisfying φ_1 too. Similarly, we have that $\mathcal{K} \models \mathtt{AMax}_{\phi}(\varphi_1, \varphi_2)$ iff all substructures $\mathcal{K}'$ of $\mathcal{K}$ satisfying φ_1 either satisfy φ_2 or have at least one superstructure that satisfies φ_1. In the latter case, $\mathcal{K}'$ is not maximal w.r.t. φ_1, thus, we do not have to verify any further requirement on it. Observe that, also in this case, a duality law holds, i.e., $\neg\mathtt{EMax}_{\phi}(\varphi_1, \varphi_2) \equiv \mathtt{AMax}_{\phi}(\varphi_1, \neg\varphi_2)$. Moreover, note that these two constructs cannot belong to any of the weak sublogics of STL*, since they strictly require the use of $\overline{\mathbb{U}}$ and $\overline{\mathbb{R}}$. It is important to observe that the semantics of the latter operators cannot be reformulated using the simpler $\mathbb{F}$ and $\mathbb{G}$, exactly as in the classic case of LTL, where temporal operators U and R cannot be expressed by F, G, and X, only.

IV. Inspiring Applications

A distinguishing feature of STL* is the ability to quantify over substructures and to express (relative) minimality and maximality properties. In this section we show how these features allow to encode in the logic a number of relevant problems arose in the literature.

Module checking: In open finite-state system model checking (module checking, for short) [15], we check whether a system interacting with an external component, the environment, is correct with respect to a desired behavior. In this setting, we formally represent the system and its possible interactions with the environment by a *module*, i.e., a KS $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_0 \rangle$, where the set of worlds $\mathrm{W} \triangleq \mathrm{W}_1 \cup \mathrm{W}_2$ is partitioned into two components: W_1 contains all and only the worlds labeled by the ad-hoc atomic proposition $1 \in \mathrm{AP}$, representing the positions where the system is allowed to take a move, i.e., *system worlds*, while the *environment worlds* are those in W_2 where the environment takes moves. Given a module $\mathcal{K}$ and a CTL* specification φ, the module checking problem is to check whether $\mathcal{K}$ satisfies φ no matter how the environment behaves. Let us consider the unwinding $\mathcal{K}^U$ of $\mathcal{K}$. Checking whether $\mathcal{K}^U$ satisfies φ is the usual model-checking problem. On the other hand, for an open system, $\mathcal{K}^U$ describes the interaction of the system with a maximal environment, i.e. an environment that enables all the external nondeterministic choices. To take into account all possible behaviors of the environment, we consider all the trees $\mathcal{T}$ obtained from $\mathcal{K}^U$ by pruning subtrees whose roots are successors of an environment world. Then, a module $\mathcal{K}^U$ satisfies φ if all these trees $\mathcal{T}$ satisfy φ. The set of these trees coincides with the filtering $\mathfrak{F}_{\mathcal{K}^U}(1)$, which preserves all the system choices. Hence, the module checking problem can be expressed in WSTL* by checking whether $\mathcal{K}^U$ satisfies the formula $\varphi_{MC}(\varphi) \triangleq \overline{\mathbb{G}}[1](\varphi)$.

Turn-based games: The arena of a two-player turn-based game can be formalized by means of a KS $\mathcal{K}$ as above, where W_i contains all and only the worlds where player i takes a move, for all $i \in \{1, 2\}$. Given such a turn-based arena, the notion of *strategy for player i*, with $i \in \{1, 2\}$, is typically defined as a function $\sigma_i : \mathrm{W}^*\mathrm{W}_i \rightarrow \mathrm{W}$ mapping sequences of worlds ending with one of W_i to worlds. A strategy σ_i induces a set of paths (the plays of the game), namely the *outcomes of* σ_i, compatible with that strategy. Formally, $\mathrm{Out}(\sigma_i) \triangleq \{\pi \in \mathrm{Pth}_{\mathcal{K}} : \forall j \in \mathbb{N} \,.\, (\pi)_j \in \mathrm{W}_i \rightarrow (\pi)_{j+1} = \sigma_i((\pi)_{\leq j})\}$. Intuitively, the outcomes of a strategy σ_i of player i are the plays of the game which agree with σ_i, while leaving the other player play according to any one of its possible response strategies. Finally, given an LTL requirement ψ, we say that a strategy σ_i for player i is *winning w.r.t.* ψ, if all the outcomes of σ_i satisfy ψ. The decision problem we consider is, therefore, to verify whether there exists a winning strategy for one player, say player 1, w.r.t. ψ. This can be encoded quite naturally in the WSTL* logic, by nesting an existential and a universal relative minimality constructs. Player 1 has a winning strategy (resp., memoryless winning strategy) for $\mathcal{K}$ iff $\mathcal{K}^U$ (resp., $\mathcal{K}$) satisfies the formula $\varphi_{TG}(\psi) \triangleq \mathtt{EMin}_{\neg 1}(\mathsf{t}, \mathtt{AMin}_1(\mathsf{t}, \mathsf{A}\psi))$. The existential minimal operator $\mathtt{EMin}_{\neg 1}$ selects a minimal substructure where all possible moves of Player 2 are preserved. Indeed, the selector $\neg 1$ allows only for substructures whose worlds non labeled by 1 retain all the outgoing edges of the original structure in the semilattice where the operator acts. This corresponds to a strategy of Player 1, in the sense that the set of paths of the substructure selected by the operator is exactly the set of outcomes induced by the strategy. Similarly, the universal minimal operator $\mathtt{AMin}_1$ selects all minimal substructures, which preserve the choices made by Player 1. This corresponds to selecting, in turn, all possible strategies that Player 2 can follow in response to the strategy of Player 1.

Concurrent games: Also in the case of two-player concurrent games, we can encode the corresponding arenas by means of KSs. However, the encoding is slightly more complicated, as explained below. Let Ac_1 and Ac_2 be the sets of possible actions the two players can take and assume that the set of atomic propositions AP contains the product $\mathrm{Ac}_1 \times \mathrm{Ac}_2$, representing all possible decisions. Then, a concurrent arena can be formalized as a KS $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_0 \rangle$, where, for each world $w \in \mathrm{W}$ and decision $(a_1, a_2) \in \mathrm{Ac}_1 \times \mathrm{Ac}_2$, there is exactly one successor $v \in R(w)$ of w with $(a_1, a_2) \in \mathsf{L}(v)$. Observe that the uniqueness of the successor for each decision is required to encode that the transition function of the game is deterministic. Given the concurrent arena, a *strategy for player i*, with $i \in \{1, 2\}$, is defined as a function $\sigma_i : \mathrm{W}^+ \rightarrow \mathrm{Ac}_i$ mapping sequences of worlds to actions. Accordingly, the set of *outcomes* compatible with a strategy σ_i is defined as follows: $\mathrm{Out}(\sigma_i) \triangleq \{\pi \in \mathrm{Pth}_{\mathcal{K}} : \forall j \in \mathbb{N} \,.\, \exists (a_1, a_2) \in \mathsf{L}((\pi)_{j+1}) \cap (\mathrm{Ac}_1 \times \mathrm{Ac}_2) \,.\, a_i = \sigma_i((\pi)_{\leq j})\}$. The concept of winning strategy and the related decision problem are exactly the same of those ones for the turn-based case. Now, to encode a quantification of a strategy by means of a suitable STL* formula, we exploit the following observations. First, a strategy σ_i identifies a substructure $\mathcal{T}_{\sigma_i}$ of $\mathcal{K}^U$ having, for each world $w \in \mathrm{W}_{\mathcal{T}_{\sigma_i}}$, only those successors $v \in R_{\mathcal{T}_{\sigma_i}}(w)$ for which exists a decision $(a_1, a_2) \in \mathsf{L}(v) \cap (\mathrm{Ac}_1 \times \mathrm{Ac}_2)$ such that $a_i = \sigma_i(w)$. Second, the CTL formula $\varphi_i \triangleq \mathsf{AG} \bigvee_{\mathsf{a}_i \in \mathrm{Ac}_i} \mathsf{AX} \bigvee_{\mathsf{a}_{3-i} \in \mathrm{Ac}_{3-i}} (\mathsf{a}_1, \mathsf{a}_2)$, with $i \in \{1, 2\}$, requires that, for every world w, there is an action of player i that allows to reach all its successors. Clearly, every maximal substructure of $\mathcal{K}^U$ satisfying φ_i preserves all the actions of the opponent. So, it corresponds to a substructure $\mathcal{T}_{\sigma_i}$ associated with the strategy σ_i. As a consequence, to verify whether there is a winning strategy for player 1 w.r.t. ψ, we can use a nesting of an existential and a universal relative maximality constructs. Finally, Player 1 has a winning strategy (resp., memoryless winning strategy) for $\mathcal{K}$ iff $\mathcal{K}^U$ (resp., $\mathcal{K}$) satisfies the formula $\varphi_{CG}(\psi) \triangleq \mathtt{EMax}(\varphi_1, \mathtt{AMax}(\varphi_2, \mathsf{A}\psi))$.

Reactive Synthesis: In the formulation proposed by Pnueli and Rosner in [21], the reactive synthesis problem consists of the construction of a *deterministic program* that interacts with an environment providing sets of input signals, of which some are visible and some are hidden to the program itself. Obviously, this program must respond to the inputs it can read, the visible ones, with some set of output signals. In other words, the problem is to synthesize a function $\mathsf{P} : \left(2^{\mathrm{I}}\right)^* \rightarrow 2^{\mathrm{O}}$ from finite sequences of (sets of) visible inputs to (sets of) outputs, if it exists. In addition, P must be such that the KT $\mathcal{T}_{\mathsf{P}}$ induced by its interaction with the environment also satisfies some given CTL* (or CTL) specification φ. If I denotes the set of possible visible inputs, H the set of hidden inputs and O the set of outputs, the KT $\mathcal{T}_{\mathsf{P}}$ of a solution program P to the above problem shall contain worlds labeled with sets of visible inputs $\Sigma_i \subseteq \mathrm{I}$ and hidden inputs $\Sigma_h \subseteq \mathrm{H}$ issued by the environment and sets of outputs $\Sigma_o \subseteq \mathrm{O}$ issued by the program P in response to the inputs received in that world.

To ensure that P behaves like a function, we need to enforce some additional requirements. Since P cannot read hidden inputs, given a world of the KT $\mathcal{T}_{\mathsf{P}}$ and two of its successors with the same set of visible inputs, but possibly different hidden inputs, it must be the case that P responds to them with the same set of outputs.

Condition 1: For all worlds $w \in \mathrm{W}_{\mathcal{T}_{\mathsf{P}}}$ and successors $v_1, v_2 \in R_{\mathcal{T}_{\mathsf{P}}}(w)$ with $\mathsf{L}(v_1) \cap \mathrm{I} = \mathsf{L}(v_2) \cap \mathrm{I}$, it holds that $\mathsf{L}(v_1) \cap \mathrm{O} = \mathsf{L}(v_2) \cap \mathrm{O}$.

However, Condition 1 is not enough to ensure that P is deterministic, hence a function, as it is still possible to have multiple copies of the same successor (with the same set of signals), which may have different future behaviors in response to the same visible inputs. If this is the case, P would be non-deterministic (see Figure 6). Therefore, we must also ensure that any world does not have more than one successor for each possible signal set.

Condition 2: For all worlds $w \in \mathrm{W}_{\mathcal{T}_{\mathsf{P}}}$ and successors $v_1, v_2 \in R_{\mathcal{T}_{\mathsf{P}}}(w)$, if $\mathsf{L}(v_1) = \mathsf{L}(v_2)$ then $v_1 = v_2$.

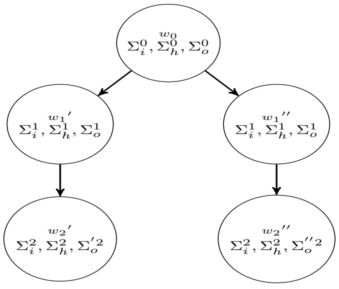
w_0
$\Sigma_i^0, \Sigma_h^0, \Sigma_o^0$
${w_1}'$
$\Sigma_i^1, \Sigma_h^1, \Sigma_o^1$
${w_1}''$
$\Sigma_i^1, \Sigma_h^1, \Sigma_o^1$
${w_2}'$
$\Sigma_i^2, \Sigma_h^2, {\Sigma'_o}^2$
${w_2}''$
$\Sigma_i^2, \Sigma_h^2, {\Sigma''_o}^2$

Figure 6: Violation of w_0 successors uniqueness.

The two conditions can be expressed in WSTL by means of the following formulas φ_1 and φ_2. For the sake of readability, we abuse the notation and write $\Sigma \subseteq \mathrm{AP}$ as an abbreviation for the conjunction of the atomic propositions in Σ. Similarly, $\overline{\Sigma}$ abbreviates the conjunction of the negations of atomic propositions in Σ.

Then, the formula $\varphi_1 \triangleq \mathsf{AG} \bigwedge_{\Sigma_i \subseteq \mathrm{I}} \bigvee_{\Sigma_o \subseteq \mathrm{O}} \overline{\mathbb{G}}(\mathsf{AX}(\Sigma_i \wedge \overline{\mathrm{I} \setminus \Sigma_i}) \rightarrow \mathsf{AX}(\Sigma_o \wedge \overline{\mathrm{O} \setminus \Sigma_o}))$ ensures that Condition 1 is satisfied in every reachable world of the KT $\mathcal{T}_{\mathsf{P}}$. Intuitively, it requires that, for every set of visible inputs Σ_i, there is a set of outputs Σ_o such that, in all the substructures (selected in turn by the operator $\overline{\mathbb{G}}$) of the KT rooted in the current world, the following holds: if all the successors of that world contain exactly the inputs in Σ_i then all of them must contain exactly the outputs in Σ_o.

Condition 2 can be expressed, instead, by the formula $\varphi_2 \triangleq \mathtt{Min}(\mathsf{AG} \bigwedge_{\Sigma \subseteq \mathrm{I} \cup \mathrm{H}} \mathsf{EX}(\Sigma \wedge \overline{(\mathrm{I} \cup \mathrm{H}) \setminus \Sigma}))$. The argument of $\mathtt{Min}$ guarantees that, for all reachable worlds, every set of inputs is contained in the labeling of some successor. In addition, the minimality required by the construct ensures that each such successor is unique w.r.t. that labeling.

Finally, the solution to the synthesis problem can be encoded as the WSTL* (or WSTL) formula $\varphi_{RS}(\varphi) \triangleq \varphi \wedge \varphi_1 \wedge \varphi_2$. Indeed, $\varphi_{RS}(\varphi)$ is satisfied by a KT $\mathcal{T}$ iff it satisfies the original CTL* (or CTL) requirement φ together with the two formulas encoding the conditions above.

V. Model-Theoretic Analysis

Let us now turn our attention to the formal properties of the logic and concentrate on a model theoretic analysis of the STL* semantics. The formal proofs of the results are omitted for the lack of space and reported in the extended version.

We first discuss the power of the logic in describing properties of the underlying semilattice of structures, such as density and discreteness, that can only be encoded in very expressive logics, such as MSOL [22] and the graded μCALCULUS [13], [2].

Density and discreteness: Let us consider the following STL formula: $\mathtt{Den}_\phi \triangleq \mathbb{F}[\phi]\mathfrak{t} \wedge \mathbb{AX}[\phi]\mathfrak{f}$. Intuitively, it states that a given KS has at least one strict substructure in the semilattice selected by ϕ (this is required by $\mathbb{F}[\phi]\mathfrak{t}$), but none of them can be an immediate substructure (this is required by $\mathbb{AX}[\phi]\mathfrak{f}$), since no KS satisfies $\mathfrak{f}$. More formally, $\mathcal{K} \models \mathtt{Den}_\phi$ iff *(i)* $\mathfrak{S}_{\mathcal{K}}(\phi) \neq \emptyset$ and *(ii)*, for all $\mathcal{K}' \in \mathfrak{S}_{\mathcal{K}}(\phi)$, there exists a $\mathcal{K}'' \in \mathfrak{S}_{\mathcal{K}}(\phi)$ such that $\mathcal{K}' \sqsubset \mathcal{K}'' \sqsubset \mathcal{K}$.

As an example, Figure 7 shows the unwinding $\mathcal{K}_A^U$ of the Ks $\mathcal{K}_A$ in Figure 2, which does satisfy $\mathtt{Den}$. Indeed, $\mathfrak{S}_{\mathcal{K}_A^U}(\mathfrak{f}) \neq \emptyset$, since $(\mathcal{K}_{AB})_U \in \mathfrak{S}_{\mathcal{K}_A^U}(\mathfrak{f})$, i.e., the infinite path containing only • is one of the substructures of $\mathcal{K}_A^U$. Moreover, for each substructure $\mathcal{T} \in \mathfrak{S}_{\mathcal{K}_A^U}(\mathfrak{f})$ and edge $(w, v) \in R_{\mathcal{K}_A^U} \setminus R_{\mathcal{T}}$ pruned in $\mathcal{T}$, we can always obtain a strict superstructure $\mathcal{T}' \in \mathfrak{S}_{\mathcal{K}_A^U}(\mathfrak{f})$ of $\mathcal{T}$, where some edge $(u, t) \in R_{\mathcal{K}_A^U} \setminus R_{\mathcal{T}'}$ from $u \in R^*_{\mathcal{K}_A^U}(v)$ is pruned in $\mathcal{T}'$ instead of $(w, v) \in R_{\mathcal{T}'}$. Note that it is always possible to find, along any path, a world • with two outgoing edges. By iterating this argument, it is easy to see that any strict substructure $\mathcal{T}$ of $\mathcal{K}_A^U$ has an infinite chain of superstructures in the restricted filtering $\mathfrak{S}_{\mathcal{K}_A^U}(\mathfrak{f})$. Consequently, we have that $|\mathfrak{F}_{\mathcal{K}_A^U}(\mathfrak{f})| = \infty$.

The property we describe by means of the $\mathtt{Den}_\phi$ construct actually corresponds to a weak form of density of an ordered set. Recall that a set S, ordered by a relation $\leq$, is dense in the classical sense iff, for all pairs of elements $x, y \in \mathrm{S}$ with $x < y$, there is a $z \in \mathrm{S}$ such that $x < z < y$. In our framework, this property does not hold for any pair of substructures in $\mathfrak{F}_{\mathcal{K}}(\phi)$, but surely for those ones having the greater component fixed to $\mathcal{K}$. For instance, given two Kss $\mathcal{K}', \mathcal{K}'' \in \mathfrak{F}_{\mathcal{K}}(\phi)$ minimal in this filtering, it holds that their join $\mathcal{K}' \sqcup \mathcal{K}''$ does not have any substructure $\mathcal{K}''' \in \mathfrak{F}_{\mathcal{K}}(\phi)$ such that $\mathcal{K}' \sqsubset \mathcal{K}''' \sqsubset \mathcal{K}' \sqcup \mathcal{K}''$. We can also express the classic concept of density by means of the formula $\overline{\mathbb{G}}[\phi]\mathtt{Den}_\phi$. However, as just shown, that formula is not satisfiable, since the filtering $\mathfrak{F}_{\mathcal{K}}(\phi)$ always contains minimal elements, whose join has immediate substructures. In order to make such a formula satisfiable, one can change the definition of substructure by only allowing either a finite or a non-co-finite number of edge prunings. In this way, the filtering would not be forced to contain minimal elements. However, the resulting logic would have a completely different semantics, with different model-theoretic properties, and we shall not deal with it in this paper.

Before stating a fundamental result about the density construct, we need to introduce some preliminary definitions.

A Kt $\mathcal{B} \in \mathrm{KT(AP)}$ is *binary* if its set of states is a full Δ-tree $\mathrm{W_B} = \Delta^*$, for a given set of directions Δ with cardinality $|\Delta| = 2$.

Let $\mathcal{K}, \mathcal{K}' \in \mathrm{KS(AP)}$ be two Kss. Then, $\mathcal{K}'$ is a *minor* of $\mathcal{K}$, in symbols $\mathcal{K}' \preccurlyeq \mathcal{K}$, if there exists an injective embedding $\mathsf{m} : \mathrm{W}_{\mathcal{K}'} \to \mathrm{W}_{\mathcal{K}}$ such that, for all $w_1, w_2 \in \mathrm{W}_{\mathcal{K}'}$, it holds that $w_2 \in R_{\mathcal{K}'}(w_1)$ iff there is a track $\rho \in \mathrm{Trk}_{\mathcal{K}_{\mathsf{m}(w_1)}}$ for which *(i)* $\mathsf{lst}(\rho) = \mathsf{m}(w_2)$ and *(ii)* $(\rho)_i \neq \mathsf{m}(w_3)$, for all $i \in\,]0, |\rho| - 1[$ and $w_3 \in R_{\mathcal{K}'}(w_1)$. Observe that the second item ensures that different outgoing edges from a state in the minor are mapped onto tracks of the original Ks, neither of which is a prefix of the other. Intuitively, $\mathcal{K}' \preccurlyeq \mathcal{K}$ if $\mathcal{K}'$ is isomorphic to the Ks obtained from a substructure of $\mathcal{K}$ by applying zero or more edge contractions, namely, by removing step by step an edge while simultaneously merging its incident worlds. As an example, consider again the unwinding $\mathcal{K}_A^U$ of Figure 7. $\mathcal{K}_A^U$ has a binary Kt $\mathcal{B}$ with $\Delta \triangleq \{a, b\}$ as a minor, i.e., $\mathcal{B} \preccurlyeq \mathcal{K}_A^U$. This is witnessed by the following embedding m: *(i)* $\mathsf{m}(\varepsilon) = \varepsilon$; *(ii)* for all $w \in \Delta^+$, it holds that: $\mathsf{m}(w \cdot a) \triangleq \mathsf{m}(w) \cdot$ ■◆• and $\mathsf{m}(w \cdot b) \triangleq \mathsf{m}(w) \cdot$ •. Intuitively, $\mathcal{B}$ is isomorphic to the Ks obtained by contracting all pairs of consecutive edges between the states labeled by ■, ◆, and •. On the contrary, the unwinding $\mathcal{K}_B^U$ of the same figure does not contain any binary Kt as minor, since each world labeled by • has a successor which leads only to worlds with a unique successor. Another way to understand this fact is that it is impossible to embed a binary tree into a tree with only a countable number of paths.

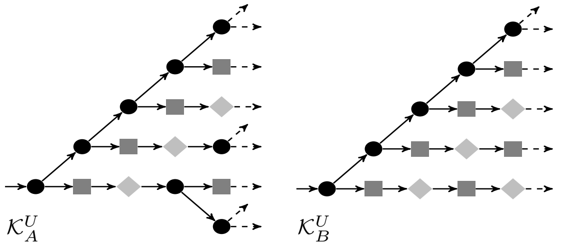

Figure 7: The $\mathcal{K}_A$ and $\mathcal{K}_B$ unwindings $\mathcal{K}_A^U$ and $\mathcal{K}_B^U$ (we only report the labeling of the worlds instead of worlds themselves).

We now have all we need to characterize the class of Kss satisfying the density constraint.

Theorem V.1 (Density Characterization). *For each* Ks $\mathcal{K} \in \mathrm{KS(AP)}$, *it holds that* $\mathcal{K} \models \mathtt{Den}$ *iff* (i) $\mathcal{K}$ *is isomorphic to a* Kt *and* (ii) $\mathcal{K}_w$ *has a binary* Kt *as a minor, for all* $w \in \mathrm{W}_{\mathcal{K}}$.

Intuitively, this theorem states that each world of a Ks satisfying $\mathtt{Den}$ is the root of a tree substructure embedding a binary Kt.

The operator $\mathtt{Den}_\phi$ also allows us to express discreteness of the underlying semilattice with the following formula: $\mathtt{Dis}_\phi \triangleq \overline{\mathbb{G}}[\phi]\neg\mathtt{Den}_\phi$. Intuitively, $\mathtt{Dis}_\phi$ states that no substructure of a given Ks satisfies the density constraint. Formally, we have that $\mathcal{K} \models \mathtt{Dis}_\phi$ iff, for all substructures $\mathcal{K}' \in \mathfrak{F}_{\mathcal{K}}(\phi)$, it holds that either *(i)* $\mathcal{K}'$ does not admit any strict substructure in the filtering, i.e., it is minimal in $\mathfrak{F}_{\mathcal{K}}(\phi)$, or *(ii)* no substructure $\mathcal{K}''' \in \mathfrak{F}_{\mathcal{K}}(\phi)$ satisfies $\mathcal{K}'' \sqsubset \mathcal{K}''' \sqsubset \mathcal{K}'$, i.e., there is an immediate strict substructure $\mathcal{K}'' \in \mathfrak{F}_{\mathcal{K}}(\phi)$ of $\mathcal{K}'$. As an example, Figure 7 shows the unwinding $\mathcal{K}_B^U$ of the Ks $\mathcal{K}_B$ in Figure 2, which does satisfy $\mathtt{Dis}$. Indeed, any substructure $\mathcal{T} \in \mathfrak{S}_{\mathcal{K}_B^U}(\mathfrak{f})$ having at least a node $w \in \mathrm{W}_{\mathcal{T}}$ with $|R_{\mathcal{T}}(w)| = 2$ has an immediate strict substructure $\mathcal{T}' \in \mathfrak{S}_{\mathcal{T}}(\mathfrak{f})$ such that, for all $u \in \mathrm{W}_{\mathcal{T}}$, it holds that $u \notin \mathrm{W}_{\mathcal{T}'}$ iff $u \in R^*_{\mathcal{T}}(v)$, where $v \in R_{\mathcal{T}}(w)$ is labeled by ■. This means that $\mathcal{T}$ and $\mathcal{T}'$ differ exactly on the worlds reachable from w passing through v.

Similarly to the density constraint, we can characterize the discreteness constraint by means of the minor relation.

Theorem V.2 (Discreteness Characterization). *For each* Ks $\mathcal{K} \in \mathrm{KS(AP)}$, *it holds that* $\mathcal{K} \models \mathtt{Dis}$ *iff* $\mathcal{K}$ *does not have a binary* Kt *as a minor.*

Observe that there are Kss $\mathcal{K} \in \mathrm{KS(AP)}$ such that neither

$\mathcal{K} \models \mathtt{Den}$ nor $\mathcal{K} \models \mathtt{Dis}$. An example is given by the KT whose root has $\mathcal{K}_A^U$ and $\mathcal{K}_B^U$ as the only children.

Expressiveness and succinctness: Before proceeding to discuss further model theoretic properties, expressiveness and succinctness of STL*, STL and their weaker fragments, we need to introduce few additional definitions.

A logic $\mathfrak{L}$ enjoys the *tree* (resp., *finite*) *model property* if every satisfiable formula $\varphi \in \mathfrak{L}$ has a KT $\mathcal{T}$ (resp., KS $\mathcal{K}$ with $|\mathrm{W}_{\mathcal{K}}| < \omega$) as model. Moreover, $\mathfrak{L}$ is *invariant under bisimulation* if, for all pairs of bisimilar KSs $\mathcal{K}_1, \mathcal{K}_2 \in \mathrm{KS}(\mathrm{AP})$, it holds that φ is an invariant for $\mathcal{K}_1$ and $\mathcal{K}_2$. Finally, $\mathfrak{L}$ is *invariant under unwinding* if, for every KS $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$, it holds that φ is an invariant for $\mathcal{K}$ and $\mathcal{K}^U$.

Two logics $\mathfrak{L}_1$ and $\mathfrak{L}_2$ can be compared in terms of their expressiveness w.r.t. a given class of KSs $\aleph \subseteq \mathrm{KS}(\mathrm{AP})$. Formally, we say that $\mathfrak{L}_1$ is *at least as expressive as* $\mathfrak{L}_2$ w.r.t. $\aleph$, in symbols $\mathfrak{L}_2 \leq_{\aleph} \mathfrak{L}_1$, if every formula $\varphi_2 \in \mathfrak{L}_2$ is $\aleph$-equivalent to some formula $\varphi_1 \in \mathfrak{L}_1$. If $\mathfrak{L}_2 \leq_{\aleph} \mathfrak{L}_1$, but $\mathfrak{L}_1 \not\leq_{\aleph} \mathfrak{L}_2$ then $\mathfrak{L}_1$ is *more expressive than* $\mathfrak{L}_2$ w.r.t. $\aleph$, in symbols $\mathfrak{L}_2 <_{\aleph} \mathfrak{L}_1$.

We can now give some results about the comparison of STL* and its fragments w.r.t. classic temporal logics. In particular, we start with a theorem about the lack of classic model-theoretic properties for WSTL[KS].

Theorem V.3 (WSTL[KS] Negative Properties). WSTL[KS] *satisfies the following:* (i) *it does not enjoy the tree model property;* (ii) *it is not invariant under unwinding;* (iii) *it is not invariant under bisimulation.*

Intuitively, by using the EMin construct, it is possible to express a property satisfied only by a KS $\mathcal{K}$ containing a loop. Hence, $\mathcal{K}$ cannot be a KT and Item *(i)* follows. Items *(ii)* and *(iii)* are immediate consequences.

CTL is clearly a syntactic fragment of WSTL that is known to have the tree model property. Thus, the following result is an immediate consequence of Item *(i)* of the above theorem.

Corollary V.1 (WSTL[KS] Expressiveness). CTL $<_{\mathrm{KS}}$ WSTL.

A deeper result about the impossibility of a finitary representation of some STL[KS] models directly follows from the density characterization of Theorem V.1.

Theorem V.4 (STL[KS] Negative Property). STL[KS] *does not enjoy the finite model property.*

It is known that *Counting* CTL* (CTL*+C, for short) [18] has the finite model property. We recall that this logic is obtained by adding to CTL* the successor counting operator $\mathsf{E}^{\geq g}\mathsf{X}\varphi$, which is satisfied in a world if this has at least g different successors satisfying the argument φ. Now, since the density construct has only infinite models, we immediately derive that it cannot have any KS-equivalent in CTL*+C.

Theorem V.5 (Density on KSs). *There is no* CTL*+*C formula* KS*-equivalent to* Den.

Differently from the KS case, the density construct is easily expressible in CTL+C interpreted over KTs.

Theorem V.6 (Density on KTs). $\mathtt{Den} \equiv_{\mathrm{KT}} \mathsf{AGEFE}^{\geq 2}\mathsf{Xt}$.

Theorem V.6 follows from the observation that $\mathsf{AGEFE}^{\geq 2}\mathsf{Xt}$ requires that, from every world of a KT, a world with at least two successors is eventually reached. It is an easy exercise to show that any such KT embeds a binary KT.

It can be proved that the STL discreteness construct Dis cannot be expressed in CTL*+C and, consequently, in *monadic path logic* (MPL, for short) [11]. Conversely, one can show that WSTL* is reducible to MPL. However, such results are far beyond the scope of this paper.

We now turn our attention to WSTL interpreted over KTs. Invariance under unwinding and the tree model property hold trivially for KTs. However, by observing that a KT with a single path is bisimilar to a KT with two paths, assuming the worlds in the two KT are equally labeled, but that the former is minimal and the latter is not, we immediately obtain the the following result.

Theorem V.7 (WSTL[KT] Negative Property). WSTL[KT] *is not invariant under bisimulation.*

Since CTL is known to be invariant under bisimulation, the first item of the following result immediately follows from the previous theorem. The second item, instead, follows from the observation that $\mathtt{AMin}(\mathsf{t}, \varphi)$ verifies φ on every path of the underlying KT. Therefore, for every LTL formula ψ, it holds that $\mathsf{A}\psi \equiv_{\mathrm{KT}} \mathtt{AMin}(\mathsf{t}, \varphi)$, where the CTL state formula φ is obtained from ψ by coupling each temporal operator occurring in it with some path quantifier.

Theorem V.8 (WSTL[KT] Expressiveness). WSTL[KT] *satisfies the following:* (i) CTL $<_{\mathrm{KT}}$ WSTL*;* (ii) LTL $<_{\mathrm{KT}}$ WSTL.

Finally, by adapting the classic (linear) reduction proposed in [11], showing that CTL* $\leq_{\mathrm{KT}}$ MPL, we can prove that STL* can only express regular languages over trees, namely the class of languages expressible in MSOL.

Theorem V.9 (STL*[KT] Regularity). STL* $\leq_{\mathrm{KT}}$ MSOL.

VI. Decision Problems

Depending on the class of models over which the logic is interpreted, complexity results on the standard decision problems, namely satisfiability and model checking, differ significantly. For instance, when interpreted over arbitrary Kripke structures, satisfiability is undecidable already for WSTL*. However, the problem for the full STL* remains decidable, in non-elementary time, when interpreted on Kripke trees. The situation is somewhat different for the model checking problem, which is decidable under both interpretations, though simpler, in PSPACE, for finite Kripke structures, while much harder, in non-elementary time, for Kripke trees. The following theorems summarize the results.

Theorem VI.1 (WSTL*[KS] Undecidable Satisfiability). WSTL*[KS] *satisfiability problem is highly undecidable, i.e., it is* Σ_1^1-HARD.

Theorem VI.1 follows from a reduction from the *recurrent domino problem* [12], which is known to be highly undecidable and, in particular, Σ_1^1-COMPLETE, i.e., not even computably enumerable. A recurrent tiling system can be embedded into a model of a particular WSTL* formula, which is satisfiable iff the tiling system allows for an admissible tiling.

Theorem VI.2 (STL*[KS] Decidable Model Checking). STL*[KS] *model-checking problem is decidable in* PSPACE *w.r.t. both the size of the* STL* *formula* φ *and the finite* KS *model* $\mathcal{K}$.

Theorem VI.2 follows by showing a brute-force recursive algorithm that checks in PSPACE whether a finite KS model $\mathcal{K}$ satisfies an STL* formula φ.

Theorem VI.3 (STL*[KT] Decision Problem Complexity). STL*[KT] *satisfiability and model-checking problems have a* $(k+1)$-EXPTIME *formula complexity w.r.t. the alternation* k *of semilattice operators in the* STL* *formula* φ. *The latter problem has a* PTIME *data complexity w.r.t. the size of the finite* KS $\mathcal{K} \in \text{KS}(\text{AP})$ *encoding the* KT *model* $\mathcal{K}^{\mathcal{U}}$.

Theorem VI.3 follows from an *automata-theoretic approach* in which we reduce both decision problems to the emptiness problem of a suitable *alternating parity tree automaton* [14]. Due to the operations of projection required by the extraction of substructure, which induce at any alternation an exponential blow-up, the overall size of the required automaton is non-elementary in the size of the formula, while it is only polynomial in the size of the model, if it is involved in the construction. Thus, together with the complexity of the automata non-emptiness calculation [14], we obtain the required complexity.

Theorem VI.4 (STL*[KT] Decision Problem Hardness). STL*[KT] *satisfiability and model-checking problems are* k-EXPSPACE-HARD *w.r.t. the alternation* k *of semilattice operators in the* STL* *formula* ψ. *The latter problem is* PTIME-HARD *w.r.t. the size of the finite* KS $\mathcal{K} \in \text{KS}(\text{AP})$ *encoding the* KT *model* $\mathcal{K}^{\mathcal{U}}$.

In Theorem VI.4, the formula complexity for both the problems follows by a linear reduction from the QPTL satisfiability problem [23], in which each existential (universal) propositional quantification is translated into the `EMax` (resp., `AMax`) construct. The PTIME hardness follows by a reduction from the reachability problem on And-Or graphs.

VII. CONCLUSION

Reasoning about substructures has proved to be a crucial aspect for a number of problems in formal system verification and design. The solutions of many fundamental problems addressed in the literature share the need of selecting a portion of the model of interest and then verify on that portion a specification requirement. This is the case for decision problems like module checking, turn-based games, concurrent games, reactive synthesis, and many others. The typical approach to these problems has been to define ad-hoc extensions of temporal logics, tailored to the specific problem.

In this paper we have taken a different stance, attempting to define a unifying temporal framework to reason about substructures. To this aim, we have defined a "two-layer semantics", where the standard temporal layer is coupled with an upper layer of partially ordered substructures. We have then introduced and studied Substructure Temporal Logic (STL*, for short), a branching-time temporal-logic obtained by simply adding to CTL* two operators used to select suitable substructures from the upper layer.

The resulting logic turns out to be very powerful and versatile. It strictly subsumes CTL* and can embed in a natural and elegant way several classical decision problems, including those mentioned above. We have also investigated the classical decision problems for STL*, w.r.t. both Kripke structures and infinite regular trees. While satisfiability is undecidable when interpreted over Kripke structures, it is decidable in non-elementary time when interpreted over infinite regular trees. On the other hand, the model checking problem is decidable under both interpretations, in PSPACE and in non-elementary time, respectively.

Future work may proceed along various directions. While, for the sake of space, we had to confine the analysis of STL* properties to its expressiveness with respect to "standard" temporal logics only, a deeper comparison is in order with respect both to very expressive logics like MPL, and to popular related logical frameworks like, for instance, ATL, Strategy Logic and Sabotage Logic. Some of these analysis are currently underway, along with the study of succinctness properties, suggesting, e.g., an exponential gain of STL with respect to CTL. We also plan to study variants of STL*. In particular, it would be of special interest to consider a version of STL* where the ordering between substructures is induced by the minor ordering $\preccurlyeq$, instead of $\sqsubseteq$.

REFERENCES

[1] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *JACM*, 49(5):672–713, 2002.

[2] P.A. Bonatti, C. Lutz, A. Murano, and M.Y. Vardi. The Complexity of Enriched Mu-Calculi. *LMCS*, 4(3):1–27, 2008.

[3] A. Church. Logic, Arithmetics, and Automata. In *ICM'62*, pages 23–35, 1963.

[4] E.M. Clarke and E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71. Springer, 1981.

[5] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *TOPLAS*, 8(2):244–263, 1986.

[6] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 2002.

[7] E.A. Emerson and J.Y. Halpern. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *JCSS*, 30(1):1–24, 1985.

[8] E.A. Emerson and J.Y. Halpern. “Sometimes” and “Not Never” Revisited: On Branching Versus Linear Time. *JACM*, 33(1):151–178, 1986.

[9] E.A. Emerson and C.-L. Lei. Temporal Reasoning Under Generalized Fairness Constraints. In *86*, LNCS 210, pages 267–278. Springer, 1986.

[10] J. Gerbrandy and W. Groeneveld. Reasoning About Information Change. *JLLI*, 6(2):147–169, 1997.

[11] T. Hafer and W. Thomas. Computation Tree Logic CTL* and Path Quantifiers in the Monadic Theory of the Binary Tree. In *ICALP’87*, LNCS 267, pages 269–279. Springer, 1987.

[12] D. Harel. A Simple Highly Undecidable Domino Problem. In *LCC’84*, 1984.

[13] O. Kupferman, U. Sattler, and M.Y. Vardi. The Complexity of the Graded μ-Calculus. In *CADE’02*, LNCS 2392, pages 423–437. Springer, 2002.

[14] O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.

[15] O. Kupferman, M.Y. Vardi, and P. Wolper. Module Checking. *IC*, 164(2):322–344, 2001.

[16] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic. In *CONCUR’12*, LNCS 7454, pages 193–208. Springer, 2012.

[17] F. Mogavero, A. Murano, and M.Y. Vardi. Reasoning About Strategies. In *FSTTCS’10*, LIPIcs 8, pages 133–144, 2010.

[18] F. Moller and A.M. Rabinovich. Counting on CTL*: On the Expressive Power of Monadic Path Logic. *IC*, 184(1):147–159, 2003.

[19] J.A. Plaza. Logics of Public Communications. *Synthese*, 158(2):165–179, 2007.

[20] A. Pnueli. The Temporal Logic of Programs. In *FOCS’77*, pages 46–57, 1977.

[21] A. Pnueli and R. Rosner. On the Synthesis of a Reactive Module. In *POPL’89*, pages 179–190. Association for Computing Machinery, 1989.

[22] M.O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *TAMS*, 141:1–35, 1969.

[23] A.P. Sistla, M.Y. Vardi, and P. Wolper. The Complementation Problem for Büchi Automata with Applications to Temporal Logic. *TCS*, 49:217–237, 1987.

[24] J. van Benthem. An Essay on Sabotage and Obstruction. In *05*, LNCS 2605, pages 268–276. Springer, 2005.

APPENDIX

A. Mathematical Notation

In this short reference appendix, we report the classical mathematical notation and some common definitions that are used along the whole work.

Classic objects: We consider $\mathbb{N}$ as the set of *natural numbers* and $[m, n] \triangleq \{k \in \mathbb{N} : m \leq k \leq n\}$, $[m, n[\triangleq \{k \in \mathbb{N} : m \leq k < n\}$, $]m, n] \triangleq \{k \in \mathbb{N} : m < k \leq n\}$, and $]m, n[\triangleq \{k \in \mathbb{N} : m < k < n\}$ as its *interval* subsets, with $m \in \mathbb{N}$ and $n \in \widehat{\mathbb{N}} \triangleq \mathbb{N} \cup \{\omega\}$, where ω is the *numerable infinity*, i.e., the *least infinite ordinal*. Given a *set* X of *objects*, we denote by $|\mathrm{X}| \in \widehat{\mathbb{N}} \cup \{\infty\}$ the *cardinality* of X, i.e., the number of its elements, where ∞ represents a *more than countable* cardinality, and by $2^{\mathrm{X}} \triangleq \{\mathrm{Y} : \mathrm{Y} \subseteq \mathrm{X}\}$ the *powerset* of X, i.e., the set of all its subsets.

Relations: By $R \subseteq \mathrm{X} \times \mathrm{Y}$ we denote a *relation* between the *domain* $\mathsf{dom}(R) \triangleq \mathrm{X}$ and *codomain* $\mathsf{cod}(R) \triangleq \mathrm{Y}$, whose *range* is indicated by $\mathsf{rng}(R) \triangleq \{y \in \mathrm{Y} : \exists x \in \mathrm{X}. (x, y) \in R\}$. We use $R^{-1} \triangleq \{(y, x) \in \mathrm{Y} \times \mathrm{X} : (x, y) \in R\}$ to represent the *inverse* of R itself. Moreover, by $S \circ R$, with $R \subseteq \mathrm{X} \times \mathrm{Y}$ and $S \subseteq \mathrm{Y} \times \mathrm{Z}$, we denote the *composition* of R with S, i.e., the relation $S \circ R \triangleq \{(x, z) \in \mathrm{X} \times \mathrm{Z} : \exists y \in \mathrm{Y}. (x, y) \in R \wedge (y, z) \in S\}$. We also use $R^n \triangleq R^{n-1} \circ R$, with $n \in [1, \omega[$, to indicate the n*-iteration* of $R \subseteq \mathrm{X} \times \mathrm{Y}$, where $\mathrm{Y} \subseteq \mathrm{X}$ and $R^0 \triangleq \{(y, y) : y \in \mathrm{Y}\}$ is the *identity* on Y. With $R^+ \triangleq \bigcup_{n=1}^{<\omega} R^n$ and $R^* \triangleq R^+ \cup R^0$ we denote, respectively, the *transitive* and *reflexive-transitive closure* of R. Finally, for an *equivalence* relation $R \subseteq \mathrm{X} \times \mathrm{X}$ on X, we represent with $(\mathrm{X}/R) \triangleq \{[x]_R : x \in \mathrm{X}\}$, where $[x]_R \triangleq \{x' \in \mathrm{X} : (x, x') \in R\}$, the *quotient* set of X w.r.t. R, i.e., the set of all related equivalence *classes* $[\cdot]_R$.

Functions: We use the symbol $\mathrm{Y}^{\mathrm{X}} \subseteq 2^{\mathrm{X} \times \mathrm{Y}}$ to denote the set of *total functions* f from X to Y, i.e., the relations $\mathsf{f} \subseteq \mathrm{X} \times \mathrm{Y}$ such that for all $x \in \mathsf{dom}(\mathsf{f})$ there is exactly one element $y \in \mathsf{cod}(\mathsf{f})$ such that $(x, y) \in \mathsf{f}$. Often, we write $\mathsf{f} : \mathrm{X} \to \mathrm{Y}$ and $\mathsf{f} : \mathrm{X} \rightharpoonup \mathrm{Y}$ to indicate, respectively, $\mathsf{f} \in \mathrm{Y}^{\mathrm{X}}$ and $\mathsf{f} \in \bigcup_{\mathrm{X}' \subseteq \mathrm{X}} \mathrm{Y}^{\mathrm{X}'}$. Regarding the latter, note that we consider f as a *partial function* from X to Y, where $\mathsf{dom}(\mathsf{f}) \subseteq \mathrm{X}$ contains all and only the elements for which f is defined. Given a set Z, by $\mathsf{f}_{\upharpoonright \mathrm{Z}} \triangleq \mathsf{f} \cap (\mathrm{Z} \times \mathrm{Y})$ we denote the *restriction* of f to the set $\mathrm{X} \cap \mathrm{Z}$, i.e., the function $\mathsf{f}_{\upharpoonright \mathrm{Z}} : \mathrm{X} \cap \mathrm{Z} \rightharpoonup \mathrm{Y}$ such that, for all $x \in \mathsf{dom}(\mathsf{f}) \cap \mathrm{Z}$, it holds that $\mathsf{f}_{\upharpoonright \mathrm{Z}}(x) = \mathsf{f}(x)$. Moreover, with $\varnothing$ we indicate a generic *empty function*, i.e., a function with empty domain. Note that $\mathrm{X} \cap \mathrm{Z} = \emptyset$ implies $\mathsf{f}_{\upharpoonright \mathrm{Z}} = \varnothing$. Finally, for two partial functions $\mathsf{f}, \mathsf{g} : \mathrm{X} \rightharpoonup \mathrm{Y}$, we use $\mathsf{f} \Cup \mathsf{g}$ and $\mathsf{f} \Cap \mathsf{g}$ to represent, respectively, the *union* and *intersection* of these functions defined as follows: $\mathsf{dom}(\mathsf{f} \Cup \mathsf{g}) \triangleq \mathsf{dom}(\mathsf{f}) \cup \mathsf{dom}(\mathsf{g}) \setminus \{x \in \mathsf{dom}(\mathsf{f}) \cap \mathsf{dom}(\mathsf{g}) : \mathsf{f}(x) \neq \mathsf{g}(x)\}$, $\mathsf{dom}(\mathsf{f} \Cap \mathsf{g}) \triangleq \{x \in \mathsf{dom}(\mathsf{f}) \cap \mathsf{dom}(\mathsf{g}) : \mathsf{f}(x) = \mathsf{g}(x)\}$, $(\mathsf{f} \Cup \mathsf{g})(x) = \mathsf{f}(x)$ for $x \in \mathsf{dom}(\mathsf{f} \Cup \mathsf{g}) \cap \mathsf{dom}(\mathsf{f})$, $(\mathsf{f} \Cup \mathsf{g})(x) = \mathsf{g}(x)$ for $x \in \mathsf{dom}(\mathsf{f} \Cup \mathsf{g}) \cap \mathsf{dom}(\mathsf{g})$, and $(\mathsf{f} \Cap \mathsf{g})(x) = \mathsf{f}(x)$ for $x \in \mathsf{dom}(\mathsf{f} \Cap \mathsf{g})$.

Words: By X^n, with $n \in \mathbb{N}$, we denote the set of all *n-tuples* of elements from X, by $\mathrm{X}^* \triangleq \bigcup_{n=0}^{<\omega} \mathrm{X}^n$ the set of *finite words* on the *alphabet* X, by $\mathrm{X}^+ \triangleq \mathrm{X}^* \setminus \{\varepsilon\}$ the set of *non-empty words*, and by X^ω the set of *infinite words*, where, as usual, $\varepsilon \in \mathrm{X}^*$ is the *empty word*. The *length* of a word $w \in \mathrm{X}^\infty \triangleq \mathrm{X}^* \cup \mathrm{X}^\omega$ is represented with $|w| \in \widehat{\mathbb{N}}$. By $(w)_i$ we indicate the *i-th letter* of the finite word $w \in \mathrm{X}^*$, with $i \in [0, |w|[$. Furthermore, by $\mathsf{fst}(w) \triangleq (w)_0$ (resp., $\mathsf{lst}(w) \triangleq (w)_{|w|-1}$), we denote the *first* (resp., *last*) letter of w. In addition, by $(w)_{\leq i}$ (resp., $(w)_{>i}$), we indicate the *prefix* up to (resp., *suffix* after) the letter of index i of w, i.e., the finite word built by the first $i+1$ (resp., last $|w|-i-1$) letters $(w)_0, \ldots, (w)_i$ (resp., $(w)_{i+1}, \ldots, (w)_{|w|-1}$). We also set, $(w)_{<0} \triangleq \varepsilon$, $(w)_{<i} \triangleq (w)_{\leq i-1}$, $(w)_{\geq 0} \triangleq w$, and $(w)_{\geq i} \triangleq (w)_{>i-1}$, for $i \in [1, |w|[$. Mutatis mutandis, the notations of i-th letter, first, prefix, and suffix apply to infinite words too. Finally, by $\mathsf{pfx}(w_1, w_2) \in \mathrm{X}^\infty$ we denote the *maximal common prefix* of two different words $w_1, w_2 \in \mathrm{X}^\infty$, i.e., the finite word $w \in \mathrm{X}^*$ for which there are two words $w'_1, w'_2 \in \mathrm{X}^\infty$ such that $w_1 = w \cdot w'_1$, $w_2 = w \cdot w'_2$, and $\mathsf{fst}(w'_1) \neq \mathsf{fst}(w'_2)$. By convention, we set $\mathsf{pfx}(w, w) \triangleq w$.

Trees: For a set Δ of objects, called *directions*, a *Δ-tree* is a set $\mathrm{T} \subseteq \Delta^*$ closed under prefix, i.e., if $t \cdot d \in \mathrm{T}$, with $d \in \Delta$, then also $t \in \mathrm{T}$. We say that it is *complete* if it holds that $t \cdot d' \in \mathrm{T}$ whenever $t \cdot d \in \mathrm{T}$, for all $d' < d$, where $< \subseteq \Delta \times \Delta$ is an a priori fixed strict total order on the set of directions that is clear from the context. Moreover, it is *full* if $\mathrm{T} = \Delta^*$. The elements of T are called *nodes* and the empty word ε is the *root* of T. For every $t \in \mathrm{T}$ and $d \in \Delta$, the node $t \cdot d \in \mathrm{T}$ is a *successor* of t in T. The tree is *b-bounded* if the maximal number b of its successor nodes is finite, i.e., $b = \max_{t \in \mathrm{T}} |\{t \cdot d \in \mathrm{T} : d \in \Delta\}| < \omega$. A *branch* of the tree is an infinite word $w \in \Delta^\omega$ such that $(w)_{\leq i} \in \mathrm{T}$, for all $i \in \mathbb{N}$. For a finite set Σ of objects, called *symbols*, a *Σ-labeled Δ-tree* is a quadruple $\langle \Sigma, \Delta, \mathrm{T}, \mathsf{v} \rangle$, where T is a Δ-tree and $\mathsf{v} : \mathrm{T} \to \Sigma$ is a *labeling function*. When Δ and Σ are clear from the context, we call $\langle \mathrm{T}, \mathsf{v} \rangle$ simply a (labeled) tree.

B. Proofs of Section V

This appendix is for reviewing purposes only. In case of acceptance, it will be published in an accompanying technical report.

Density and discreteness:

Lemma A.1 (Binary Minor Characterization)**.** *For each* KS $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$*, it holds that, for all $w \in \mathrm{W}_{\mathcal{K}}$, there are $v \in R^*_{\mathcal{K}}(w)$ and $u_1, u_2 \in R_{\mathcal{K}}(v)$ such that $u_1 \neq u_2$ iff $\mathcal{K}_w$ has a binary* KT *as a minor, for all $w \in \mathrm{W}_{\mathcal{K}}$.*

Proof: [Only if] Suppose that, for every $w \in \mathrm{W}_{\mathcal{K}}$, there are $v \in R^*_{\mathcal{K}}(w)$ and $u_1, u_2 \in R_{\mathcal{K}}(v)$ such that $u_1 \neq u_2$. It is immediate to see that there exist two functions $\mathsf{F} : \mathrm{W}_{\mathcal{K}} \to \mathrm{W}_{\mathcal{K}}$ and $\mathsf{M} : \mathrm{W}_{\mathcal{K}} \times \Delta \to \mathrm{W}_{\mathcal{K}}$, with $\Delta \triangleq \{a, b\}$, such that *(i)* $\mathsf{F}(w) \in R^*_{\mathcal{K}}(w)$, *(ii)* $\mathsf{M}(w, a), \mathsf{M}(w, b) \in R_{\mathcal{K}}(\mathsf{F}(w))$, and *(iii)* $\mathsf{M}(w, a) \neq \mathsf{M}(w, b)$. Now, let $\mathcal{B}$ be a binary KT with direction set Δ. We want to show that, for all $w \in \mathrm{W}_{\mathcal{K}}$, it holds that $\mathcal{B} \preccurlyeq \mathcal{K}_w$. To do this, consider the following injective embedding $\mathsf{m}_w : \mathrm{W}_{\mathcal{B}} \to \mathrm{W}_{\mathcal{K}_w}$: *(i)* $\mathsf{m}_w(\varepsilon) \triangleq w$; *(ii)* $\mathsf{m}_w(t \cdot a) \triangleq \mathsf{M}(\mathsf{m}_w(t), a)$; *(iii)* $\mathsf{m}_w(t \cdot b) \triangleq \mathsf{M}(\mathsf{m}_w(t), b)$. It is easy to see that m_w satisfies the defining constraints of the minor relation. Consequently, $\mathcal{B} \preccurlyeq \mathcal{K}_w$.

[If] Suppose that, for every $w \in \mathrm{W}_{\mathcal{K}}$, it holds that $\mathcal{B} \preccurlyeq \mathcal{K}_w$, for some binary KT $\mathcal{B}$ having w.l.o.g. the set of directions $\Delta = \{a, b\}$. Thus, there exists an injective embedding $\mathsf{m}_w : \mathrm{W}_{\mathcal{B}} \to \mathrm{W}_{\mathcal{K}_w}$ satisfying the defining constraints of the minor relation. Consequently, there are two tracks $\rho_a, \rho_b \in \mathrm{Trk}_{\mathcal{T}_w}$ with $\mathsf{lst}(\rho_a) = \mathsf{m}(a)$ and $\mathsf{lst}(\rho_b) = \mathsf{m}(b)$ such that $(\rho_a)_{i_a} \neq \mathsf{m}(b)$ and $(\rho_b)_{i_b} \neq \mathsf{m}(a)$, for all $i_a \in [0, |\rho_a| - 1[$ and $i_b \in [0, |\rho_b| - 1[$. Now, let $j \in [0, \min\{|\rho_a|, |\rho_b|\} - 1[$ be the first index in which the two tracks diverge, i.e., $(\rho_a)_{\leq j} = (\rho_b)_{\leq j}$ and $(\rho_a)_{j+1} \neq (\rho_b)_{j+1}$. The existence of such an index is ensured by the previous properties on ρ_a and ρ_b. Then, it is immediate to see that $u_1 = (\rho_a)_{j+1}$, $u_2 = (\rho_b)_{j+1}$, and $v \triangleq (\rho_a)_j$ satisfy the thesis. ■

Theorem V.1 (Density Characterization)**.** *For each* KS $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$*, it holds that $\mathcal{K} \models \mathtt{Den}$ iff* (i) *$\mathcal{K}$ is isomorphic to a* KT *and* (ii) *$\mathcal{K}_w$ has a binary* KT *as a minor, for all worlds $w \in \mathrm{W}_{\mathcal{K}}$.*

Proof: [If] Suppose that $\mathcal{K}$ is isomorphic to a KT and, for all worlds $w \in \mathrm{W}_{\mathcal{K}}$, that $\mathcal{K}_w$ has a binary KT $\mathcal{B}$ as a minor, where w.l.o.g. its set of directions is $\Delta = \{a, b\}$. Therefore, there is an injective embedding $\mathsf{m}_w : \mathrm{W}_{\mathcal{B}} \to \mathrm{W}_{\mathcal{K}_w}$ satisfying the defining constraints of the minor relation, with $\mathsf{m}(\varepsilon) = w$. As first thing, it is immediate to see that $\mathfrak{S}_{\mathcal{K}}(\mathfrak{f}) \neq \emptyset$. Now, let $\mathcal{K}' \in \mathfrak{S}_{\mathcal{K}}(\mathfrak{f})$. Since $\mathcal{K}$ is isomorphic to a KT, there exists a $v \in \mathrm{W}_{\mathcal{K}} \setminus \mathrm{W}_{\mathcal{K}'}$ such that $R^*_{\mathcal{K}}(v) \cap \mathrm{W}_{\mathcal{K}'} = \emptyset$. Moreover, by definition of minor, we have that $R^*_{\mathcal{K}}(\mathsf{m}_v(a)), R^*_{\mathcal{K}}(\mathsf{m}_v(b)) \subset R^*_{\mathcal{K}}(\mathsf{m}_v(\varepsilon)) = R^*_{\mathcal{K}}(v)$ and $R^*_{\mathcal{K}}(\mathsf{m}_v(a)) \cap R^*_{\mathcal{K}}(\mathsf{m}_v(b)) = \emptyset$. At this point, let $\mathrm{W}' \triangleq \mathrm{W}_{\mathcal{K}} \setminus R^*_{\mathcal{K}}(\mathsf{m}_v(b))$. It is easy to see that $\mathrm{W}_{\mathcal{K}'} \subset \mathrm{W}' \subset \mathrm{W}_{\mathcal{K}}$. Furthermore, $R' \triangleq R_{\mathcal{K}} \cap (\mathrm{W}' \times \mathrm{W}')$ is a left-total relation such that $R'^*(w_{0\mathcal{K}}) = \mathrm{W}'$. Consequently, there exists a strict substructure $\mathcal{K}'' \in \mathfrak{S}_{\mathcal{K}}(\mathfrak{f})$ such that $\mathrm{W}_{\mathcal{K}''} = \mathrm{W}'$. Hence, $\mathcal{K}' \sqsubset \mathcal{K}'' \sqsubset \mathcal{K}$. So, by definition of the density constraint $\mathtt{Den}$, we have that $\mathcal{K} \models \mathtt{Den}$.

[Only if] Suppose that $\mathcal{K} \models \mathtt{Den}$. As first thing, $\mathcal{K}$ needs to be isomorphic to a KT. Indeed, assume the converse by contradiction. Thus, there exists a world $w \in \mathrm{W}_{\mathcal{K}}$ and two edges $(v_1, w), (v_2, w) \in R_{\mathcal{K}}$ such that $v_1 \neq v_2$. Now, let $\rho \in \mathrm{Trk}_{\mathcal{K}}$ be a track such that $\mathsf{lst}(\rho) = v_1$. Then, there is an index $j \in [0, |\rho|[$ such that $|R_{\mathcal{K}}((\rho)_j)| = 2$ and $|R_{\mathcal{K}}((\rho)_i)| = 1$, for all $i \in]j, |\rho|[$. At this point, let $\mathrm{W}' \triangleq \mathrm{W}_{\mathcal{K}} \setminus \{(\rho)_i : i \in]j, |\rho|[\}$. It is easy to see that $R' \triangleq (R_{\mathcal{K}} \cap (\mathrm{W}' \times \mathrm{W}')) \setminus \{(v_1, w)\}$ is a left-total relation such that $R'^*(w_{0\mathcal{K}}) = \mathrm{W}'$. Hence, $\mathcal{K}' = \langle \mathrm{AP}, \mathrm{W}', R', \mathsf{L}_{\restriction \mathrm{W}'}, w_{0\mathcal{K}} \rangle$ is a KS which is also a strict substructure of $\mathcal{K}$ in $\mathfrak{S}_{\mathcal{K}}(\mathfrak{f})$. However, $\mathcal{K}'$ does not have any strict superstructure $\mathcal{K}'' \in \mathfrak{S}_{\mathcal{K}}(\mathfrak{f})$. Indeed, if $\mathrm{W}_{\mathcal{K}''} \setminus \mathrm{W}_{\mathcal{K}'} \neq \emptyset$,

there exists some $u \in \{(\rho)_i : i \in]j, |\rho|[\,\}$ such that $u \in \mathrm{W}_{\mathcal{K}''} \setminus \mathrm{W}_{\mathcal{K}'}$. Now, due to the constraints on the transition relation of a KS, we necessarily have $\mathrm{W}_{\mathcal{K}''} \setminus \mathrm{W}_{\mathcal{K}'} = \{(\rho)_i : i \in]j, |\rho|[\}$. Otherwise, we have that $R_{\mathcal{K}''} \setminus R_{\mathcal{K}'} = \{(v_1, w)\}$. Hence, $\mathcal{K}'' = \mathcal{K}$, contradicting that $\mathcal{K}'' \in \mathfrak{S}_{\mathcal{K}}(\mathfrak{f})$.

Now, it remains to show that, for all $w \in \mathrm{W}_{\mathcal{K}}$, it holds that $\mathcal{B} \preccurlyeq \mathcal{K}_w$, for a binary KT $\mathcal{B}$ having as set of directions $\Delta = \{a, b\}$. This fact follows directly from Lemma A.1 after proving that, for all $w \in \mathrm{W}_{\mathcal{K}}$, there are $v \in R^*_{\mathcal{K}}(w)$ and $u_1, u_2 \in R_{\mathcal{K}}(v)$ such that $u_1 \neq u_2$. Suppose by contradiction that there is a world $w \in \mathrm{W}_{\mathcal{K}}$ such that $|R_{\mathcal{K}}(v)| = 1$, for every $v \in R^*_{\mathcal{K}}(w)$, and let $\rho \in \mathrm{Trk}_{\mathcal{K}}$ be the track such that $\mathsf{lst}(\rho) = w$. Since $\mathcal{K} \models \mathtt{Den}$, it holds that $\mathfrak{S}_{\mathcal{K}}(\mathfrak{f}) \neq \emptyset$. Therefore, there exists an index $i \in [0, |\rho| - 1[$ such that $|R_{\mathcal{K}}((\rho)_i)| > 1$ and $|R_{\mathcal{K}}((\rho)_j)| = 1$, for all $j \in]i, |\rho|[$. Now, let $\mathrm{W}' \triangleq \mathrm{W}_{\mathcal{K}} \setminus R^*_{\mathcal{K}}((\rho)_{i+1})$. Since $\mathcal{K}$ is isomorphic to a KT, it is easy to see that $R' \triangleq R_{\mathcal{K}} \cap (\mathrm{W}' \times \mathrm{W}')$ is a left-total relation such that $R'^*(w_{0\mathcal{K}}) = \mathrm{W}'$. Hence, $\mathcal{K}' = \langle \mathrm{AP}, \mathrm{W}', R', \mathsf{L}_{\upharpoonright \mathrm{W}'}, w_{0\mathcal{K}} \rangle$ is a KT which is also a strict substructure of $\mathcal{K}$ in $\mathfrak{S}_{\mathcal{K}}(\mathfrak{f})$. However, $\mathcal{K}'$ does not have any strict superstructure $\mathcal{K}'' \in \mathfrak{S}_{\mathcal{K}}(\mathfrak{f})$. Indeed, if $\mathrm{W}_{\mathcal{K}''} \setminus \mathrm{W}_{\mathcal{K}'} \neq \emptyset$, there exists some $u \in R^*_{\mathcal{K}}((\rho)_{i+1})$ such that $u \in \mathrm{W}_{\mathcal{K}''} \setminus \mathrm{W}_{\mathcal{K}'}$. Now, due to the constraints on the transition relation of a KS, we necessarily have $\mathrm{W}_{\mathcal{K}''} \setminus \mathrm{W}_{\mathcal{K}'} = R^*_{\mathcal{K}}((\rho)_{i+1})$. Hence, $\mathcal{K}'' = \mathcal{K}$, contradicting that $\mathcal{K}'' \in \mathfrak{S}_{\mathcal{K}}(\mathfrak{f})$. ■

Theorem V.2 (Discreteness Characterization). *For each* KS $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$*, it holds that* $\mathcal{K} \models \mathtt{Dis}$ *iff* $\mathcal{K}$ *does not have a binary* KT *as a minor.*

Proof: [If] Suppose that $\mathcal{K} \not\models \mathtt{Dis}$. Then, there exists a substructure $\mathcal{K}' \in \mathfrak{F}_{\mathcal{K}}(\emptyset)$ such that $\mathcal{K}' \models \mathtt{Den}$. Therefore, by Theorem V.1, we have that $\mathcal{K}'$ has a binary KT $\mathcal{B}$ as a minor. Now, it is immediate to see that, given three KSs $\mathcal{K}_1$, $\mathcal{K}_2$, and $\mathcal{K}_3$, with $\mathcal{K}_1 \preccurlyeq \mathcal{K}_2$ and $\mathcal{K}_2 \sqsubseteq \mathcal{K}_3$, it holds that $\mathcal{K}_1 \preccurlyeq \mathcal{K}_3$. Consequently, we have that $\mathcal{B} \preccurlyeq \mathcal{K}$.

[Only if] Suppose that $\mathcal{B} \preccurlyeq \mathcal{K}$, for some binary KT $\mathcal{B}$ having w.l.o.g. the set of directions $\Delta = \{a, b\}$. Then, there exists an injective embedding $\mathsf{m} : \mathrm{W}_{\mathcal{B}} \to \mathrm{W}_{\mathcal{K}}$ satisfying the defining constraints of the minor relation, with $\mathsf{m}(\varepsilon) = w_{0\mathcal{K}}$. In particular, for all $t \in \mathrm{W}_{\mathcal{B}}$, there are two tracks $\rho^t_a, \rho^t_b \in \mathrm{Trk}_{\mathcal{K}_{\mathsf{m}(t)}}$ such that $\mathsf{lst}(\rho^t_a) = \mathsf{m}(t \cdot a)$, $\mathsf{lst}(\rho^t_b) = \mathsf{m}(t \cdot b)$. Now, let $j_t \in [0, \min\{|\rho^t_a|, |\rho^t_b|\} - 1[$ be the last index in which the two tracks diverge, i.e., $(\rho^t_a)_{j_t} = (\rho^t_b)_{j_t}$ and $(\rho^t_a)_i \neq (\rho^t_b)_i$, for all $i \in]j_t, \min\{|\rho^t_a|, |\rho^t_b|\} - 1[$. In addition, set $\mathrm{W}' \triangleq \bigcup_{t \in \mathrm{W}_{\mathcal{B}}} \{(\rho^t_a)_i : i \in [0, |\rho^t_a|[\,\} \cup \{(\rho^t_b)_i : i \in]j_t, |\rho^t_b|[\}$. It is not hard to see that $R' \triangleq \bigcup_{t \in \mathrm{W}_{\mathcal{B}}} \{((\rho^t_a)_i, (\rho^t_a)_{i+1}) : i \in [0, |\rho^t_a| - 1[\,\} \cup \{((\rho^t_b)_i, (\rho^t_b)_{i+1}) : i \in]j_t, |\rho^t_b| - 1[\,\}$ is a left-total relation such that $R'^*(w_{0\mathcal{K}}) = \mathrm{W}'$. Moreover, if $(v_1, w), (v_2, w) \in R'$ then $v_1 = v_2$. Therefore, there exists a substructure $\mathcal{K}' \sqsubseteq \mathcal{K}$, with $\mathrm{W}_{\mathcal{K}'} = \mathrm{W}'$, that is isomorphic to a KT $\mathcal{T}$ having $\mathcal{B}$ as a minor. At this point, by Theorem V.1, in order to prove that $\mathcal{K}' \models \mathtt{Den}$ and, consequently, that $\mathcal{K} \not\models \mathtt{Dis}$, we have only to show that $\mathcal{K}'_w$ has $\mathcal{B}$ as a minor, for all $w \in \mathrm{W}_{\mathcal{K}'}$. By the construction of $\mathcal{K}'$, for each of its worlds $w \in \mathrm{W}_{\mathcal{K}'}$, it surely exists a $t_w \in \mathrm{W}_{\mathcal{B}}$ such that $\mathsf{m}(t_w) \in R^*_{\mathcal{K}'}(w)$. So, let $\mathsf{m}_w : \mathrm{W}_{\mathcal{B}} \to \mathrm{W}_{\mathcal{K}'_w}$ be the injective embedding defined as follows: $\mathsf{m}_w(t) \triangleq \mathsf{m}(t_w \cdot t)$. Now, it is easy to see that m_w satisfies the defining constraints of the minor relation. Therefore, $\mathcal{B} \preccurlyeq \mathcal{K}'_w$. ■

Expressiveness and succinctness:

Theorem V.3 (WSTL[KS] Negative Properties)**.** WSTL[KS] *satisfies the following:* (i) *it does not enjoy the tree model property;* (ii) *it is not invariant under unwinding;* (iii) *it is not invariant under bisimulation.*

Proof: [Item i] Consider the WSTL formula $\varphi = \mathtt{EMin}(\varphi_1, \varphi_2)$, with $\varphi_1 \triangleq \mathsf{EX}(\bullet \wedge \varphi_2)$ and $\varphi_2 \triangleq \mathsf{EX}\blacksquare$. It is easy to see that φ is satisfied by the KS $\mathcal{K}$ of Figure 2, since the KS $\mathcal{K}_A$ of the same figure, which is minimal w.r.t. φ_1, also satisfies φ_2. Now, suppose by contradiction that there exists a KT $\mathcal{T}$ such that $\mathcal{T} \models \varphi$. Then, there exists a substructure $\mathcal{T}' \sqsubseteq \mathcal{T}$ minimal w.r.t. φ_1 such that $\mathcal{T}' \models \varphi_2$. However, such a substructure necessarily has a unique edge outgoing from the root, which also leads to a state labeled by $\bullet$. Consequently, $\mathcal{T}' \not\models \varphi_2$, which is impossible. Thus, we have that WSTL[KS] does not enjoy the tree model property.

[Items ii & iii] Consider again the formula φ and the KS $\mathcal{K}$ of the previous item. We know that $\mathcal{K} \models \varphi$ but $\mathcal{K}^U \not\models \varphi$. Consequently, φ is not an invariant for $\mathcal{K}$ and $\mathcal{K}^U$, which implies that WSTL[KS] is not invariant under unwinding. Moreover, $\mathcal{K}$ and $\mathcal{K}^U$ are also bisimilar. Therefore, φ is not an invariant for two bisimilar structures, which implies that WSTL[KS] is not invariant under bisimulation. ■

Theorem V.4 (STL[KS] Negative Property)**.** STL[KS] *does not enjoy the finite model property.*

Proof: Consider the density construct Den. By Theorem V.1, we know that it is satisfied on a binary KT. Moreover, by the same theorem, we derive that every KS $\mathcal{K}$ satisfying Den needs to contain a binary KT as a minor. Consequently, $\mathcal{K}$ necessarily has an infinite number of worlds, which implies that Den has only infinite models. ■

Theorem V.6 (Density on KTs)**.** $\mathtt{Den} \equiv_{\mathrm{KT}} \mathsf{AGEFE}^{\geq 2}\mathsf{Xt}$.

Proof: [$\mathtt{Den} \Rightarrow_{\mathrm{KT}} \mathsf{AGEFE}^{\geq 2}\mathsf{Xt}$*]* Consider a KT $\mathcal{T}$ such that $\mathcal{T} \models \mathtt{Den}$. Then, by Theorem V.1, for every $w \in \mathrm{W}_{\mathcal{T}}$, it holds that $\mathcal{T}_w$ has a binary KT as a minor. Now, by Lemma A.1, we have that, for all $w \in \mathrm{W}_{\mathcal{T}}$, there are $v \in R^*_{\mathcal{T}}(w)$ and $u_1, u_2 \in R_{\mathcal{T}}(v)$ with $u_1 \neq u_2$. Then, it is immediate to see that $\mathcal{T}_v \models \mathsf{E}^{\geq 2}\mathsf{Xt}$, which implies $\mathcal{T}_w \models \mathsf{EFE}^{\geq 2}\mathsf{Xt}$. Hence, $\mathcal{T} \models \mathsf{AGEFE}^{\geq 2}\mathsf{Xt}$.

[$\mathsf{AGEFE}^{\geq 2}\mathsf{Xt} \Rightarrow_{\mathrm{KT}} \mathtt{Den}$*]* Consider a KT $\mathcal{T}$ such that $\mathcal{T} \models \mathsf{AGEFE}^{\geq 2}\mathsf{Xt}$. It is easy to see that, for all $w \in \mathrm{W}_{\mathcal{T}}$, there is $v \in R^*_{\mathcal{T}}(w)$ such that $\mathcal{T}_v \models \mathsf{E}^{\geq 2}\mathsf{Xt}$. Consequently, there are $u_1, u_2 \in R_{\mathcal{T}}(v)$ with $u_1 \neq u_2$ and so, by Lemma A.1, $\mathcal{K}_w$ has a binary KT as a minor, for all $w \in \mathrm{W}_{\mathcal{K}}$. Hence, the conclusion immediately follows from Theorem V.1. ■

Theorem V.8 (WSTL[KT] Expressiveness)**.** WSTL[KT] *satisfies the following:* (i) CTL $<_{\mathrm{KT}}$ WSTL*;* (ii) LTL $<_{\mathrm{KT}}$ WSTL.

Proof: [Item ii] First, recall that CTL $\not\leq_{\mathrm{KS}}$ LTL. Clearly, since CTL $\leq_{\mathrm{KS}}$ WSTL, we immediately obtain that WSTL $\not\leq_{\mathrm{KS}}$ LTL and, so, WSTL $\not\leq_{\mathrm{KT}}$ LTL. Thus, it is only left to prove LTL $\leq_{\mathrm{KT}}$ WSTL. To do this, we show that $\mathsf{A}\psi \equiv_{\mathrm{KT}} \mathtt{AMin}(\mathsf{t}, \varphi)$, for every LTL formula ψ, where the CTL state formula φ is obtained from ψ by coupling each temporal operator occurring in it with some path quantifier. Now, it is easy to see that, fixed a KT $\mathcal{T}$, for every path $\pi \in \mathrm{Pth}_{\mathcal{T}}$, there is a minimal KT $\mathcal{T}'_\pi \in \mathfrak{F}_{\mathcal{T}}(\emptyset)$ such that $\mathrm{Pth}_{\mathcal{T}'_\pi} = \{\pi\}$, and vice-versa. This is due to the fact that a minimal structure in $\mathfrak{F}_{\mathcal{T}}(\emptyset)$ can only be an infinite chain of worlds. Therefore, $\pi \models \psi$ iff $\mathcal{T}'_\pi \models \varphi$. By definition, $\mathcal{T} \models \mathtt{AMin}(\mathsf{t}, \varphi)$ iff $\mathcal{T}' \models \varphi$, for all minimal KTs $\mathcal{T}' \in \mathfrak{F}_{\mathcal{T}}(\emptyset)$. Consequently, $\mathcal{T} \models \mathsf{A}\psi$ iff $\mathcal{T} \models \mathtt{AMin}(\mathsf{t}, \varphi)$.

■

Theorem V.9 (STL*[KT] Regularity)**.** STL* $\leq_{\mathrm{KT}}$ MSOL.

Proof: To prove the statement, we provide a linear translation from STL* to MSOL, by means of two functions Γ_s : STL* × SVar × FVar → MSOL and Γ_p : LTL(STL*) × SVar × SVar × FVar → MSOL define below, where SVar and FVar are the sets of second- and first-order variable. Such a translation extends the classic reduction CTL* $\leq_{\mathrm{KT}}$ MPL proposed in [11]. In particular, by an easy but long induction on the structure of the formula, it is possible to prove that, for every STL* formula φ, it holds that $\mathcal{T} \models_{\mathrm{STL}^*} \varphi$ iff $\mathcal{T} \models_{\mathrm{MSOL}} \exists T.\exists x.\mathtt{Mod}(T, x) \wedge \Gamma_s(\varphi, T, x)$, where $\mathtt{Mod}(T, x) \triangleq \forall y.y \in T \wedge x \leq y$. Before defining Γ_s and Γ_p, we have to introduce the following simple constructs.

- $x \leq y \triangleq (x < y) \vee (x = y)$.
- $x \lessdot y \triangleq (x < y) \wedge (\neg \exists z \,.\, x < z \wedge z < y)$.
- $\mathtt{Path}(T, P, x)$ is the conjunction of the following formulas:
  - $x \in P$;
  - $\forall y \in P \,.\, x \leq y \wedge y \in T$;
  - $\forall y \in P \,.\, \exists z \in P \,.\, y \lessdot z$;
  - $\forall y \in P \,.\, \forall z \in P \,.\, y \leq z \vee z \leq y$.
- $\mathtt{SubTree}_\phi(T, T', x)$ is the conjunction of the following formulas:
  - $x \in T'$;
  - $\forall y \in T' \,.\, x \leq y \wedge y \in T$;
  - $\forall y \in T' \,.\, \exists z \in T' \,.\, y \lessdot z$;
  - $\exists y \in T \,.\, x < y \wedge \neg y \in T'$;
  - $\forall y \in T' \,.\, \phi(y) \rightarrow \forall z \in T \,.\, y \lessdot z \rightarrow z \in T'$.

We are now able to define the state formula translation Γ_s.

1) $\Gamma_s(p, T, x) \triangleq p(x)$.
2) a) $\Gamma_s(\neg\varphi, T, x) \triangleq \neg\Gamma_s(\varphi, T, x)$;
   b) $\Gamma_s(\varphi_1 \wedge \varphi_2, T, x) \triangleq \Gamma_s(\varphi_1, T, x) \wedge \Gamma_s(\varphi_2, T, x)$;
   c) $\Gamma_s(\varphi_1 \vee \varphi_2, T, x) \triangleq \Gamma_s(\varphi_1, T, x) \vee \Gamma_s(\varphi_2, T, x)$.
3) a) $\Gamma_s(\varphi_1 \mathbb{U}[\phi] \varphi_2, T, x) \triangleq \exists T'.\mathtt{SubTree}_{\phi'(y)}(T, T', x) \wedge \Gamma_s(\varphi_2, T', x) \wedge \forall T''.\mathtt{SubTree}_{\phi'(y)}(T, T'', x) \wedge \mathtt{SubTree}_{\phi'(y)}(T'', T', x) \rightarrow \Gamma_s(\varphi_1, T'', x)$;
   b) $\Gamma_s(\varphi_1 \mathbb{R}[\phi] \varphi_2, T, x) \triangleq \forall T'.\mathtt{SubTree}_{\phi'(y)}(T, T', x) \rightarrow \Gamma_s(\varphi_2, T', x) \vee \exists T''.\mathtt{SubTree}_{\phi'(y)}(T, T'', x) \wedge \mathtt{SubTree}_{\phi'(y)}(T'', T', x) \wedge \Gamma_s(\varphi_1, T'', x)$;
   where $\phi'(y) \triangleq \Gamma_s(\phi, T, y)$.
4) a) $\Gamma_s(\mathsf{E}\psi, T, x) \triangleq \exists P \,.\, \mathtt{Path}(T, P, x) \wedge \Gamma_p(\psi, T, P, x)$;
   b) $\Gamma_s(\mathsf{A}\psi, T, x) \triangleq \forall P \,.\, \mathtt{Path}(T, P, x) \rightarrow \Gamma_p(\psi, T, P, x)$.

Finally, we define the path formula translation Γ_p.

5) $\Gamma_p(\varphi, T, P, x) \triangleq \Gamma_s(\varphi, T, x)$.
6) a) $\Gamma_p(\neg\psi, T, P, x) \triangleq \neg\Gamma_p(\psi, T, P, x)$;
   b) $\Gamma_p(\psi_1 \wedge \psi_2, T, P, x) \triangleq \Gamma_p(\psi_1, T, P, x) \wedge \Gamma_p(\psi_2, T, P, x)$;
   c) $\Gamma_p(\psi_1 \vee \psi_2, T, P, x) \triangleq \Gamma_p(\psi_1, T, P, x) \vee \Gamma_p(\psi_2, T, P, x)$.
7) a) $\Gamma_p(\mathsf{X}\psi, T, P, x) \triangleq \exists y \,.\, y \in P \wedge x \lessdot y \wedge \Gamma_p(\psi, T, P, y)$;
   b) $\Gamma_p(\psi_1 \mathsf{U} \psi_2, T, P, x) \triangleq \exists y \,.\, y \in P \wedge x \leq y \wedge \Gamma_p(\psi_2, T, P, y) \wedge \forall z \,.\, x \leq z \wedge z < y \rightarrow \Gamma_p(\psi_1, T, P, z)$;
   c) $\Gamma_p(\psi_1 \mathsf{R} \psi_2, T, P, x) \triangleq \forall y \,.\, y \in P \wedge x \leq y \rightarrow \Gamma_p(\psi_2, T, P, y) \vee \exists z \,.\, x \leq z \wedge z < y \wedge \Gamma_p(\psi_1, T, P, z)$.

■

C. Proofs of Section VI

***Results on* KS*s*:**

Theorem VI.1 (WSTL*[KS] Undecidable Satisfiability)**.** WSTL*[KS] *satisfiability problem is highly undecidable, i.e., it is* Σ^1_1-HARD.

Proof: To prove the undecidability of the satisfiability problem, we provide a reduction from the recurrent domino problem, which has been shown to be highly undecidable and, in particular, Σ^1_1-COMPLETE, i.e., not even computably enumerable [12]. We achieve the task by describing how a given recurrent tiling system can be embedded into a model of a particular WSTL* formula, which is satisfiable iff the tiling system allows for an admissible tiling. The difficult part of the proof is the construction of a satisfiable WSTL* formula φ_{grd} having only models $\mathcal{K}_{grd}$ in which it is possible to embed the infinite grid $\mathbb{N} \times \mathbb{N}$, i.e., such that they have the infinite square grid graph as a minor. The remaining part of the reduction can be easily done by using CTL formulas only, in a way that is similar to the one explained in the undecidability proof of CTL with minimal model quantifier [1]. Therefore, in the rest of the proof, we focus on the construction of φ_{grd} only. It is important to

[1] F. Mogavero and A. Murano. Branching-Time Temporal Logics with Minimal Model Quantifiers. In DLT'09, LNCS 77 5583, pages 396-409. Springer, 2009.

observe that our formula φ_{grd} is significantly different from the corresponding one used in [1], since we restrict to total structures only.

To distinguish between the four vertexes of each square of the grid, we label all $\mathcal{K}_{grd}$ worlds with the atomic propositions a and b. For the sake of clarity, we name every one of the four possible labelings by means of the Boolean formulas $0 \triangleq \neg a \wedge \neg b$, $1 \triangleq \neg a \wedge b$, $2 \triangleq a \wedge \neg b$, and $3 \triangleq a \wedge b$, called from now one *colors*. Moreover, a necessary condition for $\mathcal{K}_{grd}$ to embed the grid as a minor is the existence of an infinite number of worlds having at least two successors. We use the additional atomic proposition c, called *flag*, to this purpose and require that every world satisfies $\varphi_{flg} \triangleq \mathsf{EX}c \wedge \mathsf{EX}\neg c$. As explained later, the flag is also used to distinguish between the four squares having a given common vertex. In order to encode a square structure, we need to identify a path in $\mathcal{K}_{grd}$ passing trough its four vertexes, whose first four worlds cover the colors 0, 1, 2, and 3 in cyclic increasing order modulo four. This is ensured by requiring every world to satisfy $\varphi_{num} \triangleq \bigwedge_{i=0}^{3} i \rightarrow \mathsf{AX}((i+1) \bmod 4)$, which intuitively asserts that, if a world is colored by $i \in [0,3]$, all its successors are colored by $(i+1) \bmod 4$. Observe that this formula also ensures that all cycles in $\mathcal{K}_{grd}$ have length multiple of four (see Figure 8).

At this point, to build the four squares of the grid having a given common vertex w of $\mathcal{K}_{grd}$, we need to identify four tracks starting and ending in w of length five, which, from now on, we call 4-*tracks*, since they corresponds to four adjacent edges in the underlying graph. To every 4-track, a notion of *parity* is also associated, which accounts for whether the number of occurrences of the flag c in its first four worlds is even or not. The formula φ_{num} already guaranties that every 4-track from w reaches a world with the same coloring as w itself. For example, if w is the central node of the Ks $\mathcal{K}_{grd}$ of Figure 8, there are other eight worlds with the same color reachable from w through some 4-track. To tell the four 4-tracks leading to w apart from the other ones, we we exploit the following observation. For every world w, there are sixteen 4-tracks starting from w, eight of which end in a world with the same flag as w itself. The latter ones can be further split in two groups, one of which contains only 4-tracks of even parity. For this reason, we encode a grid in which the 4-tracks leading from w to w have even parity. The following auxiliary CTL* path formula $\psi_{pth} \triangleq \bigvee_{(\flat_0,\flat_1,\flat_2,\flat_3)\in \mathrm{P}} (\flat_0 \wedge \mathsf{X}(\flat_1 \wedge \mathsf{X}(\flat_2 \wedge \mathsf{X}(\flat_3 \wedge \mathsf{X}\flat_0))))$, with $\mathrm{P} \triangleq \{(\neg c, \neg c, \neg c, \neg c), (\neg c, \neg c, c, c), (\neg c, c, \neg c, c), (\neg c, c, c, \neg c), (c, \neg c, \neg c, c), (c, \neg c, c, \neg c), (c, c, \neg c, \neg c), (c, c, c, c)\}$, precisely characterizes the 4-tracks with even parity that start and end with the same flag. To enforce that such 4-tracks actually start and end in the same world w, we need to require on w itself the following formula: $\varphi_{cyc} \triangleq \bigwedge_{\flat\in\{c,\neg c\}} \mathbb{G}\varphi^{\flat}_{cyc}$, where $\varphi^{\flat}_{cyc} \triangleq \mathsf{E}(\psi_{pth} \wedge \mathsf{X}\flat \wedge \mathsf{X}^4\mathsf{EX}\neg\flat) \rightarrow \mathsf{EX}\neg\flat$. Indeed, suppose by contradiction that there is a path $\pi \in \mathrm{Pth}_{\mathcal{K}_w}$, with $\mathcal{K}_w, \pi, 0 \models \psi_{pth} \wedge \mathsf{X}\flat \wedge \mathsf{X}^4\mathsf{EX}\neg\flat$, that does not close the cycle on w after 4 steps, i.e., $(\pi)_4 \neq (\pi)_0 = w$. Now, let $\mathcal{K}' \in \mathfrak{S}_{\mathcal{K}_w}(\mathfrak{f})$ be one of the minimal substructures of $\mathcal{K}_w$ such that $\pi \in \mathrm{Pth}_{\mathcal{K}'}$. It is immediate to see that $\mathcal{K}' \models \mathsf{E}(\psi_{pth} \wedge \mathsf{X}\flat \wedge \mathsf{X}^4\mathsf{EX}\neg\flat)$. However, $\mathcal{K}' \not\models \mathsf{EX}\neg\flat$, since there is just one $v \in \mathrm{W}_{\mathcal{K}}$ such that $R_{\mathcal{K}'}(w) = \{v\}$ and $\mathcal{K}_v \models \flat$. This latter fact is due to the fact we require $\mathsf{EX}\neg\flat$ on $(\pi)_4$ in $\mathcal{K}'$ but not on $w = (\pi)_0 \neq (\pi)_4$.

At this point, requiring φ_{flg}, φ_{num}, and φ_{cyc} on all worlds of $\mathcal{K}_{grd}$ simply amounts to require the formula $\mathsf{AG}(\varphi_{flg} \wedge \varphi_{num} \wedge \varphi_{cyc})$ to be satisfied by $\mathcal{K}_{grd}$. Notice however that this formula is also satisfied on the quatrefoil Ks partially depicted in Figure 9, where the central world has more than one successor and predecessor with the same flag. To discard the Kss of that form, we need to enforce uniqueness of $\flat$-successors and $\flat$-predecessor, for each flag $\flat \in \{c, \neg c\}$.

For the uniqueness of $\flat$-successors, it suffices to require the formula $\mathbb{G}\varphi^{\flat}_{suc}$, where $\varphi^{\flat}_{suc} \triangleq \varphi^{\flat}_{s} \rightarrow \mathsf{EX}\varphi_{flg}$ and $\varphi^{\flat}_{s} \triangleq \mathsf{EX}(\flat \wedge \mathsf{EX}c) \wedge \mathsf{EX}(\flat \wedge \mathsf{EX}\neg c)$, ensure Indeed, suppose by contradiction that there are two $\flat$-successors. Due to φ_{flg}, both have at least two successors, one satisfying c and the other one $\neg c$. Consequently, the lattice operator $\mathbb{G}$ is able to select a minimal substructure w.r.t. $\varphi^{\flat}_{s}$ containing exactly two $\flat$-successors, one reaching only c and the other one only $\neg c$. However, such a substructure does not satisfy $\mathsf{EX}\varphi_{flg}$. For this reason, we set $\varphi_{suc} \triangleq \bigwedge_{\flat\in\{c,\neg c\}} \mathbb{G}\varphi^{\flat}_{suc}$. Similarly, the formula $\varphi_{pre} \triangleq \bigwedge_{\flat\in\{c,\neg c\}} \mathbb{G}\varphi^{\flat}_{pre}$, where $\varphi^{\flat}_{pre} \triangleq \varphi^{\flat}_{p} \rightarrow \mathsf{EX}^3\varphi_{flg}$ and $\varphi^{\flat}_{p} \triangleq \mathsf{E}(\psi_{pth} \wedge \mathsf{X}^3(\flat \wedge \mathsf{EX}c)) \wedge \mathsf{E}(\psi_{pth} \wedge \mathsf{X}^3(\flat \wedge \mathsf{EX}\neg c))$, ensure the uniqueness of $\flat$-predecessors.

We can finally conclude that the WSTL* formula $\varphi_{grd} \triangleq \mathsf{AG}(\varphi_{flg} \wedge \varphi_{num} \wedge \varphi_{cyc} \wedge \varphi_{suc} \wedge \varphi_{pre})$ has precisely the Ks $\mathcal{K}_{grd}$ of Figure 8 as model. ■

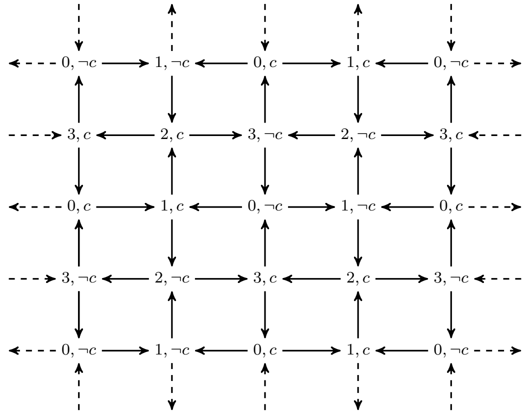

Figure 8: WSTL*[Ks] undecidability ($\mathcal{K}_{grd}$).

Theorem VI.2 (STL*[Ks] Decidable Model Checking)**.** *STL*[Ks] model-checking problem is decidable in* PSPACE *w.r.t. both the size of the* STL* *formula* φ *and the finite* Ks *model* $\mathcal{K}$.

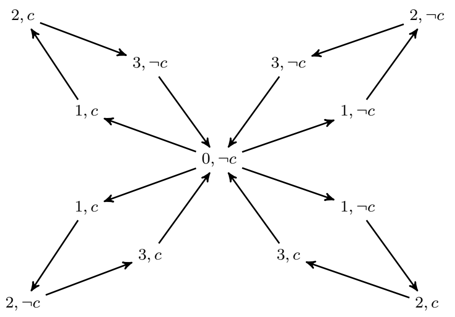

Figure 9: A quatrefoil KS.

Proof: The proof proceeds by induction on the nesting of semilattice operators. The base case is immediate, due to the fact that φ is actually a CTL* formula, for which it is known that the model-checking problem is decidable in PSPACE w.r.t. the size of φ and in LOGSPACE w.r.t. the size of $\mathcal{K}$ [14]. For the inductive case, suppose that the statement is true for all STL* formulas with nesting less than or equal to $n \in \mathbb{N}$. W.l.o.g., we just consider the case in which $\varphi = \varphi_1 \mathbb{U}[\phi] \varphi_2$ has nesting equal to $n+1$, since the remaining cases are immediate consequence of this one. This implies that φ_1, φ_2, and ϕ have nesting of at most n. At this point, the verification procedure for $\mathcal{K} \models \varphi$ is split in the following phases.

1) Identification of the subset $\mathrm{W}' \subseteq \mathrm{W}_{\mathcal{K}}$ such that $w \in \mathrm{W}'$ iff $\mathcal{K}_w \models \phi$, for all $w \in \mathrm{W}_{\mathcal{K}}$.
2) Guess of the substructure $\mathcal{K}' \in \mathfrak{F}_{\mathcal{K}}(\mathrm{W}')$.
3) Check for $\mathcal{K}' \models \varphi_2$.
4) Guess of the substructure $\mathcal{K}'' \in \mathfrak{F}_{\mathcal{K}}(\mathrm{W}')$ such that $\mathcal{K}' \sqsubset \mathcal{K}''$.
5) Check for $\mathcal{K}'' \not\models \varphi_1$.

Since, by the inductive hypothesis, all phases can be executed by a nondeterministic Turing machine linearly-bounded in both the size of the formula φ and of the model $\mathcal{K}$, and since PSPACE = NPSPACE, the thesis follows for φ too. ■

***Results on* KTs:**

Theorem VI.3 (STL*[KT] Decision Problem Complexity)**.** STL*[KT] *satisfiability and model-checking problems have a* $(k+1)$-EXPTIME *formula complexity w.r.t. the alternation* k *of semilattice operators in the* STL* *formula* φ*. The latter problem has a* PTIME *data complexity w.r.t. the size of the finite* KS $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$ *encoding the* KT *model* $\mathcal{K}^{\mathcal{U}}$.

Proof: Similarly to the case of CTL*, the proof proceeds by first constructing an alternating parity tree automaton (APT, for short) $\mathcal{A}_\varphi$ for the STL* formula φ and then by verifying the emptiness of the one-letter automaton obtained by the product of $\mathcal{A}_\varphi$ with the KS $\mathcal{K}$. To build the APT $\mathcal{A}_\varphi$, we use the following inductive procedure on the structure of the formula φ. Observe that $\mathcal{A}_\varphi$, in addition to the description of the KT $\mathcal{K}^U$, reads a labeling identifying the substructure $\mathcal{U}$ of $\mathcal{K}^U$ on which to verify φ. In the base case, φ is a CTL* formula. Therefore, we set $\mathcal{A}_\varphi$ to be the classic alternating hesitant automaton described in [14]. For the inductive case, consider w.l.o.g. $\varphi = \varphi_1 \mathbb{U}[\phi] \varphi_2$ and suppose to have already built the APTs $\mathcal{A}_{\varphi_1}$, $\mathcal{A}_{\varphi_2}$, and $\mathcal{A}_\phi$. By using the APTs $\mathcal{A}_{\varphi_1}$ and $\mathcal{A}_\phi$, via a projection operation, we construct an automaton $\mathcal{A}'_{\varphi_1}$ verifying the formula φ_1 on all substructures $\mathcal{U}'' \in \mathfrak{F}_{\mathcal{U}}(\phi)$ of the $\mathcal{U}$ that has been read in input, which are superstructures of the other one $\mathcal{U}'$ identified by the additional labeling. Then, by making the product of $\mathcal{A}'_{\varphi_1}$ with $\mathcal{A}_{\varphi_2}$ and projecting out the labeling identifying the substructure $\mathcal{U}'$, we obtain the desired automaton $\mathcal{A}_\varphi$ for φ. Observe that, due to the projection operations, the latter automaton has size non-elementary in the alternation of the semilattice operators contained in φ. ■

Theorem VI.4 (STL*[KT] Decision Problem Hardness)**.** STL*[KT] *satisfiability and model-checking problems are* k-EXPSPACE-HARD *w.r.t. the alternation* k *of the* STL* *formula* ψ*. Moreover, the latter problem is* PTIME-HARD *w.r.t. the size of the finite* KS $\mathcal{K} \in \mathrm{KS}(\mathrm{AP})$ *encoding the* KT *model* $\mathcal{K}^{\mathcal{U}}$.

Proof: To prove the hardness results w.r.t. the formula complexity for both the model checking and satisfiability of STL*, we make a linear reduction from QPTL satisfiability problem [23], which is known to be k-EXPSPACE-HARD in the alternation $\mathsf{alt}(\psi)$ of the QPTL formula ψ.

Let us first consider the case of the model-checking problem. Assume $\mathrm{AP} = \{p_1, \ldots, p_n\}$ to be the set of all atomic propositions occurring in ψ and, for a given fresh element $\top$, set $\mathrm{AP}^\top \triangleq \mathrm{AP} \cup \{\top\}$. The idea is to reduce the satisfiability of ψ to the model checking of a suitable STL* formula $\widetilde{\psi}$ against the KS $\mathcal{K}_{MC}$ over $\mathrm{AP}^\top$ of Figure 10, in which each pair of worlds $w_i^\top$ and $w_i^\bot$, for $i \in [1, n]$, encodes the truth values, true and false, of the corresponding proposition p_i. The initial world w_0, used to reach such worlds, also allows to encode the linear structure of QPTL models. Observe that it is the unique world satisfying the Boolean formula $\flat \triangleq \bigwedge_{p \in \mathrm{AP}} \neg p$. Moreover, every subset of worlds w_i^ℓ, with $\ell \in \{\top, \bot\}$, containing exactly one between $w_i^\top$ and $w_i^\bot$, for each $i \in [1, n]$, corresponds to a possible assignment for the propositions in AP. Now, in order to encode an existential quantifier $\exists p_i$ we need to select a single truth value for each time instant and, so, a single successor between $w_i^\top$ and $w_i^\bot$. The formula $\flat_i \triangleq \mathsf{EG}(\flat \wedge \neg \varphi_i)$, with $\varphi_i \triangleq \mathsf{EX}(p_i \wedge \top) \wedge \mathsf{EX}(p_i \wedge \neg \top)$, for each $i \in [1, n]$, ensures that every world of the KT $\mathcal{K}_{MC}^U$ satisfying $\flat$ has at most one successor for each proposition p_i encoding its truth value. However, the maximal substructure of $\mathcal{K}_{MC}^U$ satisfying $\flat_i$ surely have at least one of these successors. Therefore, the construct $\mathtt{EMax}(\flat_i, \varphi)$ select a maximal substructure of $\mathcal{K}_{MC}^U$ w.r.t. $\flat_i$ which must also satisfy the formula φ and whose worlds satisfying $\flat$ have a single successor encoding a possibly different truth value of the atomic proposition p_i. In a similar way, the construct $\mathtt{AMax}(\flat_i, \varphi)$ can be used to encode the universal quantification $\forall p_i$. We can now define the following translation function $\widetilde{\cdot}$: QPTL → STL*.

- $\widetilde{\exists p_i \,.\, \psi'} \triangleq \mathtt{EMax}(\flat_i, \widetilde{\psi'})$.
- $\widetilde{\forall p_i \,.\, \psi'} \triangleq \mathtt{AMax}(\flat_i, \widetilde{\psi'})$.
- $\widetilde{\psi'} \triangleq \mathsf{E}(\mathsf{G}\flat \wedge \psi'')$, for the LTL formula ψ', where $\psi'' \triangleq \psi'[p_i/\mathsf{EX}(p_i \wedge \top) | i \in [1, n]]$ is obtained from ψ' by replacing each atomic proposition $p_i \in \mathrm{AP}$ occurring in it with the CTL formula $\mathsf{EX}(p_i \wedge \top)$.

It is easy to observe that $|\widetilde{\psi}| = \mathrm{O}(|\psi|)$ and $\mathsf{alt}(\widetilde{\psi}) = \mathsf{alt}(\psi)$. Intuitively, this translation replace each propositional quantification by a suitable choice of a subtree of $\mathcal{K}^U_{MC}$. Moreover, the verification of the truth value of a proposition p_i, for a given instant of the time, is done by checking the existence of a successor of such an instant that is labeled with both p_i and the auxiliary symbol $\top$. At this point, an easy induction on the structure of the formula ψ allows to prove that ψ is satisfiable iff $\mathcal{K}^U_{MC} \models \widetilde{\psi}$. Hence, the thesis for the model-checking problem follows.

Let now consider the case of the satisfiability problem. To make the reduction, we check for the satisfiability of the STL* formula $\varphi_{\mathcal{K}} \wedge \widetilde{\psi}$, where $\varphi_{\mathcal{K}}$ is used to characterize the KTs of the same form of tree structure $\mathcal{T}_{Sat}$ depicted in Figure 11, which corresponds to the unwindings of KSs that equals to $\mathcal{K}_{MC}$ except, possibly, for the self loops on the worlds w^{ℓ}_i, with $\ell \in \{\top, \bot\}$. First, we have to ensure that all $\varphi_{\mathcal{K}}$ models contain a spine globally satisfying $\flat$, whose worlds have two successors for each proposition p_i, one labeled by $\top$ and the other one not. This can be easily achieved by using the CTL formula $\varphi_{spn} \triangleq \flat \wedge \mathsf{AG}(\flat \rightarrow (\mathsf{EX}\flat \wedge \bigwedge_{i=1}^{n} \varphi_i))$, where φ_i is the same formula used above for the hardness of model-checking problem. Moreover, to enforce that the flow of time is linear, we have to impose uniqueness of that spine. This can be ensured by means of the WSTL formula $\varphi_{min} \triangleq \mathbb{G}((\mathsf{AG}\flat) \rightarrow \mathtt{Min}(\mathfrak{t}))$. Therefore, we set $\varphi_{\mathcal{K}} \triangleq \varphi_{spn} \wedge \varphi_{min}$. At this point, again by induction on the structure of the formula ψ, it is possible to prove that ψ is satisfiable iff $\varphi_{\mathcal{K}} \wedge \widetilde{\psi}$ is. Hence, the thesis for the satisfiability problem follows.

Finally, to prove the PTIME hardness of the model checking w.r.t. the size of the finite encoding of the model, we simply make a reduction from the reachability problem on And-Or graphs [2]. In particular, we check the formula $\mathtt{EMin}_{Or}(\mathfrak{t}, \mathtt{AMin}_{And}(\mathfrak{t}, \mathsf{EF}p))$ against the unwinding of the KS $\mathcal{K}_{\mathcal{G}}$ obtained from the And-Or graph $\mathcal{G}$, in which the reachability target is identified with the proposition p and each And (resp., Or) node is represented by a world labeled by the proposition And (resp., Or). ■

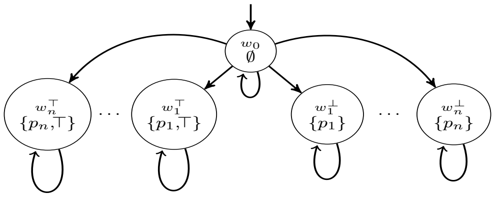

Figure 10: STL*[KT] model checking hardness ($\mathcal{K}_{MC}$).

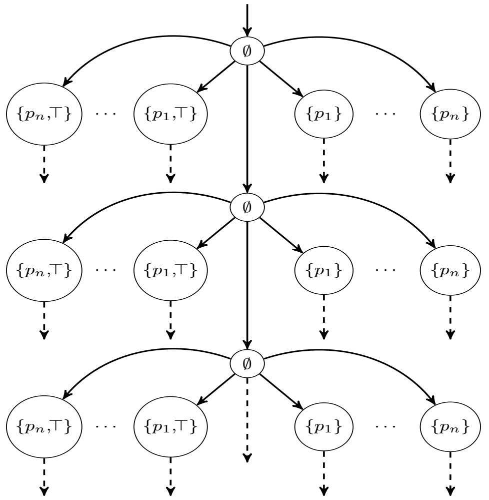

Figure 11: STL*[KT] satisfiability hardness ($\mathcal{T}_{Sat}$).

[2] N. Immerman. Number of Quantifiers is Better Than Number of Tape Cells. JCSS, 22(3):384-406, 1981.