Reasoning About Substructures and Games

MASSIMO BENERECETTI, FABIO MOGAVERO, and ANIELLO MURANO, Università degli Studi di Napoli Federico II

Many decision problems in formal verification and design can be suitably formulated in game-theoretic terms. This is the case for the model checking of open and closed systems and both controller and reactive synthesis. Interpreted in this context, these problems require one to find a strategy (i.e., a plan) to force the system to fulfill some desired goal, no matter what the opponent (e.g., the environment) does. A strategy essentially constrains the possible behaviors of the system to those that are compatible with the decisions dictated by the plan itself. Therefore, finding a strategy to meet some goal basically reduces to identifying a portion of the model of interest (i.e., one of its substructures) that satisfies that goal. In this view, the ability to reason about substructures becomes a crucial aspect for several fundamental problems.

In this article, we present and study 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. The logic is obtained by adding four new temporal-like 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 one to capture in a very natural way many well-known problems, such as module checking, reactive synthesis, and reasoning about games in a wide sense. 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.

Categories and Subject Descriptors: F.3.1 [**Logics and Meanings of Programs**]: Specifying and Verifying and Reasoning about Programs—*Specification techniques*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Modal logic; Temporal logic*

General Terms: Theory, Specification, Verification

Additional Key Words and Phrases: Temporal logics, reasoning about games, quantification on substructures

ACM Reference Format:
Massimo Benerecetti, Fabio Mogavero, and Aniello Murano. 2015. Reasoning about substructures and games. ACM Trans. Comput. Logic 16, 3, Article 25 (June 2015), 51 pages.
DOI: http://dx.doi.org/10.1145/2757286

1. INTRODUCTION

Since the seminal papers by Pnueli [1977, 1981], *temporal logic*, a special kind of *modal logic* geared toward 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

An extended version of the article [Benerecetti et al. 2013] appeared in the proceedings of LICS'13.
This article is partially supported by the FP7 European Union project 600958-SHERPA, the IndAM 2013 project "Logiche di Gioco Estese," and the Embedded System Cup Project B25B09090100007 (POR Campania FSE 2007/2013, asse IV e asse V), Italian Ministry of University and Research.
Authors' addresses: M. Benerecetti, F. Mogavero, and A. Murano, Department of Electrical Engineering and Information Technology, Via Claudio, 21, I-80125, Napoli, Italy; emails: bene@na.infn.it, fm@fabiomogavero.com, murano@na.infn.it.

DOI: http://dx.doi.org/10.1145/2757286

temporal logics are mainly considered in the literature [Lamport 1980]. In *linear-time temporal logics*, such as LTL [Pnueli 1977], 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 [Clarke and Emerson 1981], CTL$^+$ [Emerson and Halpern 1985], and CTL* [Emerson and Halpern 1986], 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 noncompletely 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* [Clarke and Emerson 1981; Queille and Sifakis 1981; Clarke et al. 1986, 2002] is a well-established formal method that allows one 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* [Kripke 1963] or *labeled transition systems* [Keller 1976]. A more challenging problem, from the standpoint of design, is *synthesis* [Church 1963], 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 [Pnueli and Rosner 1989; Rosner 1992], who introduced LTL *reactive synthesis*.

Over the years, an enormous body of work has been devoted to increase the expressive power of temporal logics 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 is to instead extend the syntax by replacing or introducing new operators. The success of the resulting extensions often depends on the ratio between the achieved gain in expressiveness or succinctness and the consequent increase in the complexity of the corresponding 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 [Emerson and Lei 1986; Francez 1986]. 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 [Kupferman et al. 2001], which corresponds to model checking in the context of *open system* analysis. An open system is modeled as a module interacting with the environment, that is, a Kripke structure whose worlds are partitioned into system and environment states, 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 a witness belong 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 *multiagent games*, such as ATL [Alur et al. 2002] and SL [Mogavero et al. 2010, 2012, 2013, 2014]. These logics

syntactically extend classic temporal logics by means of suitable modal operators that quantify over agent strategies, in order to express properties about cooperation and competition among agents. In particular, these modalities allow for 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* [Gerbrandy and Groeneveld 1997; Plaza 2007] and *sabotage logic* [van Benthem 2005; Löding and Rohde 2003], both of which contain operators able to select and predicate on parts of the model under exam. These two languages also can be seen as logics about dynamically changing structures driven by the context.

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 one to extract and analyze portions of the model of interest. In other words, within these logics, one can express and verify specific requirements over particular substructures either of the original model or of its unwinding.

For example, on the side of semantic extensions, CTL with fairness allows one 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 pruning 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 dependencies between input and output signals, which satisfies a given specification. On the side of syntactic extensions, the strategy quantifiers available in almost all logics to reasoning about multiagent 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 multimodal 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 of sabotage logic, and it is explicit in the definition of the interpretations of their characterizing modal operators.

Many of the practical problems that have fostered the development of these extensions can also be recast in game-theoretic terms. The module checking and both controller and reactive synthesis, for example, can all be seen as instances of two-player games between the system and the environment. Interpreted in this context, these problems require one to find a strategy (i.e., a plan) to force the system to fulfill some desired goal, no matter what the environment does. A strategy for the system constrains its possible behaviors to those which are compatible with the decisions dictated by the associated plan. Therefore, finding a strategy to meet some goal basically reduces to identifying a substructure of the original model that satisfies that goal. In this view, most of the extensions described previously can be reinterpreted as attempts to incorporate game-theoretic aspects into temporal logic, either implicitly at the semantic level or explicitly at a syntactic one. These observations suggests that reasoning about substructures appears as an essential feature when reasoning about games in a general sense.

In this article, we propose and study a new temporal 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 counterparts are four new constructs, called *semilattice operators*, provided to switch reasoning between the two different levels. The semantics of the semilattice operators resembles one of the classic *"until," "release," "since,"* and *"back to"* 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 we want to reason about, an additional parameter of the semilattice operators, called *selector parameter*, is provided. This parameter allows one to select, as elements of the semilattice, precisely those substructures preserving the desired information. As a consequence, different substructure semilattices can be induced by a single structure, depending on the chosen selector parameter. This turns out to be a crucial element to allow for modeling (e.g., reasoning about multiagent games), where decisions of different players must preserve decisions made by the other players.

The resulting logic is based on classic concepts in the context of formal verification, such as Kripke structure and temporal operators, both at the semantic level and at the syntactic one. It therefore remains well within the conceptual boundaries of temporal logic notions, without the need to resort to external and more complex ones, like that of strategy and of game structure. In this sense, STL* serves as a purely temporal framework to reason about open systems and games in general. STL* turns out to be very expressive, allowing one 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 toward providing a unifying temporal framework, encompassing those previous approaches. Depending on 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 nonelementary time when interpreted over regular infinite trees. On the other hand, the model-checking problem can be solved under both interpretations, being decidable in PSPACE and in nonelementary time, respectively.

Organization. The article is organized as follows. Section 2 provides some basic definitions and the underlying semantic framework for the logic. The syntax and the semantics are presented in Section 3, where some basic properties are discussed as well. Section 4 discusses several interesting properties about substructures that can be expressed by the logic. Those properties are then applied to model some concrete problems in Section 5, where we show that module checking, turn-based and concurrent games, reactive synthesis and various forms of Nash equilibria can all be captured very naturally within the logic. Sections 6 and 7 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 some of its fragments are reported and discussed. Finally, some conclusions and future work are also proposed.

2. PRELIMINARIES

We will first provide some preliminary definitions and notations, which are needed to set the basis for the semantic framework of the logic STL*.

2.1. Basic Definitions

A *Kripke structure* (KS, for short) [Kripke 1963] over a finite set of *atomic propositions* AP is a tuple $\mathcal{K} \triangleq \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_I \rangle \in \mathrm{KS}(\mathrm{AP})$, where W is an enumerable nonempty 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 domain with $\mathrm{W}' \triangleq R^*(w)$, containing only those worlds that reachable from w; and its labeling function with the corresponding restriction $\mathsf{L}_{\restriction \mathrm{W}'}$ of L to W'. Notice that, since we are interested in expressing temporal properties, there is no loss of generality in requiring the reachability constraint on the transition relation, due to the fact that each portion of a KS not reachable from the initial world does not affect the satisfiability of a temporal formula.

Given a sequence of symbols $\sigma = \sigma_0 \cdot \sigma_1 \cdots \in \Sigma^\infty$, for a generic set Σ, we denote by $\mathsf{fst}(\sigma) \triangleq \sigma_0$ and $(\sigma)_i \triangleq \sigma_i$ the first and ith element of σ itself, where $i \in [0, |\sigma|[$. Moreover, $(\sigma)_{\leq i} \triangleq \sigma_0 \cdots \sigma_i$ and $(\sigma)_{\geq i} \triangleq \sigma_i \cdot \sigma_i \cdots$ represent the prefix up to and the suffix from position i. If the sequence $\sigma = \sigma_0 \cdot \sigma_1 \cdots \sigma_n$ is finite, we also indicate with $\mathsf{lst}(\sigma) \triangleq \sigma_n$ its last element.

A *track* (*path*, respectively) in $\mathcal{K}$ is a finite (infinite, respectively) sequence of worlds $\rho \in \mathrm{Trk} \subseteq \mathrm{W}^+$ ($\pi \in \mathrm{Pth} \subseteq \mathrm{W}^\omega$, respectively) such that (1) $\mathsf{fst}(\rho) = w_0$ ($\mathsf{fst}(\pi) = w_0$, respectively) and (2), for all $i \in [0, |\rho| - 1[$ ($i \in \mathbb{N}$, respectively), it holds that $((\rho)_i, (\rho)_{i+1}) \in R$ ($((\pi)_i, (\pi)_{i+1}) \in R$, respectively). Intuitively, tracks (paths, respectively) of a KS $\mathcal{K}$ are legal sequences of reachable worlds that can be seen as partial (complete, respectively) descriptions of possible *computations* of the system modeled by $\mathcal{K}$.

In the following, we use the name of a KS as subscript to extract the components from its tuple structure; that is, if $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_I \rangle$, we have that $\mathrm{W}_\mathcal{K} \triangleq \mathrm{W}$, $R_\mathcal{K} \triangleq R$, $\mathsf{L}_\mathcal{K} \triangleq \mathsf{L}$, 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 unambiguously 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 (1) $\mathrm{W}_\mathcal{T} \subseteq \mathrm{Dir}^*$ is a Dir-tree for a set Dir of directions, (2) $w_0\mathcal{T} = \epsilon$, and (3), for all $t \in \mathrm{W}_\mathcal{T}$ and $d \in \mathrm{Dir}$, it holds that $t \cdot d \in \mathrm{W}_\mathcal{T}$ if and only if $(t, t \cdot d) \in R_\mathcal{T}$.

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 (1) $\mathrm{W}_\mathcal{K}$ is the set of its directions, (2) 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 $w_0\mathcal{K}$, (3) $(\rho, \rho \cdot w) \in R_{\mathcal{K}^U}$ if and only if $(\mathsf{lst}(w_0\mathcal{K} \cdot \rho), w) \in R_\mathcal{K}$, and (4) there is a surjective function $\mathsf{unw} : \mathrm{W}_{\mathcal{K}^U} \to \mathrm{W}_\mathcal{K}$, called *unwinding function*, such that (5a) $\mathsf{unw}(\rho) = \mathsf{lst}(w_0\mathcal{K} \cdot \rho)$ and (5b) $\mathsf{L}'(\rho) = \mathsf{L}(\mathsf{unw}(\rho))$, 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 $\mathrm{AP} \triangleq \{\bullet, \blacksquare, \blacklozenge\}$ and its unwinding $\mathcal{K}^U$. We use them as a running example along the whole article. Note also that we are assuming all worlds in $\mathcal{K}$ to be labeled by their own shapes. Therefore, AP is the set of $\mathcal{K}^U$ directions too. In addition, 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 $\mathcal{K}$ (see the definition of the unwinding function unw introduced earlier).

2.2. Substructure Semilattice

At the base of the semantics definition of the logic we introduce resides 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 (1) $\mathrm{W}_{\mathcal{K}'} \subseteq \mathrm{W}_\mathcal{K}$,

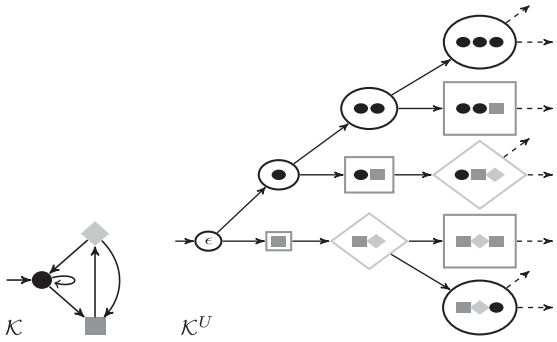

Fig. 1. A KS $\mathcal{K}$ and its unwinding $\mathcal{K}^U$.

(2) $R_{\mathcal{K}'} \subseteq R_{\mathcal{K}} \cap (\mathrm{W}_{\mathcal{K}'} \times \mathrm{W}_{\mathcal{K}'})$, (3) $\mathsf{L}_{\mathcal{K}'} = (\mathsf{L}_{\mathcal{K}})_{\restriction \mathrm{W}_{\mathcal{K}'}}$, and (4) $\mathtt{w}_0\mathcal{K}' = \mathtt{w}_0\mathcal{K}$. Moreover, $\mathcal{K}$ and $\mathcal{K}'$ are *comparable* if (1) $\mathcal{K} \sqsubseteq \mathcal{K}'$ or (2) $\mathcal{K}' \sqsubseteq \mathcal{K}$ holds; otherwise, they are *incomparable*. Observe that $\sqsubseteq$ is 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 $\aleph \subseteq \mathrm{KS(AP)}$ and a KS $\mathcal{K} \in \aleph$, we say that $\mathcal{K}$ is *minimal* (*maximal*, respectively) in $\aleph$, or simply *minimal* in case $\aleph$ equals to KS(AP), if there is no KS $\mathcal{K}' \in \aleph$ such that $\mathcal{K}' \sqsubset \mathcal{K}$ ($\mathcal{K} \sqsubset \mathcal{K}'$, respectively). Observe that minimal elements w.r.t. $\sqsubseteq$ are just those KSs for which the only part reachable from the initial world is either a single lasso or an infinite chain. This implies that $\mathcal{K}$ is minimal if and only if $|\mathrm{Pth}_{\mathcal{K}}| = 1$. Also, note that there are no maximal elements in KS(AP).

In order to identify the particular set of substructures of interest over which we can predicate in the logic, we introduce the notions of downward and upward filterings of a KS given a set of distinguished worlds to preserve in the reasoning.

Let $\mathrm{X} \subseteq \mathrm{W}_{\mathcal{K}}$ be a subset of worlds of a given KS $\mathcal{K} \in \mathrm{KS(AP)}$. Then, by $\mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathrm{X}) \triangleq \{\mathcal{K}' \in \mathrm{KS(AP)} : \mathcal{K}' \sqsubset \mathcal{K} \wedge \forall w \in \mathrm{W}_{\mathcal{K}'} \cap \mathrm{X} \,.\, R_{\mathcal{K}'}(w) = R_{\mathcal{K}}(w)\}$ and $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\mathrm{X}) \triangleq \{\mathcal{K}' \in \mathrm{KS(AP)} : \mathcal{K}' \sqsubseteq \mathcal{K} \wedge \forall w \in \mathrm{W}_{\mathcal{K}'} \cap \mathrm{X} \,.\, R_{\mathcal{K}'}(w) = R_{\mathcal{K}}(w)\} = \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathrm{X}) \cup \{\mathcal{K}\}$, we denote, respectively, the *strict* and *reflexive downward filtering* of $\mathcal{K}$ w.r.t. the *selector* X, that is, the sets of substructures of $\mathcal{K}$ (containing or not $\mathcal{K}$ itself) preserving all edges exiting from worlds in X that are also contained in the substructure $\mathcal{K}$. Observe that the lack of some edges in a substructure may determine the impossibility for a world in X to be reached by the initial one. Consequently, such a world would not be part of its domain. The ordering $\sqsubseteq$ on $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\mathrm{X})$ induces an *upper semilattice*, *a.k.a. join semilattice*, satisfying the following properties: (1) the *maximal element* is $\mathcal{K}$; (2) the *minimal elements* are exactly those KSs having a unique edge outgoing from worlds not in X; and (3) the *join* $\mathcal{K}_1 \sqcup \mathcal{K}_2$ of two elements $\mathcal{K}_1, \mathcal{K}_2 \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\mathrm{X})$ is the KS having the 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 $\mathsf{L}_{\mathcal{K}_1 \sqcup \mathcal{K}_2} \triangleq (\mathsf{L}_{\mathcal{K}})_{\restriction \mathrm{W}_{\mathcal{K}_1 \sqcup \mathcal{K}_2}}$. Observe that the downward filtering is *antimonotone*, that is, $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\mathrm{X}) \subseteq \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\mathrm{Y})$, for all $\mathrm{Y} \subseteq \mathrm{X} \subseteq \mathrm{W}_{\mathcal{K}}$. Moreover, it holds that $|\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\mathrm{X})| = \infty$ if and only if one of the following two conditions holds: (1) there is a world $w \in \mathrm{W}_{\mathcal{K}} \setminus \mathrm{X}$ having an infinite number of outgoing edges, that is, $|R_{\mathcal{K}}(w)| = \omega$, or (2) there are infinitely many worlds out of X with at least two outgoing edges, that is, $|\{w \in \mathrm{W}_{\mathcal{K}} \setminus \mathrm{X} : |R_{\mathcal{K}}(w)| \geq 2\}| = \omega$.

Similarly to the downward filterings of $\mathcal{K}$ w.r.t. X, we denote by $\mathfrak{F}^{\uparrow \mathcal{K}^\star}_{\mathcal{K}}(\mathrm{X}) \triangleq \{\mathcal{K}' \in \mathrm{KS(AP)} : \mathcal{K} \sqsubset \mathcal{K}' \sqsubseteq \mathcal{K}^\star \wedge \forall w \in \mathrm{X} \,.\, R_{\mathcal{K}'}(w) = R_{\mathcal{K}}(w)\}$ and $\overline{\mathfrak{F}}^{\uparrow \mathcal{K}^\star}_{\mathcal{K}}(\mathrm{X}) \triangleq \{\mathcal{K}' \in \mathrm{KS(AP)} : \mathcal{K} \sqsubseteq \mathcal{K}' \sqsubseteq \mathcal{K}^\star \wedge \forall w \in \mathrm{X} \,.\, R_{\mathcal{K}'}(w) = R_{\mathcal{K}}(w)\} = \mathfrak{F}^{\uparrow \mathcal{K}^\star}_{\mathcal{K}}(\mathrm{X}) \cup \{\mathcal{K}\}$ the *strict* and *reflexive upward filterings* of $\mathcal{K}$ w.r.t. X with bound $\mathcal{K}^\star \in \mathrm{KS(AP)}$, that is, the sets of superstructures

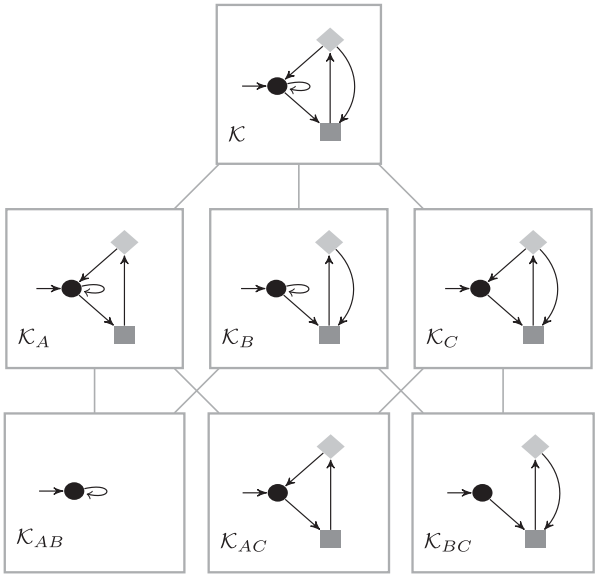

Fig. 2. The substructure semilattice $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\emptyset)$.

of $\mathcal{K}$ (containing or not $\mathcal{K}$ itself) bounded by $\mathcal{K}^\star$ having the same outgoing edges as in $\mathcal{K}$ from all worlds in X. In this case, the ordering $\sqsubseteq$ on $\overline{\mathfrak{F}}^{\uparrow\mathcal{K}^\star}_{\mathcal{K}}(\mathrm{X})$ induces a *lattice* satisfying the following properties: (1) the *minimal element* is $\mathcal{K}$; (2) the *maximal element* is the unique substructure $\mathcal{K}'$ of $\mathcal{K}^\star$ having $R_{\mathcal{K}'}(w) = R_{\mathcal{K}}(w)$, for all $w \in \mathrm{X}$, and $R_{\mathcal{K}'}(w) = R_{\mathcal{K}^\star}(w)$, for all $w \in \mathrm{W}_{\mathcal{K}'} \setminus \mathrm{X}$—that is, all worlds not in X have maximal outgoing edges w.r.t. $\mathcal{K}^\star$; (3) the *join* is defined as for the downward filtering; (4) the *meet* $\mathcal{K}_1 \sqcap \mathcal{K}_2$ of two elements $\mathcal{K}_1, \mathcal{K}_2 \in \overline{\mathfrak{F}}^{\uparrow\mathcal{K}^\star}_{\mathcal{K}}(\mathrm{X})$ is the unique maximal KS having the set of worlds $\mathrm{W}_{\mathcal{K}_1\sqcap\mathcal{K}_2} \subseteq \mathrm{W}_{\mathcal{K}_1} \cap \mathrm{W}_{\mathcal{K}_2}$, transition relation $R_{\mathcal{K}_1\sqcap\mathcal{K}_2} \subseteq R_{\mathcal{K}_1} \cap R_{\mathcal{K}_2}$, and labeling function $\mathsf{L}_{\mathcal{K}_1\sqcap\mathcal{K}_2} \triangleq (\mathsf{L}_{\mathcal{K}})_{\restriction \mathrm{W}_{\mathcal{K}_1\sqcap\mathcal{K}_2}}$. Observe that the upward filtering is antimonotone as well. Moreover, it holds that $\overline{\mathfrak{F}}^{\uparrow\mathcal{K}^\star}_{\mathcal{K}}(\emptyset) \subseteq \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}^\star}(\mathrm{X})$, for all $\mathcal{K} \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}^\star}(\mathrm{X})$ with $\mathrm{X} \subseteq \mathrm{W}_{\mathcal{K}^\star}$. Consequently, $\overline{\mathfrak{F}}^{\uparrow\mathcal{K}^\star}_{\mathcal{K}}(\mathrm{Y}) \subseteq \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}^\star}(\mathrm{X})$, for all $\mathrm{Y} \subseteq \mathrm{W}_{\mathcal{K}}$; that is, an upward filtering $\overline{\mathfrak{F}}^{\uparrow\mathcal{K}^\star}_{\mathcal{K}}(\mathrm{Y})$ is always contained in the downward one $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}^\star}(\mathrm{X})$ from which it is originated, independently from the sets of worlds to be preserved.

In Figures 2 and 3, we depict the Hasse diagrams of, respectively, the semilattice of the substructures in the downward filtering of $\mathcal{K}$ w.r.t. $\emptyset$ and the lattice of superstructures of $\mathcal{K}_{AC}$ w.r.t. $\emptyset$ with bound $\mathcal{K}$. In addition, in Figures 4 and 5, we report the diagrams of, respectively, the subsemilattices and sublattices obtained by restricting the ordering to smaller filterings. Note that no edge that is the unique outgoing one from a world (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. Finally, $\mathcal{K}_{AB}$ belongs to the downward filtering $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\{\blacklozenge\})$, since it does not contain the world ◆; thus, the defining constraint of $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\{\blacklozenge\})$ is trivially satisfied.

3. SUBSTRUCTURE TEMPORAL LOGICS

The *substructure temporal logic* (STL* for short) extends CTL* [Emerson and Halpern 1986] by adding four special ternary constructs, $\varphi_1 \mathbb{U}_\phi \varphi_2$, $\varphi_1 \mathbb{R}_\phi \varphi_2$, $\varphi_1 \mathbb{S}_\phi \varphi_2$, and $\varphi_1 \mathbb{B}_\phi \varphi_2$,

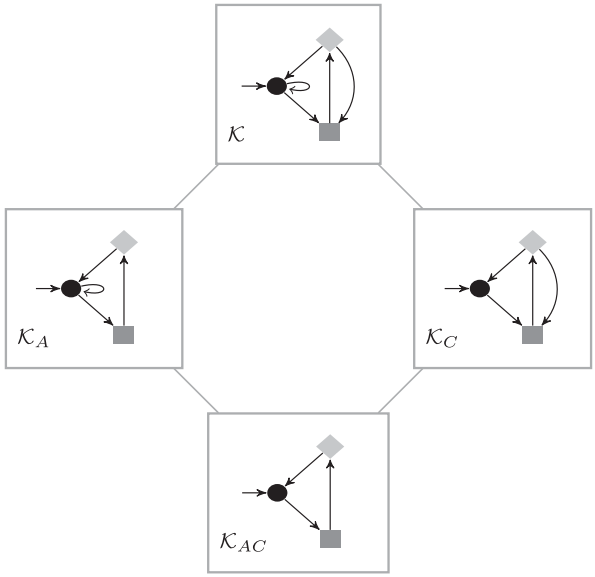

Fig. 3. The superstructure lattice $\overline{\mathfrak{F}}^{\uparrow\mathcal{K}}_{\mathcal{K}_{AC}}(\emptyset)$.

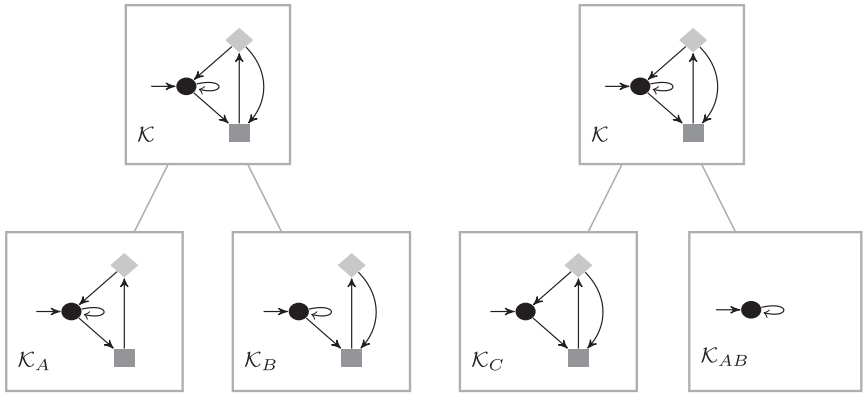

Fig. 4. The downward filterings of $\mathcal{K}$ w.r.t. {●} and {◆}.

called *semilattice operators*. The first two constructs, named *downward operators*, 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 the *selector parameter*, specifies the particular downward semilattice of substructures on which the quantifications act. Specifically, this parameter is used to identify on which worlds of the model pruning edges is forbidden. Dually, the other two constructs, named *upward operators*, can be informally read, respectively, as "there is a strict superstructure satisfying φ_2 such that all its strict substructures satisfy φ_1" and "all strict superstructures satisfy φ_2 unless one of their strict substructures satisfies φ_1," where the selector parameter ϕ specifies the particular upward lattice of superstructures on which the quantifications act. In this case, the parameter is used to identify on which worlds of the model adding edges is forbidden. From a high-level point of view, we can consider these new operators as a strict version of the until, release, since,

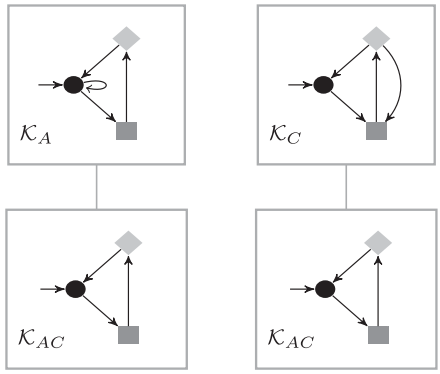

Fig. 5. The upward filterings of $\mathcal{K}_{AC}$ w.r.t. {◆} and {●} with bound $\mathcal{K}$.

and back-to temporal operators acting on substructures and superstructures instead of linear points in time. Just as in CTL*, the path quantifiers $\mathtt{E}$ and $\mathtt{A}$ of STL* can prefix a linear-time formula composed by an arbitrary Boolean combination and nesting of classic temporal operators $\mathtt{X}$, $\mathtt{U}$, and $\mathtt{R}$.

3.1. Syntax

Similarly to CTL*, the formal syntax of STL* includes both path formulas, expressing properties over sequences of worlds, and state formulas, predicating over worlds of a structure or of its substructures. State and path formulas are defined by mutual induction as follows.

Definition 3.1 (STL* *Syntax*). STL* *state* (φ) and *path* (ψ) *formulas* are built inductively from the set AP according to the following grammar, where $p \in \mathrm{AP}$:

(1) $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \mathbb{U}_\varphi \varphi \mid \varphi \mathbb{R}_\varphi \varphi \mid \varphi \mathbb{S}_\varphi \varphi \mid \varphi \mathbb{B}_\varphi \varphi \mid \mathtt{E}\psi \mid \mathtt{A}\psi$;
(2) $\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathtt{X}\psi \mid \psi \mathtt{U} \psi \mid \psi \mathtt{R} \psi$.

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 $\mathtt{t}$ and false $\mathtt{f}$ and the simpler temporal operators eventually $\mathtt{F}\varphi \triangleq \mathtt{t}\,\mathtt{U}\varphi$ and globally $\mathtt{G}\varphi \triangleq \mathtt{f}\,\mathtt{R}\varphi$. We will define the derived constructs $\mathbb{EX}_\phi\varphi \triangleq \mathtt{f}\,\mathbb{U}_\phi\varphi$, $\mathbb{AX}_\phi\varphi \triangleq \mathtt{t}\,\mathbb{R}_\phi\varphi$, $\mathbb{EY}_\phi\varphi \triangleq \mathtt{f}\,\mathbb{S}_\phi\varphi$, and $\mathbb{AY}_\phi\varphi \triangleq \mathtt{t}\,\mathbb{B}_\phi\varphi$, called *immediate substructure/superstructure operators*. Moreover, we introduce the restricted operators $\mathbb{F}_\phi\varphi \triangleq \mathtt{t}\,\mathbb{U}_\phi\varphi$, $\mathbb{G}_\phi\varphi \triangleq \mathtt{f}\,\mathbb{R}_\phi\varphi$, $\mathbb{P}_\phi\varphi \triangleq \mathtt{t}\,\mathbb{S}_\phi\varphi$, and $\mathbb{H}_\phi\varphi \triangleq \mathtt{f}\,\mathbb{B}_\phi\varphi$. Intuitively, these new semilattice operators have the informal meaning of *"there is a strict substructure," "for all strict substructures," "there is a strict superstructure,"* and *"for all strict superstructures."* In addition, we can derive the reflexive versions of all these operators as follows:

$$\text{—}\varphi_1\overline{\mathbb{U}}_\phi\varphi_2 \triangleq \varphi_2 \vee (\varphi_1 \wedge \varphi_1\mathbb{U}_\phi\varphi_2); \qquad \text{—}\overline{\mathbb{F}}_\phi\varphi \triangleq \varphi \vee \mathbb{F}_\phi\varphi;$$
$$\text{—}\varphi_1\overline{\mathbb{R}}_\phi\varphi_2 \triangleq \varphi_2 \wedge (\varphi_1 \vee \varphi_1\mathbb{R}_\phi\varphi_2); \qquad \text{—}\overline{\mathbb{G}}_\phi\varphi \triangleq \varphi \wedge \mathbb{G}_\phi\varphi;$$
$$\text{—}\varphi_1\overline{\mathbb{S}}_\phi\varphi_2 \triangleq \varphi_2 \vee (\varphi_1 \wedge \varphi_1\mathbb{S}_\phi\varphi_2); \qquad \text{—}\overline{\mathbb{P}}_\phi\varphi \triangleq \varphi \vee \mathbb{P}_\phi\varphi;$$
$$\text{—}\varphi_1\overline{\mathbb{B}}_\phi\varphi_2 \triangleq \varphi_2 \wedge (\varphi_1 \vee \varphi_1\mathbb{B}_\phi\varphi_2); \qquad \text{—}\overline{\mathbb{H}}_\phi\varphi \triangleq \varphi \wedge \mathbb{H}_\phi\varphi.$$

Note that, in the rest of the article, we may omit the selector parameter ϕ, whenever it equals to $\mathtt{f}$, in all semilattice operators, as well as in the derived ones later introduced.

By replacing the four constructs $\varphi \mathbb{U}_\phi \varphi$, $\varphi \mathbb{R}_\phi \varphi$, $\varphi \mathbb{S}_\phi \varphi$, and $\varphi \mathbb{B}_\phi \varphi$ with the restricted operators $\mathbb{F}_\phi \varphi$, $\mathbb{G}_\phi \varphi$, $\mathbb{P}_\phi \varphi$, and $\mathbb{H}_\phi \varphi$, in Rule 1 of Definition 3.1, we obtain a family of sublogics of STL* called *weak substructure temporal logics* (WSTL*, WSTL$^+$, and WSTL for short). Similarly, if we only allow the use of the downward operators $\varphi \mathbb{U}_\phi \varphi$ and $\varphi \mathbb{R}_\phi \varphi$, we construct an orthogonal family of sublogics called *downward substructure temporal logics* (DSTL*, DSTL$^+$, and DSTL for short). Finally, from the combination of these two restrictions, we derive the *downward weak substructure temporal logics* (DWSTL*, DWSTL$^+$, and DWSTL for short).

3.2. Semantics

The logic STL* shares with CTL* the same semantics for all the temporal operators, which is given, as usual, w.r.t. a KS. The novel aspect resides in the semilattice operators, which provide a mechanism to quantify over substructures and superstructures of the KS on which they are interpreted. The domains of quantification of these operators are identified by suitable filterings: downward filterings for the downward operators and upward filterings for the upward ones. While the downward filterings of a structure $\mathcal{K}$ are relative only to the current structure $\mathcal{K}$, the definition of the upward filterings is always relative also to a bounding KS $\mathcal{K}^\star$, which sets the domain of the possible states and transitions that can be added to the current one to form its superstructures. Therefore, a suitable notion of a model for an STL* formula needs to take into account two structures: the current one $\mathcal{K}$, where the formula is interpreted, and the bounding one $\mathcal{K}^\star$, which will serve as an upper bound for the admissible superstructures of $\mathcal{K}$ in any upward filtering.

We will, then, write $\mathcal{K} \models^{\mathcal{K}^\star} \varphi$, where $\mathcal{K} \sqsubseteq \mathcal{K}^\star$, to denote that a state formula φ holds in $\mathcal{K}$ or, equivalently, $\mathcal{K}$ is a *model* of φ, once the upper bound $\mathcal{K}^\star$ is given. Moreover, for a path $\pi \in \mathrm{Pth}_\mathcal{K}$ and a number $h \in \mathbb{N}$, we write $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi$ to indicate that a path formula ψ holds on π at position h.

Recall that both the notions of filtering are parameterized by a selector set X, which provides those states whose branching structure must be preserved identically in all the substructures or superstructures. At the syntactic level, this set is encoded by the selector parameter ϕ of the semilattice operators. Intuitively, those states of the model that satisfy ϕ form the selector set X parameterizing the selected filtering. We will write $[\![\phi]\!]^{\mathcal{K}^\star}_\mathcal{K} \triangleq \{w \in \mathrm{W}_\mathcal{K} : \mathcal{K}_w \models^{\mathcal{K}^\star_w} \phi\}$ in place of the denotation of ϕ in the $\mathcal{K}^\star$-bounded model $\mathcal{K}$, that is, the set of states of $\mathcal{K}$ that satisfy ϕ. As a shorthand, we will also write $\overline{\mathfrak{F}}^{\downarrow\mathcal{K}^\star}_\mathcal{K}(\phi) \triangleq \overline{\mathfrak{F}}^{\downarrow}_\mathcal{K}([\![\phi]\!]^{\mathcal{K}^\star}_\mathcal{K})$ and $\overline{\mathfrak{F}}^{\uparrow\mathcal{K}^\star}_\mathcal{K}(\phi) \triangleq \overline{\mathfrak{F}}^{\uparrow\mathcal{K}^\star}_\mathcal{K}([\![\phi]\!]^{\mathcal{K}^\star}_\mathcal{K})$ to denote the filterings identified by the selector parameter ϕ. Similarly, $\mathfrak{F}^{\downarrow\mathcal{K}^\star}_\mathcal{K}(\phi) \triangleq \overline{\mathfrak{F}}^{\downarrow\mathcal{K}^\star}_\mathcal{K}(\phi) \setminus \{\mathcal{K}\}$ and $\mathfrak{F}^{\uparrow\mathcal{K}^\star}_\mathcal{K}(\phi) \triangleq \overline{\mathfrak{F}}^{\uparrow\mathcal{K}^\star}_\mathcal{K}(\phi) \setminus \{\mathcal{K}\}$ are, respectively, the sets of all strict substructures and superstructures of $\mathcal{K}$ preserving the existing branching structure from the worlds satisfying ϕ.

We can now provide the semantics of the logic with the following definition, where the classic semantics of temporal operators and path quantifiers is retained and the semantics of the four new sublattice operators is made precise.

Definition 3.2 (STL* *Semantics*). Given two KSs $\mathcal{K}^\star, \mathcal{K} \in \mathrm{KS(AP)}$ with $\mathcal{K} \sqsubseteq \mathcal{K}^\star$, for all STL* state formulas φ_1, φ_2, and ϕ, it holds that:

(1) $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1 \mathbb{U}_\phi \varphi_2$ if there exists a $\mathcal{K}' \in \mathfrak{F}^{\downarrow\mathcal{K}^\star}_\mathcal{K}(\phi)$ such that $\mathcal{K}' \models^{\mathcal{K}^\star} \varphi_2$ and, for all strict superstructures $\mathcal{K}'' \in \mathfrak{F}^{\downarrow\mathcal{K}^\star}_\mathcal{K}(\phi)$ of $\mathcal{K}'$, it holds that $\mathcal{K}'' \models^{\mathcal{K}^\star} \varphi_1$;

(2) $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1 \mathbb{R}_\phi \varphi_2$ if, for all $\mathcal{K}' \in \mathfrak{F}^{\downarrow\mathcal{K}^\star}_\mathcal{K}(\phi)$, it holds that $\mathcal{K}' \models^{\mathcal{K}^\star} \varphi_2$ or there exists a strict superstructure $\mathcal{K}'' \in \mathfrak{F}^{\downarrow\mathcal{K}^\star}_\mathcal{K}(\phi)$ of $\mathcal{K}'$ such that $\mathcal{K}'' \models^{\mathcal{K}^\star} \varphi_1$;

Table I. Semantics of Standard STL⋆ Constructs

Given two KSs $\mathcal{K}^\star, \mathcal{K} \in \mathrm{KS(AP)}$ with $\mathcal{K} \sqsubseteq \mathcal{K}^\star$ and an STL⋆ state formula φ, the relation $\mathcal{K} \models^{\mathcal{K}^\star} \varphi$ is defined as follows:

(1) $\mathcal{K} \models^{\mathcal{K}^\star} p$ if and only if $p \in \mathsf{L}(w_0)$;
(2) $\mathcal{K} \models^{\mathcal{K}^\star} \neg\varphi$ if and only if not $\mathcal{K} \models^{\mathcal{K}^\star} \varphi$, i.e., $\mathcal{K} \not\models^{\mathcal{K}^\star} \varphi$;
(3) $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1 \wedge \varphi_2$ if and only if $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1$ and $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_2$;
(4) $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1 \vee \varphi_2$ if and only if $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1$ or $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_2$;
(5) $\mathcal{K} \models^{\mathcal{K}^\star} \mathtt{E}\psi$ if and only if there exists a path $\pi \in \mathrm{Pth}_{\mathcal{K}}$ such that $\mathcal{K}, \pi, 0 \models^{\mathcal{K}^\star} \psi$;
(6) $\mathcal{K} \models^{\mathcal{K}^\star} \mathtt{A}\psi$ if and only if for all paths $\pi \in \mathrm{Pth}_{\mathcal{K}}$ it holds that $\mathcal{K}, \pi, 0 \models^{\mathcal{K}^\star} \psi$.

Given a path $\pi \in \mathrm{Pth}_{\mathcal{K}}$, an index $h \in \mathbb{N}$, and an STL⋆ path formulas ψ, the relation $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi$ is defined as follows:

(1) $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \varphi$ if and only if $\widehat{\mathcal{K}} \models^{\widehat{\mathcal{K}^\star}} \varphi$, where $\widehat{\mathcal{K}} \triangleq \mathcal{K}_{(\pi)_h}$ and $\widehat{\mathcal{K}^\star} \triangleq \mathcal{K}^\star_{(\pi)_h}$;
(2) $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \neg\psi$ if and only if not $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi$, i.e., $\mathcal{K}, \pi, h \not\models^{\mathcal{K}^\star} \psi$;
(3) $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi_1 \wedge \psi_2$ if and only if $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi_1$ and $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi_2$;
(4) $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi_1 \vee \psi_2$ if and only if $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi_1$ or $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi_2$;
(5) $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \mathtt{X}\psi$ if and only if $\mathcal{K}, \pi, h+1 \models^{\mathcal{K}^\star} \psi$;
(6) $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi_1 \mathtt{U} \psi_2$ if and only if there exists $i \in \mathbb{N}$ with $h \leq i$ such that $\mathcal{K}, \pi, i \models^{\mathcal{K}^\star} \psi_2$ and, for all $j \in \mathbb{N}$ with $h \leq j < i$, it holds that $\mathcal{K}, \pi, j \models^{\mathcal{K}^\star} \psi_1$;
(7) $\mathcal{K}, \pi, h \models^{\mathcal{K}^\star} \psi_1 \mathtt{R} \psi_2$ if and only if, for all $i \in \mathbb{N}$ with $h \leq i$, it holds that $\mathcal{K}, \pi, i \models^{\mathcal{K}^\star} \psi_2$ or there exists $j \in \mathbb{N}$ with $h \leq j < i$ such that $\mathcal{K}, \pi, j \models^{\mathcal{K}^\star} \psi_1$.

(3) $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1 \mathbb{S}_\phi \varphi_2$ if there exists a $\mathcal{K}' \in \mathfrak{F}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi)$ such that $\mathcal{K}' \models^{\mathcal{K}^\star} \varphi_2$ and, for all strict substructures $\mathcal{K}'' \in \mathfrak{F}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi)$ of $\mathcal{K}'$, it holds that $\mathcal{K}'' \models^{\mathcal{K}^\star} \varphi_1$;
(4) $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1 \mathbb{B}_\phi \varphi_2$ if, for all $\mathcal{K}' \in \mathfrak{F}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi)$, it holds that $\mathcal{K}' \models^{\mathcal{K}^\star} \varphi_2$ or there exists a strict substructure $\mathcal{K}'' \in \mathfrak{F}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi)$ of $\mathcal{K}'$ such that $\mathcal{K}'' \models^{\mathcal{K}^\star} \varphi_1$.

The semantics of all classic CTL⋆ syntactic constructs is defined as usual and reported in Table I.

Observe that, by replacing the set $\mathfrak{F}_{\mathcal{K}}^{\downarrow\mathcal{K}^\star}(\phi)$ with $\overline{\mathfrak{F}}_{\mathcal{K}}^{\downarrow\mathcal{K}^\star}(\phi)$ in Items 1 and 2 of the previous definition, we obtain the semantics of the reflexive operators $\varphi_1 \overline{\mathbb{U}}_\phi \varphi_2$ and $\varphi_1 \overline{\mathbb{R}}_\phi \varphi_2$, respectively. Similarly, by replacing the set $\mathfrak{F}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi)$ with $\overline{\mathfrak{F}}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi)$ in Items 3 and 4, we obtain the semantics of reflexive operators $\varphi_1 \overline{\mathbb{S}}_\phi \varphi_2$ and $\varphi_1 \overline{\mathbb{B}}_\phi \varphi_2$, respectively. In addition, note that the constraint required on $\mathcal{K}''$ in Items 1 and 2 is equivalent to $\mathcal{K}'' \in \mathfrak{F}_{\mathcal{K}}^{\downarrow\mathcal{K}^\star}(\phi) \cap \mathfrak{F}_{\mathcal{K}'}^{\uparrow\mathcal{K}^\star}(\mathtt{f})$ and that in Items 3 and 4 is equivalent to $\mathcal{K}'' \in \mathfrak{F}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi) \cap \mathfrak{F}_{\mathcal{K}'}^{\downarrow\mathcal{K}^\star}(\mathtt{f})$.

For the sake of clarity and succinctness, we prefer the strict version of the semantics instead of the more usual reflexive one. All the reflexive versions can be easily derived from the strict ones.

To better understand the intuition behind the introduced semilattice operators, we present two examples about the downward operators $\mathbb{U}$ and $\mathbb{R}$ based on the KS $\mathcal{K}$ of Figure 1 and its downward filtering $\overline{\mathfrak{F}}_{\mathcal{K}}^{\downarrow}(\emptyset)$ of Figure 2. Consider the formula $\varphi_1 \mathbb{U} \varphi_2$, where $\varphi_1 \triangleq \mathtt{AGEF}\bullet$ and $\varphi_2 \triangleq (\mathtt{AGF}\blacklozenge) \wedge ((\mathtt{AGF}\bullet) \vee (\mathtt{AFG}\neg\bullet))$. 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 can immediately be seen that $\mathcal{K}_{AC}$ satisfies φ_2 and both $\mathcal{K}_A$ and $\mathcal{K}_C$ satisfy φ_1. Thus, as depicted in Figure 6 (we highlight

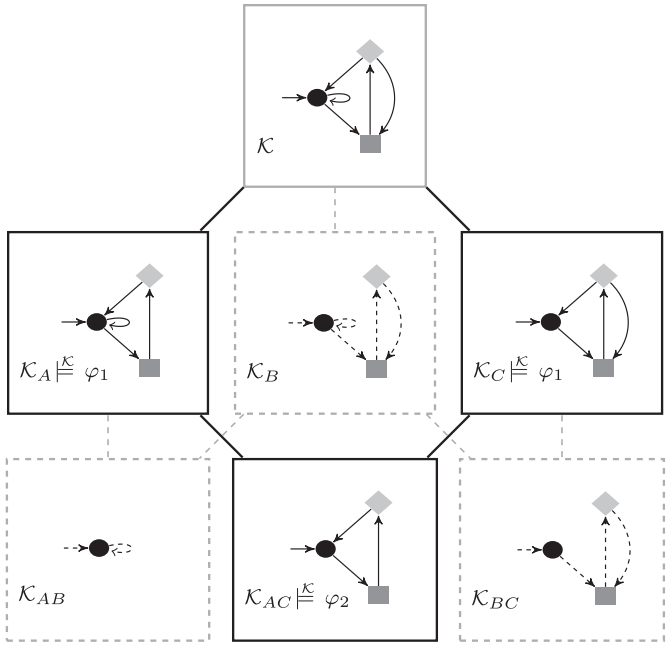

Fig. 6. $\mathbb{U}$ semantics.

the witness by using solid bold lines), we have that $\mathcal{K} \models^{\mathcal{K}} \varphi_1 \mathbb{U} \varphi_2$. 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 $\mathcal{K} \not\models^{\mathcal{K}} \varphi_1 \mathbb{U}_{\bullet} \varphi_2$ and $\mathcal{K} \not\models^{\mathcal{K}} \varphi_1 \mathbb{U}_{\blacklozenge} \varphi_2$, since in the corresponding filterings $\overline{\mathfrak{F}}_{\mathcal{K}}^{\downarrow \mathcal{K}}(\bullet)$ and $\overline{\mathfrak{F}}_{\mathcal{K}}^{\downarrow \mathcal{K}}(\blacklozenge)$ there is no KS satisfying φ_2. Now, consider the formula $\varphi_1 \mathbb{R} \varphi_2$, where $\varphi_1 \triangleq \mathtt{AF}\blacklozenge$ and $\varphi_2 \triangleq \mathtt{EGF}\bullet$. 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 $\overline{\mathfrak{F}}_{\mathcal{K}}(\emptyset)$ but $\mathcal{K}_{BC}$ satisfy φ_2 and $\mathcal{K}_C$ also satisfies φ_1. Thus, as depicted in Figure 7, we have that $\mathcal{K} \models^{\mathcal{K}} \varphi_1 \mathbb{R} \varphi_2$. 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.

As to the upward operators $\mathbb{P}$ and $\mathbb{H}$, let us consider the KS $\mathcal{K}_{AC}$ of Figure 2 and its upward filtering $\overline{\mathfrak{F}}_{\mathcal{K}_{AC}}^{\uparrow \mathcal{K}}(\emptyset)$ with bound $\mathcal{K}$, depicted in Figure 3. It can immediately be seen that $\mathcal{K}$ is the unique strict superstructure in that filtering that satisfies the formula $\varphi \triangleq (\mathtt{EG}\bullet) \wedge (\mathtt{EFG}\neg\bullet)$. Thus, we have $\mathcal{K}_{AC} \models^{\mathcal{K}} \mathbb{P}\varphi$ but $\mathcal{K}_{AC} \not\models^{\mathcal{K}_A} \mathbb{P}\varphi$ and $\mathcal{K}_{AC} \not\models^{\mathcal{K}_C} \mathbb{P}\varphi$. Now, consider the formula $\mathbb{H}\varphi$ with $\varphi \triangleq \mathtt{AGF}\blacklozenge$. It is easy to see that $\mathcal{K}_C$ is the only strict superstructure in the filtering that satisfies φ. Thus, it holds that $\mathcal{K}_{AC} \models^{\mathcal{K}_C} \mathbb{H}\varphi$ but $\mathcal{K}_{AC} \not\models^{\mathcal{K}} \mathbb{H}\varphi$ and $\mathcal{K}_{AC} \not\models^{\mathcal{K}_A} \mathbb{H}\varphi$.

3.3. Basic Concepts

Given a KS $\mathcal{K}$ and an STL* formula φ, we say that $\mathcal{K}$ is a *model* of φ, in symbols $\mathcal{K} \models \varphi$, if and only if $\mathcal{K}$ is a $\mathcal{K}$-bounded model of φ, that is, $\mathcal{K} \models^{\mathcal{K}} \varphi$. In addition, for a given set of KSs $\aleph \subseteq \mathrm{KS(AP)}$, we say that φ is $\aleph$-*satisfiable* if there is a KS $\mathcal{K} \in \aleph$ such that $\mathcal{K} \models \varphi$.

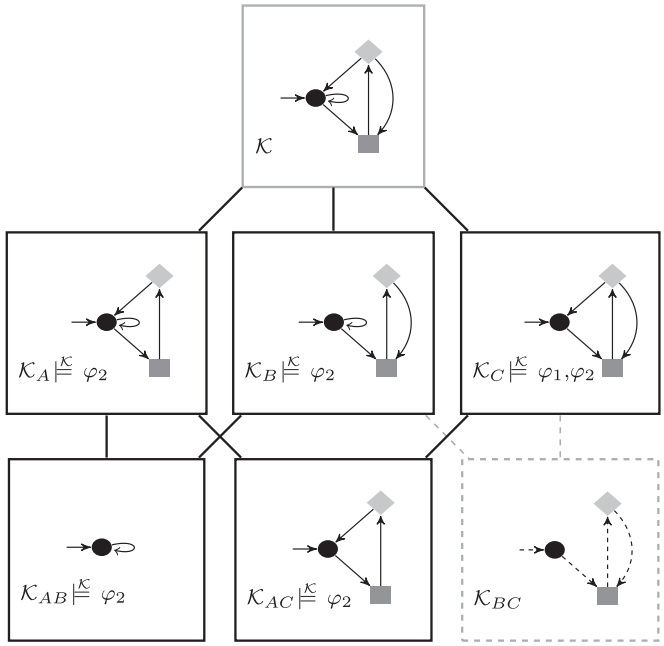

Fig. 7. $\mathbb{R}$ semantics.

A formula φ is an $\aleph$-*invariant* for two KSs $\mathcal{K}_1, \mathcal{K}_2 \in \aleph$ whenever $\mathcal{K}_1 \models^{\mathcal{K}^\star} \varphi$ if and only if $\mathcal{K}_2 \models^{\mathcal{K}^\star} \varphi$, for all $\mathcal{K}^\star \in \aleph$ with $\mathcal{K}_1, \mathcal{K}_2 \sqsubseteq \mathcal{K}^\star$. 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 $\mathcal{K}^\star, \mathcal{K} \in \aleph$ with $\mathcal{K} \sqsubseteq \mathcal{K}^\star$, it holds that if $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_1$, then $\mathcal{K} \models^{\mathcal{K}^\star} \varphi_2$; that is, 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 indicate which set of KSs $\aleph \subseteq$ KS(AP) the interpretation of formulas has to be restricted to. In particular, the corresponding *satisfiability problem* is to decide whether each state formula is $\aleph$-satisfiable or not. In addition, the *model-checking problem* asks whether $\mathcal{K} \models \varphi$, for a given KS $\mathcal{K} \in \aleph$ and state formula φ. Whenever $\aleph$ coincides with KS(AP) (KT(AP), respectively), we shall use the corresponding symbol KS (KT, respectively) instead. In the former case, we may also omit KS completely.

Given the similarity between the semilattice operators and the linear temporal operators, it is worth discussing what equivalences from classic LTL hold of the new ones. The first series of equivalences describes the fixpoint semantics of the reflexive operators:

—$\varphi_1 \overline{\mathbb{U}}_\phi \varphi_2 \equiv \varphi_2 \vee \varphi_1 \wedge \mathbb{X}_\phi \varphi_1 \overline{\mathbb{U}}_\phi \varphi_2$; —$\overline{\mathbb{F}}_\phi \varphi \equiv \varphi \vee \mathbb{X}_\phi \overline{\mathbb{F}}_\phi \varphi$;
—$\varphi_1 \overline{\mathbb{R}}_\phi \varphi_2 \equiv \varphi_2 \wedge (\varphi_1 \vee \mathbb{X}_\phi \varphi_1 \overline{\mathbb{R}}_\phi \varphi_2)$; —$\overline{\mathbb{G}}_\phi \varphi \equiv \varphi \wedge \mathbb{X}_\phi \overline{\mathbb{G}}_\phi \varphi$;
—$\varphi_1 \overline{\mathbb{S}}_\phi \varphi_2 \equiv \varphi_2 \vee \varphi_1 \wedge \mathbb{Y}_\phi \varphi_1 \overline{\mathbb{S}}_\phi \varphi_2$; —$\overline{\mathbb{P}}_\phi \varphi \equiv \varphi \vee \mathbb{Y}_\phi \overline{\mathbb{P}}_\phi \varphi$;
—$\varphi_1 \overline{\mathbb{B}}_\phi \varphi_2 \equiv \varphi_2 \wedge (\varphi_1 \vee \mathbb{Y}_\phi \varphi_1 \overline{\mathbb{B}}_\phi \varphi_2)$; —$\overline{\mathbb{H}}_\phi \varphi \equiv \varphi \wedge \mathbb{Y}_\phi \overline{\mathbb{H}}_\phi \varphi$.

The second series describes the relation between the strict operators and the immediate substructure operators:

—$\varphi_1 \mathbb{U}_\phi \varphi_2 \equiv \mathbb{X}_\phi \varphi_1 \overline{\mathbb{U}}_\phi \varphi_2$; —$\mathbb{F}_\phi \varphi \equiv \mathbb{X}_\phi \overline{\mathbb{F}}_\phi \varphi$;
—$\varphi_1 \mathbb{R}_\phi \varphi_2 \equiv \mathbb{X}_\phi \varphi_1 \overline{\mathbb{R}}_\phi \varphi_2$; —$\mathbb{G}_\phi \varphi \equiv \mathbb{X}_\phi \overline{\mathbb{G}}_\phi \varphi$;

—$\varphi_1 \mathbb{S}_\phi \varphi_2 \equiv \mathbb{EY}_\phi \varphi_1 \overline{\mathbb{S}}_\phi \varphi_2$; —$\mathbb{P}_\phi \varphi \equiv \mathbb{EY}_\phi \overline{\mathbb{P}}_\phi \varphi$;
—$\varphi_1 \mathbb{B}_\phi \varphi_2 \equiv \mathbb{AY}_\phi \varphi_1 \overline{\mathbb{B}}_\phi \varphi_2$; —$\mathbb{H}_\phi \varphi \equiv \mathbb{AY}_\phi \overline{\mathbb{H}}_\phi \varphi$.

Notice, however, that, due to the branching nature of the semilattice operators, the classic equivalences $\psi_1 \mathtt{R} \psi_2 \equiv \mathtt{G}\psi_2 \vee \psi_2 \mathtt{U}(\psi_1 \wedge \psi_2)$ and $\psi_1 \mathtt{B} \psi_2 \equiv \mathtt{H}\psi_2 \vee \psi_2 \mathtt{S}(\psi_1 \wedge \psi_2)$, linking together the LTL temporal operators R, G, U and B, H, S, respectively, do not lift to the corresponding semilattice operators. For instance, consider the formulas φ_1 and φ_2 of the example of Figure 6. 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, $\mathcal{K}_B$ does not satisfy φ_1; therefore, $\mathcal{K} \not\models \varphi_2 \mathbb{R} \varphi_1$. The same holds for the reflexive version of the two operators. Therefore, we have that, in general, $\varphi_1 \mathbb{R}_\phi \varphi_2 \not\equiv \mathbb{G}_\phi \varphi_2 \vee \varphi_2 \mathbb{U}_\phi(\varphi_1 \wedge \varphi_2)$ and $\varphi_1 \overline{\mathbb{R}}_\phi \varphi_2 \not\equiv \overline{\mathbb{G}}_\phi \varphi_2 \vee \varphi_2 \overline{\mathbb{U}}_\phi(\varphi_1 \wedge \varphi_2)$. Similarly, $\varphi_1 \mathbb{B}_\phi \varphi_2 \not\equiv \mathbb{H}_\phi \varphi_2 \vee \varphi_2 \mathbb{S}_\phi(\varphi_1 \wedge \varphi_2)$ and $\varphi_1 \overline{\mathbb{B}}_\phi \varphi_2 \not\equiv \overline{\mathbb{H}}_\phi \varphi_2 \vee \varphi_2 \overline{\mathbb{S}}_\phi(\varphi_1 \wedge \varphi_2)$. Indeed, consider the upper filtering in Figure 3 and the two formulas $\varphi_1 = \mathtt{EFG}\neg\bullet$ and $\varphi_2 = \mathtt{AF}\neg\bullet$. It is immediate to see that $\mathcal{K}_{AC} \models^{\mathcal{K}} \varphi_2 \mathbb{S}(\varphi_1 \wedge \varphi_2)$, since $\mathcal{K}_{AC} \models^{\mathcal{K}} \varphi_1 \wedge \varphi_2$. However, $\mathcal{K}_A \not\models^{\mathcal{K}} \varphi_1$, and hence, $\mathcal{K}_{AC} \not\models^{\mathcal{K}} \varphi_1 \mathbb{B} \varphi_2$.

Recall that when $\varphi_1 = \mathtt{f}$, the formulas $\varphi_1 \mathbb{R}_\phi \varphi_2$ and $\varphi_1 \mathbb{B}_\phi \varphi_2$ reduce to $\mathbb{AX}_\phi \varphi_2$ and $\mathbb{AY}_\phi \varphi_2$, while $\varphi_2 \mathbb{U}_\phi(\varphi_1 \wedge \varphi_2)$ and $\varphi_2 \mathbb{S}_\phi(\varphi_1 \wedge \varphi_2)$ reduce to $\mathbb{EX}_\phi \varphi_2$ and $\mathbb{EY}_\phi \varphi_2$. Hence, the nonequivalences stated previously follow immediately by the fact that $\mathbb{AX}_\phi \varphi \not\equiv \mathbb{EX}_\phi \varphi$ and $\mathbb{AY}_\phi \varphi \not\equiv \mathbb{EY}_\phi \varphi$.

Note that we sometimes consider formulas in *positive normal form* (pnf for short); that is, the negation is applied only to atomic propositions. In fact, to this aim, we have included in the syntax of STL* both Boolean connectives $\wedge$ and $\vee$; semilattice operators $\mathbb{U}$, $\mathbb{R}$, $\mathbb{S}$, and $\mathbb{B}$; path quantifiers E and A; and temporal operators U and R. Indeed, all formulas can be linearly translated in *pnf* by using generalized De Morgan's laws together with the following equivalence, which directly follows from the semantics of the logic: $\neg(\psi_1 \mathbb{U}_\phi \psi_2) \equiv (\neg\psi_1)\mathbb{R}_\phi(\neg\psi_2)$ and $\neg(\psi_1 \mathbb{S}_\phi \psi_2) \equiv (\neg\psi_1)\mathbb{B}_\phi(\neg\psi_2)$. Under this assumption, we sometimes consider $\neg\varphi$ as the *pnf* formula equivalent to the negation of φ.

4. REASONING ABOUT SUBSTRUCTURE PARTIAL ORDERS

As defined in the previous section, the semantics of STL* builds upon the classic semantics of temporal logic based on Kripke structures by adding the notion of filtering, namely, a partial order of substructures of a Kripke structure. In this section, we discuss some essential properties of the underlying partial order of substructures that can be expressed in the logic. Of particular interest are the notions of maximal and minimal elements of a filtering or of its subsets, which are considered in the first subsection. In the second one, instead, we show how STL* is able to express properties connected to the topological structure of its filterings.

Besides witnessing the expressive power of STL* in reasoning about the underlying semantic framework, the notions introduced in this section also have nice game-theoretic interpretations. In particular, minimality and maximality allow one to identify agent strategies in a game, while the notion of the neighborhood of a substructure allows one to express a notion of distance measure between strategies, which can be exploited to formalize finer notions of Nash equilibrium that do not seem to be expressible in other logics for games. A detailed discussion of all these connections to game theory is the focus of Section 5.

4.1. Minimality and Maximality

In the following, we will refer with $\mathcal{K}^\star$ to the bounding KS to which the evaluation process of lattice modalities is relative to.

The simplest concept that can be described in STL* is *absolute (downward) 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 $\{\mathcal{K}' \in \overline{\mathfrak{F}}_{\mathcal{K}}^{\downarrow\mathcal{K}^\star}(\phi) : \mathcal{K}' \models^{\mathcal{K}^\star} \varphi\}$; that is, $\mathcal{K}$ is the unique element of its downward filtering $\overline{\mathfrak{F}}_{\mathcal{K}}^{\downarrow\mathcal{K}^\star}(\phi)$ 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, $\mathcal{K} \models^{\mathcal{K}^\star} \mathtt{Min}_\phi(\varphi)$ if and only if $\mathcal{K}$ satisfies φ and none of its strict substructure does. So, $\mathcal{K}$ is minimal w.r.t. φ in the semilattice selected by ϕ.

The dual concept of *absolute (upward) maximality* requires $\mathcal{K}$ to be the maximal KS satisfying φ in the set $\{\mathcal{K}' \in \overline{\mathfrak{F}}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi) : \mathcal{K}' \models^{\mathcal{K}^\star} \varphi\}$. This concept can be expressed by the following construct:

$$\mathtt{Max}_\phi(\varphi) \triangleq \varphi \wedge \mathbb{H}_\phi \neg\varphi.$$

Then, $\mathcal{K} \models^{\mathcal{K}^\star} \mathtt{Max}_\phi(\varphi)$ if and only if $\mathcal{K}$ satisfies φ and none of its strict superstructure does. So, $\mathcal{K}$ is maximal w.r.t. φ in the lattice selected by ϕ.

For instance, considering the semilattice depicted in Figure 8v, it holds that $\mathcal{H}_{DE} \models^{\mathcal{H}^\star} \mathtt{Max}(\mathtt{AGF}\blacklozenge)$. Indeed, all the strict superstructures of $\mathcal{H}_{DE}$ in that semilattice, namely, $\mathcal{H}_D$, $\mathcal{H}_E$, and $\mathcal{H}$, contain a path where $\blacklozenge$ is never true. On the other hand, $\mathcal{H}_{ABC} \models^{\mathcal{H}^\star} \mathtt{Min}(\mathtt{EG}(\bullet \wedge \mathtt{EF}\blacksquare))$, since the only two strict substructures of $\mathcal{H}_{ABC}$, namely, $\mathcal{H}_{ABCD}$ and $\mathcal{H}_{ABCE}$, either do not admit a path where $\bullet$ always holds or do not contain paths reaching $\blacksquare$.

When the argument φ is $\mathtt{t}$, the resulting formulas $\bot_\phi \triangleq \mathtt{Min}_\phi(\mathtt{t}) \equiv \mathbb{G}\mathtt{f}$ and $\top_\phi \triangleq \mathtt{Max}_\phi(\mathtt{t}) \equiv \mathbb{H}\mathtt{f}$ precisely identify, respectively, the minimal elements in $\overline{\mathfrak{F}}_{\mathcal{K}}^{\downarrow\mathcal{K}^\star}(\phi)$ and the maximal element in $\overline{\mathfrak{F}}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi)$, for a given KS $\mathcal{K} \sqsubseteq \mathcal{K}^\star$. In other words, $\mathcal{K}' \models^{\mathcal{K}^\star} \bot_\phi$ ($\mathcal{K}' \models^{\mathcal{K}^\star} \top_\phi$, respectively) if and only if $\mathcal{K}'$ is a minimal (maximal, respectively) element in $\overline{\mathfrak{F}}_{\mathcal{K}}^{\downarrow\mathcal{K}^\star}(\phi)$ ($\overline{\mathfrak{F}}_{\mathcal{K}}^{\uparrow\mathcal{K}^\star}(\phi)$, respectively).

Note that if both φ and ϕ belong to any of the weak sublogics of STL*, $\mathtt{Min}_\phi(\varphi)$ and $\mathtt{Max}_\phi(\varphi)$ also do.

It is worth observing that, while both the notions of absolute downward minimality and absolute upward maximality make sense, the symmetric notions of absolute upward minimality and absolute downward maximality are trivial, as they clearly boil down to verifying the argument formula on the KS $\mathcal{K}$ itself.

By nesting the previous two constructs within lattice operators, we can also predicate over minimal (maximal, respectively) substructures and minimal (maximal, respectively) superstructures of a given KS. We call these properties *relative minimality* (*maximality*, respectively). Differently from their absolute versions, relative minimality (maximality, respectively) makes sense in both its upward and downward variations. In particular, given two formulas φ_1 and φ_2, we can assert the existence of a minimal substructure (superstructure, respectively) w.r.t. φ_1 that satisfies φ_2, or that all minimal substructures (superstructures, respectively) w.r.t. φ_1 have to satisfy φ_2. These concepts can be expressed by the following four constructs.

The existential downward operator $\mathtt{EMin}_\phi^{\downarrow}(\varphi_1, \varphi_2) \triangleq \mathbb{F}_\phi(\mathtt{Min}_\phi(\varphi_1) \wedge \varphi_2)$ is satisfied by a $\mathcal{K}^\star$-bounded model $\mathcal{K}$ if and only if there exists a substructure $\mathcal{K}'$ of $\mathcal{K}$, which is minimal w.r.t. φ_1 in the downward semilattice selected by ϕ, such that $\mathcal{K}' \models^{\mathcal{K}^\star} \varphi_2$. The corresponding universal version $\mathtt{AMin}_\phi^{\downarrow}(\varphi_1, \varphi_2) \triangleq \mathbb{G}_\phi(\mathtt{Min}_\phi(\varphi_1) \rightarrow \varphi_2)$ is satisfied by a

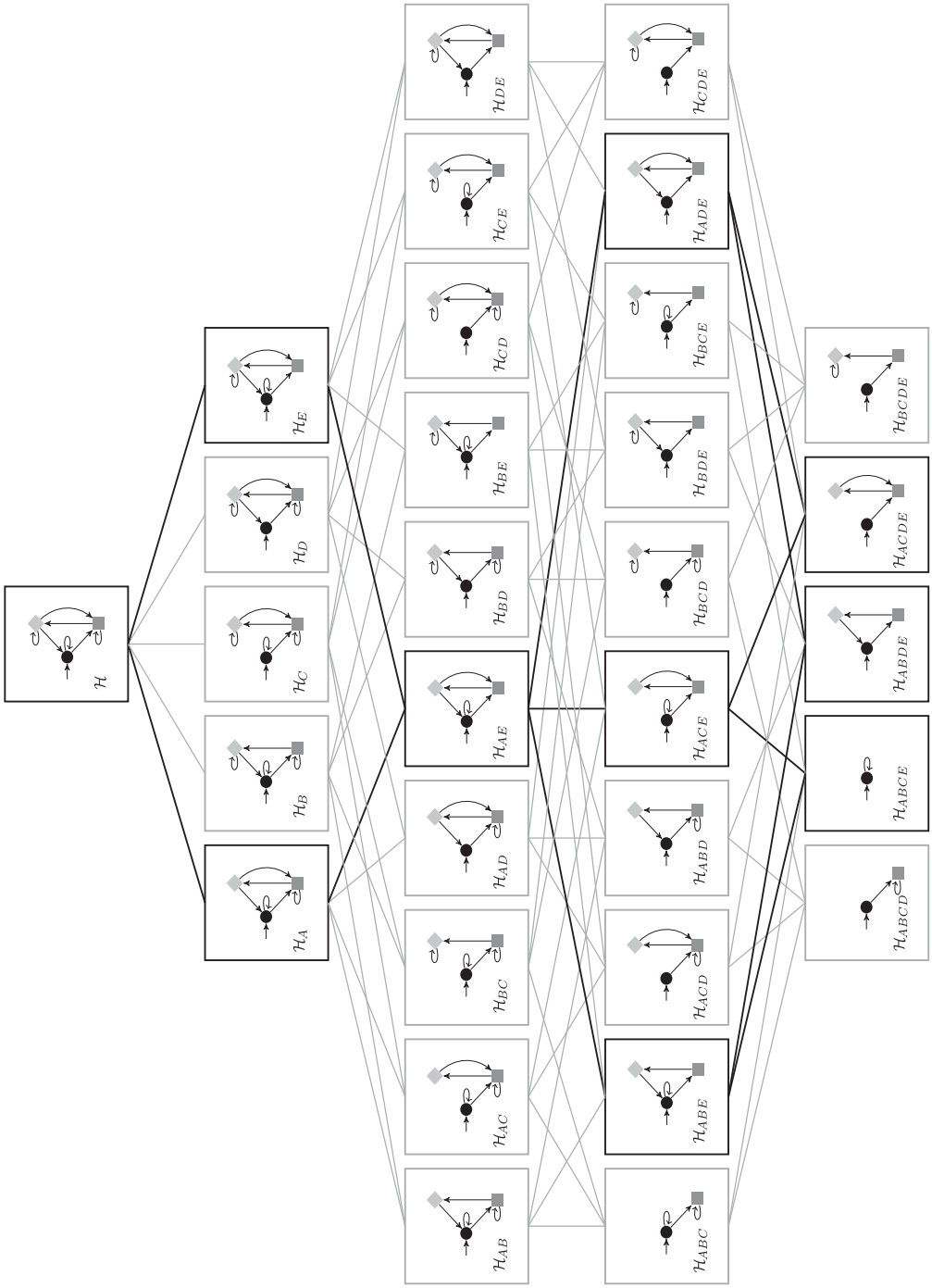

Fig. 8. The substructure semilattice of a KS $\mathcal{H}$ and its subsemilattice centered in the KS $\mathcal{H}_{AE}$.

$\mathcal{K}^\star$-bounded model $\mathcal{K}$ if and only if for all minimal substructures $\mathcal{K}'$ of $\mathcal{K}$ w.r.t. φ_1, it holds that $\mathcal{K}' \models^{\mathcal{K}^\star} \varphi_2$.

As an example, considering again Figure 8, it clearly holds that $\mathcal{H} \models^{\mathcal{H}} \mathtt{AMin}^{\downarrow}(\mathtt{EG}(\bullet \wedge \mathtt{EF}\blacksquare), \mathtt{E}(\mathtt{GF}\blacksquare \vee \mathtt{GF}\blacklozenge))$. Indeed, all four maximal substructures satisfying $\mathtt{EG}(\bullet \wedge \mathtt{EF}\blacksquare)$, namely, $\mathcal{H}_{ABC}$, $\mathcal{H}_{ABE}$, $\mathcal{H}_{ACE}$, and $\mathcal{H}_{BCE}$, have a path where at least one between $\blacksquare$ and $\blacklozenge$ holds infinitely often.

The existential upward operator $\mathtt{EMin}^{\uparrow}_{\phi}(\varphi_1, \varphi_2) \triangleq (\neg\varphi_1)\mathbb{S}_\phi(\varphi_1 \wedge \varphi_2)$ is satisfied by a $\mathcal{K}^\star$-bounded model $\mathcal{K}$ if and only if there exists a superstructure $\mathcal{K}'$ of $\mathcal{K}$, which is the minimal one in the upward lattice selected by ϕ satisfying both $\varphi_1 \wedge \varphi_2$, since it does not have any substructure satisfying φ_1. Similarly, the universal version $\mathtt{AMin}^{\uparrow}_{\phi}(\varphi_1, \varphi_2) \triangleq (\varphi_1)\mathbb{R}_\phi(\varphi_1 \rightarrow \varphi_2)$ satisfied by a $\mathcal{K}^\star$-bounded model $\mathcal{K}$ if and only if all superstructures $\mathcal{K}'$ of $\mathcal{K}$ satisfying φ_1 either satisfy φ_2 or have at least one substructure that satisfies φ_1.

Observe that the existential and universal version of the previous constructs are dual of one another, that is, $\neg\mathtt{EMin}^{\downarrow}_{\phi}(\varphi_1, \varphi_2) \equiv \mathtt{AMin}^{\downarrow}_{\phi}(\varphi_1, \neg\varphi_2)$ and $\neg\mathtt{EMin}^{\uparrow}_{\phi}(\varphi_1, \varphi_2) \equiv \mathtt{AMin}^{\uparrow}_{\phi}(\varphi_1, \neg\varphi_2)$. Once again, note that if φ_1, φ_2, and ϕ belong to any of the weak sublogics of STL*, the same holds of $\mathtt{EMin}^{\downarrow}_{\phi}(\varphi_1, \varphi_2)$ and $\mathtt{AMin}^{\downarrow}_{\phi}(\varphi_1, \varphi_2)$. On the contrary, the same does not hold of the corresponding upward versions, being explicitly defined in terms of $\mathbb{S}$ and $\mathbb{B}$.

The symmetric notions of relative downward (upward, respectively) maximality can be expressed by the following four constructs: $\mathtt{EMax}^{\downarrow}_{\phi}(\varphi_1, \varphi_2) \triangleq (\neg\varphi_1)\mathbb{U}_\phi(\varphi_1 \wedge \varphi_2)$, $\mathtt{AMax}^{\downarrow}_{\phi}(\varphi_1, \varphi_2) \triangleq (\varphi_1)\mathbb{R}_\phi(\varphi_1 \rightarrow \varphi_2)$, $\mathtt{EMax}^{\uparrow}_{\phi}(\varphi_1, \varphi_2) \triangleq \mathbb{P}_\phi(\mathtt{Max}_\phi(\varphi_1) \wedge \varphi_2)$, and $\mathtt{AMax}^{\uparrow}_{\phi}(\varphi_1, \varphi_2) \triangleq \mathbb{H}_\phi(\mathtt{Max}_\phi(\varphi_1) \rightarrow \varphi_2)$. Intuitively, $\mathtt{EMax}^{\downarrow}_{\phi}(\varphi_1, \varphi_2)$ ($\mathtt{AMax}^{\downarrow}_{\phi}(\varphi_1, \varphi_2)$, respectively) requires that some (every, respectively) strict substructure $\mathcal{K}'$ satisfying φ_1 also satisfies φ_2, whose maximality is ensured as none of its superstructures satisfies φ_1. On the other hand, $\mathtt{EMax}^{\uparrow}_{\phi}(\varphi_1, \varphi_2)$ ($\mathtt{AMax}^{\uparrow}_{\phi}(\varphi_1, \varphi_2)$, respectively) requires that some (every, respectively) strict superstructure $\mathcal{K}'$ maximal w.r.t. φ_1 also satisfies φ_2. Also in this case, duality laws hold, that is, $\neg\mathtt{EMax}^{\downarrow}_{\phi}(\varphi_1, \varphi_2) \equiv \mathtt{AMax}^{\downarrow}_{\phi}(\varphi_1, \neg\varphi_2)$ and $\neg\mathtt{EMax}^{\uparrow}_{\phi}(\varphi_1, \varphi_2) \equiv \mathtt{AMax}^{\uparrow}_{\phi}(\varphi_1, \neg\varphi_2)$. Differently from the case of relative minimality, the two upward versions of the relative maximality constructs belong to the weak sublogics of STL* whenever their arguments do, while the same does not hold of the corresponding downward versions.

For example, in Figure 8, $\mathcal{H}_{AE} \models^{\mathcal{H}} \mathtt{EMin}^{\uparrow}(\mathtt{EFG}\blacklozenge, \mathtt{EG}\bullet)$. This is witnessed by the superstructure $\mathcal{H}_E$, which is the smallest one satisfying both of the arguments. It also holds that $\mathcal{H}_{AE} \models^{\mathcal{H}} \mathtt{EMax}^{\uparrow}(\mathtt{EFG}\blacklozenge, \mathtt{EG}\bullet)$; however, in this case, the witness is the top structure $\mathcal{H}$ itself, which is the greatest one satisfying the arguments.

Observe that all the relative maximality and minimality constructs introduced previously quantify over strict substructures or superstructures. For each of those operators, a reflexive version can be defined by substituting the reflexive versions of the sublattice operators for their nonreflexive ones in the corresponding definitions. Following the notational convention we used in the article, the overlined operator names will denote the variant of the corresponding constructs. For instance, $\overline{\mathtt{EMin}}^{\downarrow}_{\phi}(\varphi_1, \varphi_2)$ corresponds to the reflexive version of $\mathtt{EMin}^{\downarrow}_{\phi}(\varphi_1, \varphi_2)$.

Finally, by means of a simple nesting of the weak upward operators, we are able to define a construct that precisely identifies the top of a lattice of superstructures determined by a parameter ϕ on which we can then verify a formula φ. Formally, we set $\uparrow_\phi(\varphi) \triangleq \overline{\mathbb{P}}_\phi(\mathbb{H}_\phi \mathtt{f} \wedge \varphi) \equiv \overline{\mathbb{H}}_\phi(\mathbb{H}_\phi \mathtt{f} \rightarrow \varphi)$. Therefore, regardless of the KS in which the formula $\uparrow_\phi(\varphi)$ is interpreted, its truth value depends on whether the argument φ is

Table II. Definition of the Neighborhood, Boundary, and Sphere Constructs

		Neighborhood: $\{\mathtt{E/A}\}\{\mathtt{N/F}\}\mathtt{Ngh}_\phi(\varphi_1, \varphi_2) \triangleq$
$\exists$	Nearest	$\neg\varphi_1 \overline{\mathbb{S}}_\phi(\neg\varphi_1 \wedge \varphi_2) \vee \neg\varphi_1 \overline{\mathbb{U}}_\phi(\neg\varphi_1 \wedge \varphi_2)$
	Farthest	$\neg\mathtt{Max}_\phi(\varphi_1) \overline{\mathbb{S}}_\phi(\neg\mathtt{Max}_\phi(\varphi_1) \wedge \varphi_2) \vee \neg\mathtt{Min}_\phi(\varphi_1) \overline{\mathbb{U}}_\phi(\neg\mathtt{Min}_\phi(\varphi_1) \wedge \varphi_2)$
$\forall$	Nearest	$\varphi_1 \overline{\mathbb{B}}_\phi(\neg\varphi_1 \rightarrow \varphi_2) \wedge \varphi_1 \overline{\mathbb{R}}_\phi(\neg\varphi_1 \rightarrow \varphi_2)$
	Farthest	$\mathtt{Max}_\phi(\varphi_1) \overline{\mathbb{B}}_\phi(\neg\mathtt{Max}_\phi(\varphi_1) \rightarrow \varphi_2) \wedge \mathtt{Min}_\phi(\varphi_1) \overline{\mathbb{R}}_\phi(\neg\mathtt{Min}_\phi(\varphi_1) \rightarrow \varphi_2)$

		Boundary: $\{\mathtt{E/A}\}\{\mathtt{N/F}\}\mathtt{Bnd}_\phi(\varphi_1, \varphi_2) \triangleq$
$\exists$	Nearest	$\mathtt{EMin}_\phi^{\uparrow}(\varphi_1, \varphi_2) \vee \mathtt{EMax}_\phi^{\downarrow}(\varphi_1, \varphi_2)$
	Farthest	$\mathtt{EMax}_\phi^{\uparrow}(\varphi_1, \varphi_2) \vee \mathtt{EMin}_\phi^{\downarrow}(\varphi_1, \varphi_2)$
$\forall$	Nearest	$\mathtt{AMin}_\phi^{\uparrow}(\varphi_1, \varphi_2) \wedge \mathtt{AMax}_\phi^{\downarrow}(\varphi_1, \varphi_2)$
	Farthest	$\mathtt{AMax}_\phi^{\uparrow}(\varphi_1, \varphi_2) \wedge \mathtt{AMin}_\phi^{\downarrow}(\varphi_1, \varphi_2)$

		Sphere: $\{\mathtt{E/A}\}\{\mathtt{N/F}\}\mathtt{Sph}_\phi(\varphi_1, \varphi_2) \triangleq$
$\exists$	Nearest	$\neg\varphi_1 \overline{\mathbb{S}}_\phi \varphi_2 \vee \neg\varphi_1 \overline{\mathbb{U}}_\phi \varphi_2$
	Farthest	$\neg\mathtt{Max}_\phi(\varphi_1) \overline{\mathbb{S}}_\phi \varphi_2 \vee \neg\mathtt{Min}_\phi(\varphi_1) \overline{\mathbb{U}}_\phi \varphi_2$
$\forall$	Nearest	$\varphi_1 \overline{\mathbb{B}}_\phi \varphi_2 \wedge \varphi_1 \overline{\mathbb{R}}_\phi \varphi_2$
	Farthest	$\mathtt{Max}_\phi(\varphi_1) \overline{\mathbb{B}}_\phi \varphi_2 \wedge \mathtt{Min}_\phi(\varphi_1) \overline{\mathbb{R}}_\phi \varphi_2$

true in the top structure of the upward filtering selected by ϕ (or in $\mathcal{K}^\star$ itself, when ϕ is $\mathtt{t}$).

4.2. A Topological Detour

The semilattice identified by the filtering of a given KS also can be interpreted as a topological space equipped with a metric, which is based on a variant of Hamming distance congruent with the substructure ordering. Its structures, indeed, represent the points of the space. The distance between two structures is either infinite, when they are not comparable, or the number of different edges in their transition relations, where edges whose presence only depends on other ones due to the reachability constraint are not considered. For example, consider the structure $\mathcal{H}$ and its downward filtering $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{H}}(\emptyset)$ depicted in Figure 8. Then, $\mathcal{H}_{AE}$ has distance 2 from both $\mathcal{H}$ and $\mathcal{H}_{ABCE}$. Indeed, $\mathcal{H}_{AE}$ lacks the two self-loops on the worlds ■ and ◆ w.r.t. $\mathcal{H}$, while it has the two edges rooted in the latter one, which instead are missing in $\mathcal{H}_{ABCE}$. Note that the arrows constituting the path from ● to ◆ are not counted in the computation of the measure, since they are simply required to connect the latter world with the origin of the structure in order to ensure the reachability constraint.

As usual in a metric space, thanks to the concepts of relative maximality and minimality and some related constructs, we can define the notions of neighborhood of KS and associated sphere. Since in STL we cannot numerically represent their radius, we implicitly identified the corresponding boundary by means of a formula. In particular, we are able to qualitatively determine the nearest and farthest of such boundaries. On that neighborhood, we can then check if another formula is verified on at least one or all its points. In Table II, we report the formal definition of the constructs that allow one to predicate, both in an existential and in a universal way, on the neighborhood, the boundary, or the entire sphere rooted in a given KS. Consider the formula $\mathtt{ENNgh}_\phi(\varphi_1, \varphi_2)$. It is true on a KS if and only if there is one of its strict superstructures or substructures that satisfies φ_2 and between this and the center of the neighborhood there is no structure that satisfies φ_1. Therefore, the corresponding boundary is the nearest one that satisfies the latter formula. Similarly, $\mathtt{ANNgh}_\phi(\varphi_1, \varphi_2)$ can be used to check that all the structures of such a neighborhood satisfy φ_2. In the case, instead, we are interested

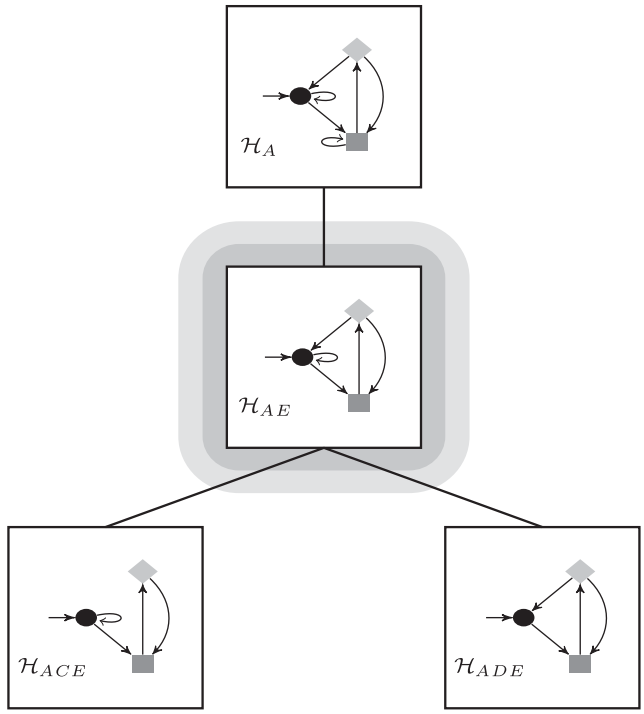

Fig. 9. Witness for $\mathcal{H}_{AE} \overset{\mathcal{H}}{\models} \texttt{ENSph}((\texttt{AGF}\neg\blacklozenge) \wedge (\texttt{EFG}\neg\bullet), \texttt{EF}(\blacklozenge \wedge \texttt{AX}\neg\bullet))$.

in the farthest neighborhood, we can use the formulas $\texttt{EFNgh}_\phi(\varphi_1, \varphi_2)$ and $\texttt{AFNgh}_\phi(\varphi_1, \varphi_2)$, which verify φ_2 on at least one or all the structures between the center and the farthest on which φ_1 is true. Naturally, for farthest, we mean the maximal structures in the upward lattice and the minimal ones in downward semilattice, both identified by the selector ϕ, that verify the required property.

To better understand the meaning of such constructs, consider again the KS $\mathcal{H}_{AE}$ and the three formulas $\varphi_1' \triangleq (\texttt{AGF}\neg\blacklozenge) \wedge (\texttt{EFG}\neg\bullet)$, $\varphi_1'' \triangleq \texttt{EGF}\bullet$, and $\varphi_2 \triangleq \texttt{EF}(\blacklozenge \wedge \texttt{AX}\neg\bullet)$. It is not hard to see that both $\mathcal{H}_{AE} \overset{\mathcal{H}}{\models} \texttt{ENSph}(\varphi_1', \varphi_2)$ and $\mathcal{H}_{AE} \overset{\mathcal{H}}{\models} \texttt{EFSph}(\varphi_1'', \varphi_2)$ hold, since $\mathcal{H}_{ACE} \overset{\mathcal{H}}{\models} \varphi_2$. In Figures 9 and 10, we respectively represent the witnesses for these two checks, where the dark gray region indicates the center of the sphere and the light gray one its interior.

More interestingly, universal quantification over the structures of the farther sphere allows STL* to express a "local" version of the notions of logical consequence, equivalence and invariance. We say that φ_2 is a *local logical consequence* of φ_1 w.r.t. a KS $\mathcal{K}$ and radius η, in symbols $\varphi_1 \Rightarrow_\eta \varphi_2$, if all the structures contained in the farthest sphere identified by η do satisfy $\varphi_1 \rightarrow \varphi_2$. When, in addition, $\varphi_2 \Rightarrow_\eta \varphi_1$ holds, then we say that φ_1 and φ_2 are *locally equivalent* w.r.t. $\mathcal{K}$ and radius η, in symbols $\varphi_2 \equiv_\eta \varphi_1$. Finally, we say that φ is a *local invariant* w.r.t. $\mathcal{K}$ and radius η if and only if φ holds in all structures contained in the sphere of maximal radius specified by η and centered in $\mathcal{K}$.

As an example, consider the farthest sphere of Figure 10 centered in $\mathcal{H}_{AE}$ and the formula $\texttt{AFSph}(\texttt{EGF}\bullet, \varphi_2)$, where $\varphi_2 = \texttt{AG}\bullet \vee \texttt{EGF}\blacklozenge$. This formula is satisfied by $\mathcal{H}_{AE}$, since every structure in the sphere either satisfies $\texttt{AG}\bullet$ (namely, $\mathcal{H}_{ABCE}$) or $\texttt{EGF}\blacklozenge$ (all the remaining structures). Then we can say that φ_2 is a local invariant w.r.t. $\mathcal{H}_{AE}$ and $\texttt{EGF}\bullet$.

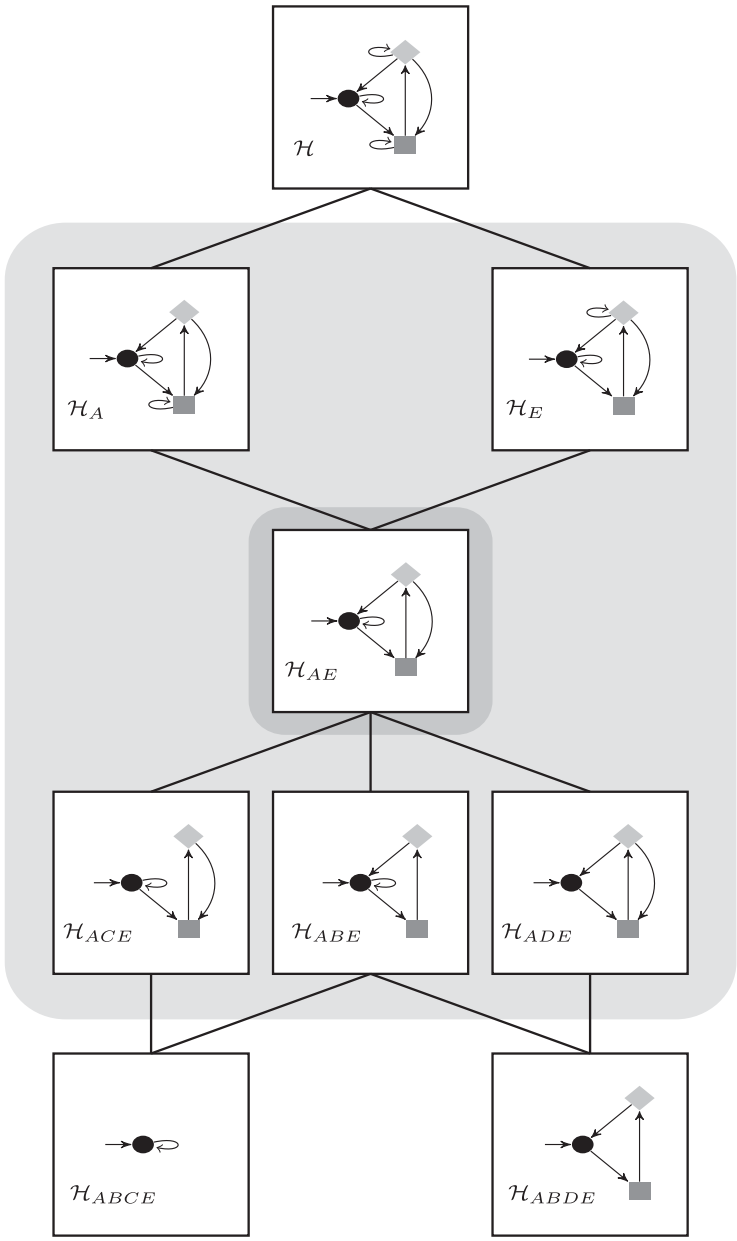

Fig. 10. Witness for $\mathcal{H}_{AE} \models^{\mathcal{H}}$ EFSph(EGF●, EF(◆ ∧ AX¬●)).

5. REASONING ABOUT GAMES

As argued in the previous section, a distinguishing feature of STL⋆ is the ability to quantify over substructures and express (relative) minimality and maximality properties. In the following, we show how these features allow us to naturally encode in the logic a number of relevant problems involving reasoning about games.

5.1. Module Checking

In open finite-state system model checking (module checking for short) [Kupferman et al. 2001], 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*, that is, a KS $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_I \rangle$, 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, that is, 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 (pruning these subtrees corresponds to disabling possible environment choices). 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 $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}^U}(1)$, which preserves all the system choices. Hence, the module-checking problem can be expressed in DWSTL* by checking whether $\mathcal{K}^U$ satisfies the formula $\varphi_{MC}(\varphi) \triangleq \overline{\mathbb{G}}_1(\varphi)$.

5.2. Turn-Based Games

The arena of a two-player turn-based game can be formalized by means of a KS $\mathcal{K}$ as earlier, 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 $\sigma_i : \mathrm{W}^*\mathrm{W}_i \to \mathrm{W}$ mapping sequences of worlds ending in a world 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, $\mathrm{Out}(\sigma_i) \triangleq \{\pi \in \mathrm{Pth}_{\mathcal{K}} : \forall j \in \mathbb{N}.(\pi)_j \in \mathrm{W}_i \to (\pi)_{j+1} = \sigma_i((\pi)_{\leq j})\}$. Intuitively, the outcomes of a strategy σ_i of player i are the plays of the game that agree with σ_i while leaving the other player to 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 DWSTL* logic, by nesting an existential and a universal relative minimality construct. Player 1 has a winning strategy for $\mathcal{K}$ if and only if $\mathcal{K}^U$ satisfies the formula $\varphi_{TG}(\psi) \triangleq \overline{\mathtt{EMin}}^{\downarrow}_{\neg 1}(\mathtt{t}, \overline{\mathtt{AMin}}^{\downarrow}_{1}(\mathtt{t}, \mathtt{A}\psi))$. The existential minimal operator $\overline{\mathtt{EMin}}^{\downarrow}_{\neg 1}$ selects a minimal substructure where all possible moves of Player 2 are preserved. Indeed, the selector $\neg 1$ allows only for substructures whose worlds not 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 $\overline{\mathtt{AMin}}^{\downarrow}_{1}$ selects all minimal substructures that 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.

Notice that, if one checks the formula $\varphi_{TG}(\psi)$ against the Kripke structure $\mathcal{K}$ itself, instead of its unwinding, the resulting problem coincides with checking for the existence of a *memoryless* winning strategy for the game. Indeed, in this case, the choices of the

agents allowed by the minimality operators at any state must always be the same in every occurrence of that state along a play, hence independent of the past history of the game.

5.3. 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 later. 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 $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_I \rangle$, 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 $\sigma_i : \mathrm{W}^+ \to \mathrm{Ac}_i$ 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 as those for the turn-based case. Now, to encode a quantification of a strategy by means of a suitable DSTL* 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 $\varphi_i \triangleq \mathtt{AG} \bigvee_{\mathtt{a}_i \in \mathrm{Ac}_i} \mathtt{AX} \bigvee_{\mathtt{a}_{3-i} \in \mathrm{Ac}_{3-i}} (\mathtt{a}_1, \mathtt{a}_2)$, with $i \in \{1, 2\}$, requires that, for every world w, there is an action of player i that allows it 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 construct. Finally, Player 1 has a winning strategy for $\mathcal{K}$ if and only if its unwinding $\mathcal{K}^U$ satisfies the formula $\varphi_{CG}(\psi) \triangleq \overline{\mathtt{EMax}}^{\downarrow}(\varphi_1, \overline{\mathtt{AMax}}^{\downarrow}(\varphi_2, \mathtt{A}\psi))$.

Similarly to the case of turn-based games, when the formula φ_{CG} is checked against the Kripke structure $\mathcal{K}$ itself, the resulting problem turns into the verification of the existence of a memoryless winning strategy for the concurrent game.

5.4. Reactive Synthesis

In the formulation proposed by Pnueli and Rosner [1989], 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 $\mathsf{P} : (2^{\mathrm{I}})^* \to 2^{\mathrm{O}}$ 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 previous problem will 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. Moreover, the worlds of the tree are in a one-to-one correspondence with the prefixes of all possible input sequences. In other words, $\mathcal{T}_{\mathsf{P}}$

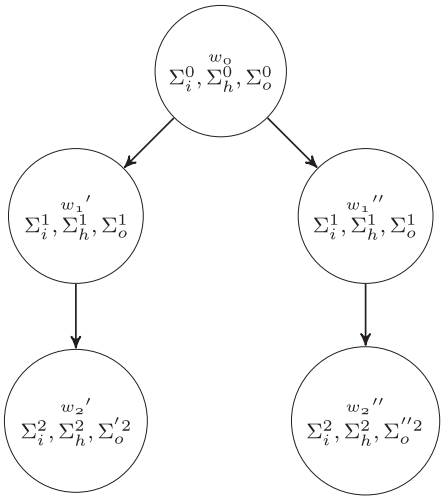

Fig. 11. Violation of w_0 successors' uniqueness.

represents all computations of the program P upon all possible inputs supplied by the environment.

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, and 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 nondeterministic (see Figure 11). 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 $v_1 = v_2$.

The two conditions can be expressed in DWSTL 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

$$\varphi_1 \triangleq \mathtt{AG} \bigwedge_{\Sigma_i \subseteq \mathrm{I}} \bigvee_{\Sigma_o \subseteq \mathrm{O}} \overline{\mathbb{G}}(\mathtt{AX}(\Sigma_i \wedge \overline{\mathrm{I} \setminus \Sigma_i}) \rightarrow \mathtt{AX}(\Sigma_o \wedge \overline{\mathrm{O} \setminus \Sigma_o}))$$

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

$$\varphi_2 \triangleq \texttt{Min}\left(\texttt{AG}\bigwedge_{\Sigma\subseteq \mathrm{I}\cup\mathrm{H}} \texttt{EX}(\Sigma \wedge \overline{(\mathrm{I}\cup\mathrm{H})\setminus\Sigma})\right).$$

The argument of 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 DWSTL* (or DWSTL) formula $\varphi_{RS}(\varphi) \triangleq \varphi \wedge \varphi_1 \wedge \varphi_2$. Indeed, $\varphi_{RS}(\varphi)$ is satisfied by a KT $\mathcal{T}$ if and only if it satisfies the original CTL* (or CTL) requirement φ together with the two formulas encoding the conditions earlier.

5.5. Nash Equilibria

We finally show how the power of reasoning about structures is useful to deal with general non-zero-sum games too. To do this, we provide an STL formalization of the classic concept of Nash equilibrium.

Informally, a Nash equilibrium [Osborne and Rubinstein 1994] is a solution concept involving two or more players, in which each of them is able to determine its own optimal strategy to achieve a certain goal and none of them is spurred to change it if all the other players do the same. In other words, it is rational for a player to adhere to the equilibrium assuming that all the other ones also do so. Obviously, to do this, each player needs to have the correct expectation about the behavior of all other players.

In the case of two-player deterministic games, where the players can exactly determine the payoff of a given play, the existence of a Nash equilibrium can be expressed in STL* by means of the formula

$$\varphi_{NE}(\psi_1,\psi_2) \triangleq \overline{\texttt{EMin}}_1^{\downarrow}(\texttt{t}, \overline{\texttt{EMin}}_{\neg 1}^{\downarrow}(\texttt{t}, \varphi_1 \wedge \varphi_{\neg 1})),\ \text{with}\ \varphi_i \triangleq \uparrow_{\neg i}(\overline{\texttt{EMin}}_{\neg i}^{\downarrow}(\texttt{t}, \texttt{A}\psi_i)) \rightarrow \texttt{A}\psi_i,$$

where φ_i represents the optimality check of player i w.r.t. to its LTL temporal goal ψ_i. With more details, similarly to the case of sum-zero games, the first two existential relative minimality constructs are used to identify the equilibrium by choosing for each of the players the corresponding deterministic strategy. Then, φ_i is used to verify that, if only player i changes its own strategy being able to satisfy its goal $\texttt{A}\psi_i$, it is already able to verify it by means of the strategy previously chosen.

When nondeterministic games are considered, the combination of the strategies of the players identifies, in general, a nondeterministic play (i.e., a computation tree) and not just a single path as in the case of deterministic games. Therefore, we generalize the player goals and express them by means of the CTL* formula ϕ_i, for player i. Strategies are now nondeterministic and the relative minimality operators are no longer adequate to extract such strategies. Instead, to correctly characterize a tree-like play winning the game, we need to replace the relative minimality constructs with the simple existential substructure operators in the previous schema, as reported in the following formulation:

$$\varphi_{NE}(\phi_1,\phi_2) \triangleq \overline{\mathbb{F}}_1\overline{\mathbb{F}}_{\neg 1}(\varphi_1 \wedge \varphi_{\neg 1}),\ \text{with}\ \varphi_i \triangleq \uparrow_{\neg i}(\overline{\mathbb{F}}_{\neg i}\phi_i) \rightarrow \phi_i.$$

In addition to the classic notions of equilibrium, STL* also allows one to formalize the concept of *local* Nash equilibrium [Alos-Ferrer and Ania 2001], where every agent checks for the optimality of its own choice in a neighborhood of the corresponding equilibrium strategy. Intuitively, this local form of equilibrium identifies local optima, where "small" changes to players' strategies do not allow them to improve them. By means of the topological constructs discussed in Section 4.2, we can first identify a neighborhood

with a formula η, determining its size. Then, we require that the equilibrium is reached within this neighborhood by checking optimality of the nondeterministic strategies via the following variation of the schema earlier:

$$\varphi_{NE}(\phi_1, \phi_2) \triangleq \overline{\mathbb{F}}_1\overline{\mathbb{F}}_{\neg 1}(\varphi_1 \wedge \varphi_{\neg 1}), \text{ with } \varphi_i \triangleq \mathtt{EFNgh}_{\neg_i}(\eta, \phi_i) \rightarrow \phi_i.$$

Notice that the formula φ_i essentially requires that, if there is a structure within the neighborhood identified by η that satisfies the ith goal ϕ_i, then this goal is already satisfied on the center of the neighborhood. In other words, if player i has a strategy close enough to the current one that satisfies ϕ_i, it is already able to satisfy this formula with the current strategy. If this is the case for all players, the corresponding strategies in the profile are indeed a local optimum.

Observe that the encodings of all versions of Nash equilibrium considered here make essential use of upward operators inside the definitions of the constructs $\uparrow(\cdot)$ and $\mathtt{EFNgh}(\cdot, \cdot)$, in order to retract players' decisions in the current strategy and consider alternative ones. As a consequence, none of them can be expressed in the downward fragment DSTL*.

The encodings for Nash equilibria we presented earlier only deal with win-loss objectives, represented by temporal formulas. We conclude this section by showing how that formalization can be extended in order to manage the more general case of games with a finite preference ordering among objectives. For simplicity, consider the case of nondeterministic strategies and suppose that each player i has a finite set of n_i goals $\{\phi_i^1, \ldots, \phi_i^{n_i}\}$, to which a preference ordering $\succcurlyeq_i$ on their indexes is associated. In other words, we are assuming that player i prefers $\phi_i^{j_1}$ to $\phi_i^{j_2}$ if and only if $j_1 \succcurlyeq_i j_2$. Then, the existence of a Nash equilibrium in this setting can be encoded by means of the following schema, where only the formula for the optimality check needs to be modified to account for the preference ordering among goals:

$$\varphi_{NE}(\phi_1, \phi_2) \triangleq \overline{\mathbb{F}}_1\overline{\mathbb{F}}_{\neg 1}(\varphi_1 \wedge \varphi_{\neg 1}), \text{ with } \varphi_i \triangleq \bigwedge_{j=1}^{n_i} \uparrow_{\neg i}(\overline{\mathbb{F}}_{\neg i}\phi_i^j) \rightarrow \bigvee_{j' \succcurlyeq_i j} \phi_i^{j'}.$$

Intuitively, the formula φ_i ensures that, if player i has a strategy to achieve a goal ϕ_i^j while the opponent stays with its own equilibrium strategy, then i can already achieve, by following the strategy previously chosen, a goal $\phi_i^{j'}$ that is at least as good as ϕ_i^j.

6. 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.

We first discuss the power of the logic in describing high-level properties of the underlying partial order of structures. In particular, we characterize the notions of density and discreteness of the partial order and show how they can be captured in STL*. The two notions have a nice and elegant characterization in graph-theoretic terms, by means of the concept of the minor of the underlying graph. Both these properties are tightly linked with the model-theoretic properties later discussed in Section 6.2 and can only be encoded in very expressive logics, such as MSOL [Rabin 1969] and the graded μCALCULUS [Kupferman et al. 2002; Bonatti et al. 2008].

6.1. Density and Discreteness

Let us consider the following DSTL formula: $\mathtt{Den}_\phi \triangleq \mathbb{F}_\phi \mathtt{t} \wedge \mathbb{X}_\phi \mathtt{f}$. Intuitively, it states that a given KS has at least one strict substructure in the semilattice selected by ϕ (this is required by $\mathbb{F}_\phi \mathtt{t}$), but none of them can be an immediate substructure (this

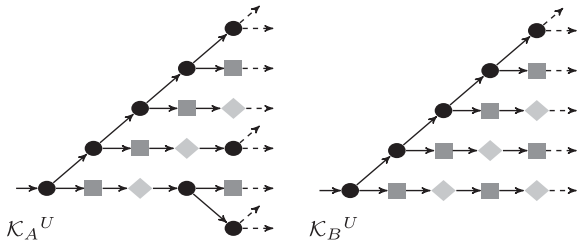

Fig. 12. The $\mathcal{K}_A$ and $\mathcal{K}_B$ unwindings $\mathcal{K}_A{}^U$ and $\mathcal{K}_B{}^U$ (we only report the labeling of the worlds instead of worlds themselves).

is required by $\mathbb{X}_\phi \mathtt{f}$), since no KS satisfies $\mathtt{f}$. More formally, $\mathcal{K} \models \mathtt{Den}_\phi$ if and only if (1) $\mathfrak{F}^{\downarrow}_{\mathcal{K}}(\phi) \neq \emptyset$ and (2), for all $\mathcal{K}' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\phi)$, there exists a $\mathcal{K}'' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\phi)$ such that $\mathcal{K}' \sqsubset \mathcal{K}'' \sqsubset \mathcal{K}$.

As an example, Figure 12 shows the unwinding $\mathcal{K}^U_A$ of the KS $\mathcal{K}_A$ in Figure 2, which does satisfy $\mathtt{Den}$. Indeed, $\mathfrak{F}^{\downarrow}_{\mathcal{K}^U_A}(\mathtt{f}) \neq \emptyset$, since $\mathcal{K}^U_{AB} \in \mathfrak{F}^{\downarrow}_{\mathcal{K}^U_A}(\mathtt{f})$; that is, the infinite path containing only $\bullet$ is one of the substructures of $\mathcal{K}^U_A$. Moreover, for each substructure $\mathcal{T} \in \mathfrak{F}^{\downarrow}_{\mathcal{K}^U_A}(\mathtt{f})$ and edge $(w, v) \in R_{\mathcal{K}^U_A} \setminus R_{\mathcal{T}}$ pruned in $\mathcal{T}$, we can always obtain a strict superstructure $\mathcal{T}' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}^U_A}(\mathtt{f})$ of $\mathcal{T}$, where some edge $(u, t) \in R_{\mathcal{K}^U_A} \setminus R_{\mathcal{T}'}$ from $u \in R^*_{\mathcal{K}^U_A}(v)$ 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 $\bullet$ with two outgoing edges. By iterating this argument, it is easy to see that any strict substructure $\mathcal{T}$ of $\mathcal{K}^U_A$ has an infinite chain of superstructures in the restricted filtering $\mathfrak{F}^{\downarrow}_{\mathcal{K}^U_A}(\mathtt{f})$. Consequently, we have that $|\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}^U_A}(\mathtt{f})| = \infty$.

The property we describe by means of the $\mathtt{Den}_\phi$ construct actually corresponds to a weak form of the density of an ordered set. Recall that a set S, ordered by a relation $\leq$, is dense in the classical sense if and only if, 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 $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\phi)$, but it always does when the greater component of the pair is fixed to $\mathcal{K}$. For instance, given two KSs $\mathcal{K}', \mathcal{K}'' \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\phi)$ minimal in this filtering, it holds that their join $\mathcal{K}' \sqcup \mathcal{K}''$ does not have any substructure $\mathcal{K}''' \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\phi)$ 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 $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\phi)$ 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 will not consider it in this article.

The density property expressed by the density construct can also be characterized in graph-theoretic terms. To this end, let us first introduce some preliminary notions.

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

Let $\mathcal{K}, \mathcal{K}' \in \mathrm{KS(AP)}$ be two Kripke structures. Then, $\mathcal{K}'$ is a *minor* of $\mathcal{K}$, in symbols $\mathcal{K}' \preccurlyeq \mathcal{K}$, if $\mathcal{K}'$ is isomorphic to the Kripke structure obtained by applying zero or more edge contractions to a substructure of $\mathcal{K}$, namely, by removing step by step an edge while simultaneously merging its incident worlds [Diestel 2012]. Formally, $\mathcal{K}' \preccurlyeq \mathcal{K}$ 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

$w_2 \in R_{\mathcal{K}'}(w_1)$ if and only if there is a track $\rho \in \mathrm{Trk}_{\mathcal{K}_{\mathsf{m}(w_1)}}$ for which (1) $\mathsf{lst}(\rho) = \mathsf{m}(w_2)$ and (2) $(\rho)_i \neq \mathsf{m}(w_3)$, 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. As an example, consider again the unwinding $\mathcal{K}_A^U$ of Figure 12. $\mathcal{K}_A^U$ has a binary KT $\mathcal{B}$ with $\mathrm{Dir} \triangleq \{a, b\}$ as a minor, that is, $\mathcal{B} \preccurlyeq \mathcal{K}_A^U$. This is witnessed by the following embedding m: (1) $\mathsf{m}(\epsilon) = \epsilon$; (2) for all $w \in \mathrm{Dir}^+$, 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 Kripke tree as the minor, since each world labeled by ● has a successor that 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.

We now have all we need to characterize the class of KSs satisfying the density constraint. The following theorem states that each world of a KS satisfying Den is the root of a tree substructure embedding a binary KT.

THEOREM 6.1 (DENSITY CHARACTERIZATION). *For each* KS $\mathcal{K} \in$ KS(AP)*, it holds that* $\mathcal{K} \models$ Den *if and only if* (1) $\mathcal{K}$ *is isomorphic to a* KT *and* (2) $\mathcal{K}_w$ *has a binary* KT *as a minor, for all* $w \in \mathrm{W}_{\mathcal{K}}$.

In order to prove the theorem, we first show that a KS $\mathcal{K}$ admits binary tree minors if and only if every world in $\mathcal{K}$ reaches a world with branching degree at least 2.

LEMMA 6.2 (BINARY MINOR CHARACTERIZATION). *For each* KS $\mathcal{K} \in$ KS(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* $u_1 \neq u_2$ *if and only if* $\mathcal{K}_w$ *has a binary* KT *as a minor, for all* $w \in \mathrm{W}_{\mathcal{K}}$.

PROOF. [*Only if*] Suppose that, for every world $w \in \mathrm{W}_{\mathcal{K}}$, there are a descendant $v \in R_{\mathcal{K}}^*(w)$ of w and two successors $u_1, u_2 \in R_{\mathcal{K}}(v)$ of v such that $u_1 \neq u_2$. Then, it is immediate to see that there exist two functions $\mathsf{F} : \mathrm{W}_{\mathcal{K}} \to \mathrm{W}_{\mathcal{K}}$ and $\mathsf{M} : \mathrm{W}_{\mathcal{K}} \times \mathrm{Dir} \to \mathrm{W}_{\mathcal{K}}$, with $\mathrm{Dir} \triangleq \{a, b\}$, such that (1) $\mathsf{F}(w) \in R_{\mathcal{K}}^*(w)$; (2) $\mathsf{M}(w, a), \mathsf{M}(w, b) \in R_{\mathcal{K}}(\mathsf{F}(w))$; and (3) $\mathsf{M}(w, a) \neq \mathsf{M}(w, b)$. In other words, $\mathsf{F}(w)$ chooses a descendant of w, while $\mathsf{M}(w, a)$ and $\mathsf{M}(w, b)$ choose two distinct successors of such descendant. Now, let $\mathcal{B}$ be a binary Kripke tree with direction set Dir. We want to show that, for all worlds $w \in \mathrm{W}_{\mathcal{K}}$, it holds that $\mathcal{B} \preccurlyeq \mathcal{K}_w$. To do this, consider the following injective embedding $\mathsf{m}_w : \mathrm{W}_{\mathcal{B}} \to \mathrm{W}_{\mathcal{K}_w}$: (1) $\mathsf{m}_w(\epsilon) \triangleq w$, (2) $\mathsf{m}_w(t \cdot a) \triangleq \mathsf{M}(\mathsf{m}_w(t), a)$, and (3) $\mathsf{m}_w(t \cdot b) \triangleq \mathsf{M}(\mathsf{m}_w(t), b)$. It is easy to see that m_w satisfies the defining constraints of the minor relation recalled earlier. Consequently, $\mathcal{B} \preccurlyeq \mathcal{K}_w$.

[*If*] Suppose that, for every world $w \in \mathrm{W}_{\mathcal{K}}$, it holds that $\mathcal{B} \preccurlyeq \mathcal{K}_w$, for some binary Kripke tree $\mathcal{B}$ having w.l.o.g. the set of directions $\mathrm{Dir} = \{a, b\}$. 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 $\mathsf{lst}(\rho_a) = \mathsf{m}_w(a)$ and $\mathsf{lst}(\rho_b) = \mathsf{m}_w(b)$ such that $(\rho_a)_{i_a} \neq \mathsf{m}_w(b)$ and $(\rho_b)_{i_b} \neq \mathsf{m}_w(a)$, for all $i_a \in [0, |\rho_a| - 1[$ and $i_b \in [0, |\rho_b| - 1[$. Now, let $j \in [0, \mathsf{min}\{|\rho_a|, |\rho_b|\} - 1[$ be the first index in which the two tracks diverge, that is, $(\rho_a)_{\leq j} = (\rho_b)_{\leq j}$ 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. □

We can finally prove Theorem 6.1.

PROOF OF THEOREM 6.1. [*If*] Suppose that $\mathcal{K}$ is isomorphic to a Kripke tree such that $\mathcal{K}_w$ has a binary Kripke tree $\mathcal{B}$ as a minor, where w.l.o.g. its set of directions

is $\mathrm{Dir} = \{a, b\}$, for all worlds $w \in \mathrm{W}_{\mathcal{K}}$. 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 $\mathsf{m}(\epsilon) = w$. As the first thing, it can immediately be seen that $\mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f}) \neq \emptyset$. Now, let $\mathcal{K}' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f})$ be a strict substructure of $\mathcal{K}$. Since $\mathcal{K}$ is isomorphic to a Kripke tree, there exists a world $v \in \mathrm{W}_{\mathcal{K}} \setminus \mathrm{W}_{\mathcal{K}'}$ that does not occur in $\mathcal{K}'$. Thus, also, its descendants are not present in this tree, that is, $R^*_{\mathcal{K}}(v) \cap \mathrm{W}_{\mathcal{K}'} = \emptyset$. Moreover, by definition of minor, we have that the two sets of descendants of $\mathsf{m}_v(a)$ and $\mathsf{m}_v(b)$ are disjoint and reachable from $\mathsf{m}_v(\epsilon)$, that is, $R^*_{\mathcal{K}}(\mathsf{m}_v(a)), R^*_{\mathcal{K}}(\mathsf{m}_v(b)) \subset R^*_{\mathcal{K}}(\mathsf{m}_v(\epsilon)) = R^*_{\mathcal{K}}(v)$ 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 $\mathcal{K}'' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f})$ 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 the first thing, $\mathcal{K}$ needs to be isomorphic to a Kripke tree. Indeed, assume the converse by contradiction. Thus, there exists a world $w \in \mathrm{W}_{\mathcal{K}}$ having two incoming 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 $\mathsf{lst}(\rho) = v_1$. Since both v_1 and v_2 are reachable from the initial world from which ρ starts, there is a world along ρ, possibly its origin, which the second track leading to v_2 forks out from. Therefore, 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 $\mathrm{W}' \triangleq \mathrm{W}_{\mathcal{K}} \setminus \{(\rho)_i : i \in]j, |\rho|[\}$. Clearly, $R' \triangleq (R_{\mathcal{K}} \cap (\mathrm{W}' \times \mathrm{W}')) \setminus \{(v_1, w)\}$ 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 Kripke structure that is also a strict substructure of $\mathcal{K}$ in $\mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f})$. However, $\mathcal{K}'$ does not have any strict superstructure $\mathcal{K}'' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f})$. 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 Kripke structure, we necessarily have that $\mathrm{W}_{\mathcal{K}''} \setminus \mathrm{W}_{\mathcal{K}'} = \{(\rho)_i : i \in]j, |\rho|[\}$. Otherwise, we have that $R_{\mathcal{K}''} \setminus R_{\mathcal{K}'} = \{(v_1, w)\}$. Hence, $\mathcal{K}'' = \mathcal{K}$, contradicting the fact that $\mathcal{K}'' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f})$.

Now, it remains to show that $\mathcal{B} \preccurlyeq \mathcal{K}_w$, for all $w \in \mathrm{W}_{\mathcal{K}}$, where $\mathcal{B}$ is a binary Kripke tree having as a set of directions $\mathrm{Dir} = \{a, b\}$. This fact follows directly from Lemma 6.2 after proving that, for all worlds $w \in \mathrm{W}_{\mathcal{K}}$, there are a descendant $v \in R^*_{\mathcal{K}}(w)$ and two of its successors $u_1, u_2 \in R_{\mathcal{K}}(v)$ 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 $\mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f}) \neq \emptyset$. 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 Kripke tree, we have 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, $\mathcal{K}' = \langle \mathrm{AP}, \mathrm{W}', R', \mathsf{L}_{\restriction \mathrm{W}'}, w_0\mathcal{K}\rangle$ is a Kripke tree that is also a strict substructure of $\mathcal{K}$ in $\mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f})$. However, $\mathcal{K}'$ does not have any strict superstructure $\mathcal{K}'' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f})$. 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 Kripke structure, we necessarily have $\mathrm{W}_{\mathcal{K}''} \setminus \mathrm{W}_{\mathcal{K}'} = R^*_{\mathcal{K}}((\rho)_{i+1})$. Hence, $\mathcal{K}'' = \mathcal{K}$, contradicting the fact that $\mathcal{K}'' \in \mathfrak{F}^{\downarrow}_{\mathcal{K}}(\mathtt{f})$. □

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 $\mathcal{K} \models \mathtt{Dis}_\phi$ if and only if, for all substructures $\mathcal{K}' \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\phi)$, it holds that either (1) $\mathcal{K}'$ does not admit any strict substructure in the filtering (i.e., it is minimal in $\overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\phi)$), or (2) no substructure $\mathcal{K}''' \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\phi)$ satisfies $\mathcal{K}'' \sqsubset \mathcal{K}''' \sqsubset \mathcal{K}'$ (i.e., there is an immediate strict substructure $\mathcal{K}'' \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\phi)$ of $\mathcal{K}'$). As an example, Figure 12 shows the unwinding $\mathcal{K}^U_B$ of the KS $\mathcal{K}_B$ in

Figure 2, which does satisfy Dis. Indeed, any substructure $\mathcal{T} \in \mathfrak{F}^{\downarrow}_{\mathcal{K}_B^U}(\mathtt{f})$ having at least a node $w \in \mathrm{W}_{\mathcal{T}}$ with $|R_{\mathcal{T}}(w)| = 2$ has an immediate strict substructure $\mathcal{T}' \in \mathfrak{F}^{\downarrow}_{\mathcal{T}}(\mathtt{f})$ such that, for all $u \in \mathrm{W}_{\mathcal{T}}$, it holds that $u \notin \mathrm{W}_{\mathcal{T}'}$ if and only if $u \in R^*_{\mathcal{T}}(v)$, 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 6.3 (DISCRETENESS CHARACTERIZATION). *For each* KS $\mathcal{K} \in$ KS(AP)*, it holds that* $\mathcal{K} \models$ Dis *if and only if* $\mathcal{K}$ *does not have a binary* KT *as a minor.*

PROOF. [*If*] Suppose that $\mathcal{K} \not\models$ Dis. Then, there exists a substructure $\mathcal{K}' \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(\emptyset)$ such that $\mathcal{K}' \models$ Den. Therefore, by Theorem 6.1, we have that $\mathcal{K}'$ has a binary Kripke tree $\mathcal{B}$ as a minor. Now, it is immediate to see that, given three Kripke structures $\mathcal{K}_1$, $\mathcal{K}_2$, 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 Kripke tree $\mathcal{B}$ having w.l.o.g. the set of directions $\mathrm{Dir} = \{a, b\}$. 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 $\mathsf{m}(\epsilon) = w_0\mathcal{K}$. In particular, for all worlds $t \in \mathrm{W}_{\mathcal{B}}$, 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 $j_t \in [0, \mathsf{min}\{|\rho^t_a|, |\rho^t_b|\} - 1[$ be the last index in which the two tracks diverge, that is, $(\rho^t_a)_{j_t} = (\rho^t_b)_{j_t}$ and $(\rho^t_a)_i \neq (\rho^t_b)_i$, for all $i \in]j_t, \mathsf{min}\{|\rho^t_a|, |\rho^t_b|\} - 1[$. In addition, assume $\mathrm{W}' \triangleq \bigcup_{t \in \mathrm{W}_{\mathcal{B}}} \{(\rho^t_a)_i : i \in [0, |\rho^t_a|[\} \cup \{(\rho^t_b)_i : i \in]j_t, |\rho^t_b|[\}$ as the set containing, for $t \in \mathrm{W}_{\mathcal{B}}$, all worlds of the first track ρ^t_a together with those in the suffix of ρ^t_b following the forking position j_t. 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, R' cannot contain distinct incoming edges for any world; that is, 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 Kripke tree $\mathcal{T}$ having $\mathcal{B}$ as a minor. At this point, by Theorem 6.1, in order to prove that $\mathcal{K}' \models$ Den and, consequently, that $\mathcal{K} \not\models$ 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: $\mathsf{m}_w(t) \triangleq \mathsf{m}(t_w \cdot t)$. The embedding m_w does satisfy the defining constraints of the minor relation, and therefore, $\mathcal{B} \preccurlyeq \mathcal{K}_{w'}$. □

Observe that there are KSs $\mathcal{K} \in$ KS(AP) such that neither $\mathcal{K} \models$ Den nor $\mathcal{K} \models$ Dis. In particular, any structure having a substructure satisfying Den and another one satisfying Dis does not satisfy either of them. An example is given by the KT whose root has $\mathcal{K}^U_A$ and $\mathcal{K}^U_B$ as the only children.

6.2. Expressiveness and Succinctness

Before proceeding to discuss further model-theoretic properties (i.e., expressiveness and succinctness of STL⋆, STL, and their fragments), we need to introduce a few additional definitions.

A logic $\mathfrak{L}$ enjoys the *tree* (*finite*, respectively) *model property* if every satisfiable formula $\varphi \in \mathfrak{L}$ has a KT $\mathcal{T}$(KS $\mathcal{K}$ with $|\mathrm{W}_{\mathcal{K}}| < \omega$, respectively) as a model. Moreover, $\mathfrak{L}$ is *invariant under bisimulation* if, for all pairs of bisimilar KSs $\mathcal{K}_1, \mathcal{K}_2 \in$ KS(AP), it holds that φ is an invariant for $\mathcal{K}_1$ and $\mathcal{K}_2$. Recall that two KSs $\mathcal{K}_1$ and $\mathcal{K}_2$ are bisimilar if there exists a relation between their worlds $B \subseteq \mathrm{W}_{\mathcal{K}_1} \times \mathrm{W}_{\mathcal{K}_2}$ that links together those worlds that behave in the same way; that is, $(w_1, w_2) \in B$ if and only if they agree on

the labeling, that is, $\mathsf{L}_{\mathcal{K}_1}(w_1) = \mathsf{L}_{\mathcal{K}_2}(w_2)$, and, for every successor w_1' of w_1, there is a successor w_2' of w_2 and, vice versa, for every successor w_2' of w_2, there is a successor w_1' of w_1, such that $(w_1', w_2') \in B$ [Sangiorgi 2009]. Finally, $\mathfrak{L}$ is *invariant under unwinding* if, for every KS $\mathcal{K} \in$ KS(AP), it holds that φ is an invariant for $\mathcal{K}$ and $\mathcal{K}^U$.

A first kind of comparison between two logics $\mathfrak{L}_1$ and $\mathfrak{L}_2$ can be done in terms of expressiveness w.r.t. a given class of KSs $\aleph \subseteq$ KS(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 $\mathfrak{L}_2 <_\aleph \mathfrak{L}_1$. If $\mathfrak{L}_1 \leq_\aleph \mathfrak{L}_2$ and $\mathfrak{L}_2 \leq_\aleph \mathfrak{L}_1$, then $\mathfrak{L}_1$ and $\mathfrak{L}_2$ are *expressively equivalent* w.r.t. $\aleph$, in symbols $\mathfrak{L}_2 \equiv_\aleph \mathfrak{L}_1$.

In addition, two logics can be compared in terms of their succinctness w.r.t. a blow-up function $\mathsf{f} : \mathbb{N} \to \mathbb{N}$. Note that such a comparison also makes sense when the logics are not expressively equivalent, by focusing on their common fragment, which may not be syntactically characterizable. Formally, we say that $\mathfrak{L}_1$ is *at most as succinct as* $\mathfrak{L}_2$ w.r.t. $\aleph$ with blow-up f, in symbols $\mathfrak{L}_1 \preccurlyeq^{\mathsf{f}}_\aleph \mathfrak{L}_2$, if, for every $\varphi_1 \in \mathfrak{L}_1$ having and $\aleph$-equivalent in $\mathfrak{L}_2$, there is a $\varphi_2 \in \mathfrak{L}_2$ with $\varphi_1 \equiv_\aleph \varphi_2$ and $|\varphi_2| \leq \mathsf{f}(|\varphi_1|)$.

By combining the previous relations, we can say that $\mathfrak{L}_1$ is *reducible to* (*strictly reducible to*, respectively) $\mathfrak{L}_2$ w.r.t. $\aleph$ with blow-up f, in symbols $\mathfrak{L}_1 \leq^{\mathsf{f}}_\aleph \mathfrak{L}_2$ ($\mathfrak{L}_1 <^{\mathsf{f}}_\aleph \mathfrak{L}_2$, respectively), if $\mathfrak{L}_1 \leq_\aleph \mathfrak{L}_2$ ($\mathfrak{L}_1 <_\aleph \mathfrak{L}_2$, respectively) and $\mathfrak{L}_1 \preccurlyeq^{\mathsf{f}}_\aleph \mathfrak{L}_2$.

Finally, we write $\mathfrak{L}_1 \preccurlyeq^{poly}_\aleph \mathfrak{L}_2$ ($\mathfrak{L}_1 \leq^{poly}_\aleph \mathfrak{L}_2$ and $\mathfrak{L}_1 <^{poly}_\aleph \mathfrak{L}_2$, respectively) if there are $k \in \mathbb{N}$ and $\mathsf{f} = \mathrm{O}(n^k)$ such that $\mathfrak{L}_1 \preccurlyeq^{\mathsf{f}}_\aleph \mathfrak{L}_2$ ($\mathfrak{L}_1 \leq^{\mathsf{f}}_\aleph \mathfrak{L}_2$ and $\mathfrak{L}_1 <^{\mathsf{f}}_\aleph \mathfrak{L}_2$, respectively), where n is the length of the formula in $\mathfrak{L}_1$. Also, we write *lin* instead of *poly* if the previous relation holds for $k = 1$.

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 DWSTL[KS].

THEOREM 6.4 (DWSTL[KS] NEGATIVE PROPERTIES). DWSTL[KS] *satisfies the following:* (1) *it does not enjoy the tree model property,* (2) *it is not invariant under unwinding, and* (3) *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 (1) follows. Items (2) and (3) are immediate consequences.

PROOF. [*Item 1*] Consider the DWSTL formula $\varphi = \mathtt{EMin}^{\downarrow}(\varphi_1, \varphi_2)$, with $\varphi_1 \triangleq \mathtt{EX}(\bullet \wedge \varphi_2)$ and $\varphi_2 \triangleq \mathtt{EX}\blacksquare$. 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 DWSTL[KS] does not enjoy the tree model property.

[*Items 2 and 3*] 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 DWSTL[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 DWSTL[KS] is not invariant under bisimulation. □

Recall that CTL enjoys the tree model property and that it is clearly a syntactic fragment of DWSTL. Thus, the following result is an immediate consequence of Item (1) of the previous theorem.

COROLLARY 6.5 (DWSTL[KS] EXPRESSIVENESS). CTL $<^{lin}_{\mathrm{KS}}$ DWSTL.

A deeper result about the impossibility of a finitary representation of some DSTL[KS] models directly follows from the density characterization of Theorem 6.1.

THEOREM 6.6 (DSTL[KS] NEGATIVE PROPERTY). DSTL[KS] *does not enjoy the finite model property.*

PROOF. Consider the density construct Den. By Theorem 6.1, 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. □

It is known that *Counting* CTL* (CTL*+C for short) [Moller and Rabinovich 2003] has the finite model property. We recall that this logic is obtained by adding to CTL* the successor counting operator $\mathtt{E}^{\geq g}\mathtt{X}\varphi$, 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 6.7 (DENSITY ON KSS). Den *is not* KS*-equivalent to any* CTL*+C *formula.*

Differently from the KS case, the density construct is easily expressible in CTL+C interpreted over KTs. Indeed, CTL+C formula $\mathtt{AGEFE}^{\geq 2}\mathtt{Xt}$ expresses that, from every world of a KT, some world with at least two successors is eventually reached. Clearly, any such KT embeds a binary KT. The result is formalized by the following theorem.

THEOREM 6.8 (DENSITY ON KTS). $\mathtt{Den} \equiv_{\mathrm{KT}} \mathtt{AGEFE}^{\geq 2}\mathtt{Xt}$.

PROOF [$\mathtt{Den} \Rightarrow_{\mathrm{KT}} \mathtt{AGEFE}^{\geq 2}\mathtt{Xt}$]. Consider a KT $\mathcal{T}$ such that $\mathcal{T} \models \mathtt{Den}$. Then, by Theorem 6.1, for every $w \in \mathrm{W}_{\mathcal{T}}$, it holds that $\mathcal{T}_w$ has a binary KT as a minor. Now, by Lemma 6.2, 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 $\mathcal{T}_v \models \mathtt{E}^{\geq 2}\mathtt{Xt}$, which implies $\mathcal{T}_w \models \mathtt{EFE}^{\geq 2}\mathtt{Xt}$. Hence, $\mathcal{T} \models \mathtt{AGEFE}^{\geq 2}\mathtt{Xt}$.

[$\mathtt{AGEFE}^{\geq 2}\mathtt{Xt} \Rightarrow_{\mathrm{KT}} \mathtt{Den}$] Consider a KT $\mathcal{T}$ such that $\mathcal{T} \models \mathtt{AGEFE}^{\geq 2}\mathtt{Xt}$. This means that, for all $w \in \mathrm{W}_{\mathcal{T}}$, there is $v \in R^*_{\mathcal{T}}(w)$ such that $\mathcal{T}_v \models \mathtt{E}^{\geq 2}\mathtt{Xt}$. Consequently, v must have two distinct successors $u_1, u_2 \in R_{\mathcal{T}}(v)$. Hence, by Lemma 6.2, $\mathcal{K}_w$ has a binary KT as a minor, for all $w \in \mathrm{W}_{\mathcal{K}}$. From Theorem 6.1, the conclusion immediately follows. □

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) [Hafer and Thomas 1987]. However, this result is far beyond the scope of this article.

When interpreted on KTs, invariance under unwinding and the tree model property of STL* hold trivially. However, by observing that a KT with a single path is bisimilar to a KT with two paths, assuming the worlds in the two KTs are equally labeled, and that the first one is minimal and the second one is not, we immediately obtain the following result for the weakest fragment DWSTL.

THEOREM 6.9 (DWSTL[KT] NEGATIVE PROPERTY). DWSTL[KT] *is not invariant under bisimulation.*

Since CTL is known to be invariant under bisimulation, the strict inclusion of CTL into DWSTL immediately follows from the previous theorem. A similar result can be stated w.r.t. LTL. Indeed, the DWSTL formula $\overline{\mathtt{AMin}}^{\downarrow}(\mathtt{t}, \varphi)$ verifies φ on every path of the underlying KT. Therefore, for every LTL formula ψ, it holds that $\mathtt{A}\psi \equiv_{\mathrm{KT}} \overline{\mathtt{AMin}}^{\downarrow}(\mathtt{t}, \varphi)$, where the CTL state formula φ is obtained from ψ by coupling each temporal operator

occurring in it with some path quantifier. As a consequence, we obtain the following theorem.

THEOREM 6.10 (DWSTL[KT] EXPRESSIVENESS). DWSTL[KT] *satisfies the following:* (1) CTL $<^{lin}_{\mathrm{KT}}$ DWSTL *and* (2) LTL $<^{lin}_{\mathrm{KT}}$ DWSTL.

Besides an increase in expressive power, the addition of the lattice operators to CTL also allows for an exponential increase in succinctness, as stated next.

THEOREM 6.11 (DWSTL[KT] SUCCINCTNESS). DWSTL$\not\leq^{poly}_{\mathrm{KT}}$ CTL.

PROOF. Let $\mathfrak{L} \subseteq \mathrm{CTL}^+$ be the subset of state formulas in which the nesting of path quantifiers is only allowed inside a CTL construct $\mathtt{AG}$; that is, we can only write path quantifications of the form $\mathtt{E}\psi, \mathtt{A}\psi, \mathtt{AGE}\psi, \mathtt{AGA}\psi \in \mathfrak{L}$, where the matrix ψ is an LTL formula without nesting of temporal operators. It is easy to see that the satisfiability problem for CTL^+ can be linearly reduced to the same problem for $\mathfrak{L}$. Indeed, consider a CTL^+ formula φ having $\{\mathtt{E}\psi_i^{\mathtt{E}}, \mathtt{A}\psi_i^{\mathtt{A}}\}$ as its set of subformulas. Then, by using the fresh atomic propositions $p_i^{\mathtt{E}}$ and $p_i^{\mathtt{A}}$, we can construct an equisatisfiable $\mathfrak{L}$ formula $\varphi' \wedge \bigwedge_i \mathtt{AGE}(p_i^{\mathtt{E}} \leftrightarrow \psi_i^{\mathtt{E}'}) \wedge \bigwedge_i \mathtt{AGA}(p_i^{\mathtt{A}} \leftrightarrow \psi_i^{\mathtt{A}'})$, where $\psi_i^{\mathtt{E}'}$, $\psi_i^{\mathtt{A}'}$, and φ' are obtained from $\psi_i^{\mathtt{E}}$, $\psi_i^{\mathtt{A}}$, and φ, respectively, by replacing each occurrence of a path quantification $\mathtt{E}\psi_i^{\mathtt{E}}$ or $\mathtt{A}\psi_i^{\mathtt{A}}$ with the corresponding proposition $p_i^{\mathtt{E}}$ or $p_i^{\mathtt{A}}$ [Emerson and Sistla 1983]. Therefore, $\mathfrak{L}$ has a 2$2^{\mathrm{EXPTIME}}$-COMPLETE satisfiability problem, as CTL^+ does. At this point, to prove the statement, it suffices to show that $\mathfrak{L} \leq^{lin}_{\mathrm{KT}}$ DWSTL. Indeed, suppose by contradiction that DWSTL $\not\leq^{poly}_{\mathrm{KT}}$ CTL. Then, LTL $\not\leq^{poly}_{\mathrm{KT}}$ CTL. Now, since it is known that $\mathfrak{L} \leq_{\mathrm{KS}} \mathrm{CTL}^+ \leq_{\mathrm{KS}}$ CTL, we immediately derive that $\mathfrak{L} \leq^{poly}_{\mathrm{KT}}$ CTL. Consequently, CTL should have a 2$2^{\mathrm{EXPTIME}}$-HARD satisfiability problem, which is in contradiction with the know EXPTIME upper bound. The reduction from $\mathfrak{L}$ to DWSTL simply extends the one described in Item (2) of Theorem 6.10, by using the following two equivalences: $\mathtt{E}\psi \equiv_{\mathrm{KT}} \overline{\mathtt{EMin}}^{\downarrow}(\mathtt{t}, \varphi)$ and $\mathtt{A}\psi \equiv_{\mathrm{KT}} \overline{\mathtt{AMin}}^{\downarrow}(\mathtt{t}, \varphi)$, where φ is obtained from ψ by coupling each temporal operator with a path quantifier. □

These results show that DWSTL encompasses both LTL and CTL. Any uniform translation from CTL* into either DWSTL or DSTL does not seem to exist, however. Interestingly enough, such a translation becomes immediate if the weakest upward operators $\mathbb{H}$ and $\mathbb{P}$ are allowed, since through them we can define the construct $\uparrow(\cdot)$ that allows for reasoning about the original structure once some pruning is already done. The following result shows that CTL* can indeed be linearly translated into WSTL.

THEOREM 6.12 (WSTL[KT] EXPRESSIVENESS). CTL* $<^{lin}_{\mathrm{KT}}$ WSTL.

PROOF. The proof generalizes the technique used earlier to encode LTL into DWSTL. Similarly to the LTL case, checking temporal subformulas over individual paths of a tree can be simulated by restricting, using relative minimal operators, the verification to minimal subtrees, each of which only contains a single path. However, as opposed to LTL, the path quantifiers in CTL* can select paths of the original tree, which may not be present in the current subtree. In WSTL, we can recover those paths by moving up to the original uppermost tree by means of the construct $\uparrow(\cdot)$ and, depending on the required path quantification in the formula, combine it with the corresponding universal or existential relative minimal operator, which quantifies over individual paths.

For each CTL* formula φ, we construct the equivalent WSTL formula $\widetilde{\varphi}$ by recursively replacing in φ all occurrences of a path quantifier $\mathtt{E}\psi$ or $\mathtt{A}\psi$ with the formula $\uparrow(\overline{\mathtt{EMin}}^{\downarrow}(\mathtt{t}, \widehat{\psi}))$ or $\uparrow(\overline{\mathtt{AMin}}^{\downarrow}(\mathtt{t}, \widehat{\psi}))$, respectively, where the internal part $\widehat{\psi}$ is obtained from ψ by

coupling a path quantifier (either A or E) to each temporal operator occurring in it. For example, the formula $\mathtt{EGFAFG}p$ can be transformed into $\uparrow(\overline{\mathtt{EMin}}^{\downarrow}(\mathtt{t}, \mathtt{EGEF}\uparrow(\overline{\mathtt{AMin}}^{\downarrow}(\mathtt{t}, \mathtt{EFEG}p))))$. Note that we can avoid the outermost construct $\uparrow(\cdot)$, since the current structure equals the bounding one in the evaluation of the external part $\overline{\mathtt{EMin}}^{\downarrow}(\mathtt{t}, \ldots)$.

The translation described previously is clearly linear in the size of the original CTL* formula. In addition, since CTL* is known to be invariant under bisimulation, by Theorem 6.9, we obtain the thesis. □

Finally, by adapting the classic (linear) reduction proposed in Hafer and Thomas [1987], showing that CTL* $\leq^{lin}_{\mathrm{KT}}$ MPL, we can prove that STL* can only express regular languages over trees, namely, the class of languages expressible in MSOL.

THEOREM 6.13 (STL*[KT] REGULARITY). STL* $\leq^{lin}_{\mathrm{KT}}$ MSOL.

PROOF. To prove the statement, it suffices to provide a linear translation from STL* to MSOL. This can be done by means of the two functions $\Gamma_s : \mathrm{STL}^\star \times \mathrm{SVr} \times \mathrm{SVr} \times \mathrm{FVr} \rightarrow \mathrm{MSOL}$ and $\Gamma_p : \mathrm{LTL}(\mathrm{STL}^\star) \times \mathrm{SVr} \times \mathrm{SVr} \times \mathrm{SVr} \times \mathrm{FVr} \rightarrow \mathrm{MSOL}$ defined later, where SVr and FVr are the sets of second- and first-order variables and LTL(STL*) denotes the set of path formulas of STL*. The function Γ_s takes care of formalizing the semantics of STL* state formulas into MSOL. Similarly, Γ_p works on path formulas. The second and third arguments of each function respectively represent the topmost tree and the current subtree w.r.t. which the formula is interpreted. The fourth argument of Γ_p encodes the path selected in the current tree, while the last argument in both functions corresponds to the current world. The translation essentially extends to STL* formulas the classic reduction from CTL* to MPL [Hafer and Thomas 1987].

The semantics of semilattice and temporal operators of STL* over trees requires one to quantify over paths of a tree and over tree substructures and tree superstructures. In order to express such quantifications, we need to introduce some simple constructs, which allow us to properly constrain the domain of the second-order quantifications of MSOL to paths and tree substructures/superstructures. The operator $\leq$ represents the reflexive version of the descendant relation, while $\lessdot$ represents the immediate successor; $\mathtt{Path}(T, P, x)$ holds when P is an infinite path of the tree T starting in x; $\mathtt{SubTree}^{\sqsubseteq}_{\phi}(T, T', x)$ encodes the nonstrict subtree relation between T', a tree rooted in x, and T, where worlds satisfying ϕ have the same outgoing edges as in T; similarly, $\mathtt{SubTree}^{\sqsubset}_{\phi}(T, T', x)$ encodes the strict subtree relation; and, finally, $\mathtt{SupTree}_{\phi}(T^\star, T, T', x)$ expresses that tree T' rooted in x is a superstructure of tree T bounded by $T^\star$, preserving all the edges of worlds satisfying ϕ. Formally:

—$x \leq y \triangleq (x < y) \vee (x = y)$.
—$x \lessdot y \triangleq (x < y) \wedge (\neg \exists z \,.\, x < z \wedge z < y)$.
—$\mathtt{Path}(T, P, x)$ is the conjunction of the following formulas:
  —$x \in P$ (the node x belongs to the tree P);
  —$\forall y \in P \,.\, x \leq y \wedge y \in T$ (all nodes in the tree P are descendants of x and belong to the tree T);
  —$\forall y \in P \,.\, \exists z \in P \,.\, y \lessdot z$ (the tree P is infinite, i.e., each of its node has a successor);
  —$\forall y \in P \,.\, \forall z \in P \,.\, y \leq z \vee z \leq y$ (the tree P is a sequence, i.e., all its nodes are totally ordered).
—$\mathtt{SubTree}^{\sqsubseteq}_{\phi}(T, T', x)$ is the conjunction of the following formulas:
  —$x \in T'$ (the node x belongs to the tree T');
  —$\forall y \in T' \,.\, x \leq y \wedge y \in T$ (all nodes in T' are descendants of x and belong to T);
  —$\forall y \in T' \,.\, \exists z \in T' \,.\, y \lessdot z$ (T' is infinite, i.e., each node has a descendant);

—$\forall y \in T' \,.\, \phi(y) \rightarrow \forall z \in T \,.\, y \lessdot z \rightarrow z \in T'$ (the edges of the nodes satisfying $\phi(y)$ are the same as in T).
—$\mathtt{SubTree}^{\sqsubset}_{\phi}(T, T', x) \triangleq \mathtt{SubTree}^{\sqsubseteq}_{\phi}(T, T', x) \wedge \exists y \in T \,.\, x < y \wedge \neg y \in T'$;
—$\mathtt{SupTree}_{\phi}(T^\star, T, T', x) \triangleq \mathtt{SubTree}^{\sqsubseteq}_{\mathtt{f}}(T^\star, T', x) \wedge \mathtt{SubTree}^{\sqsubset}_{\phi}(T', T, x)$.

Note that $\mathtt{SubTree}^{\sqsubseteq}_{\phi}(T, T', x)$ and $\mathtt{SubTree}^{\sqsubset}_{\phi}(T, T', x)$ hold if and only if $T' \in \overline{\mathfrak{F}}^{\downarrow}_{T}(\phi)$ and $T' \in \mathfrak{F}^{\downarrow}_{T}(\phi)$, respectively. Similarly, for any tree superstructure $T^\star$ of T, we have that $\mathtt{SupTree}_{\phi}(T^\star, T, T', x)$ holds if and only if $T' \in \mathfrak{F}^{\uparrow^{T^\star}}_{T}(\phi)$. With these constructs, we can now provide a suitable encoding of the semantic conditions of the semilattice operators, according to Definition 3.2.

The following first two groups of encodings are obvious and deal with atomic propositions and Boolean connectives:

(1) $\Gamma_s(p, T^\star, T, x) \triangleq p(x)$.

(2) —$\Gamma_s(\neg\varphi, T^\star, T, x) \triangleq \neg\Gamma_s(\varphi, T^\star, T, x)$;
—$\Gamma_s(\varphi_1 \wedge \varphi_2, T^\star, T, x) \triangleq \Gamma_s(\varphi_1, T^\star, T, x) \wedge \Gamma_s(\varphi_2, T^\star, T, x)$;
—$\Gamma_s(\varphi_1 \vee \varphi_2, T^\star, T, x) \triangleq \Gamma_s(\varphi_1, T^\star, T, x) \vee \Gamma_s(\varphi_2, T^\star, T, x)$.

The third group deals with the semilattice operators, essentially translating the semantic conditions for the semilattice operators, where the domain of quantification for the second-order variables is suitably constrained by means of the corresponding construct introduced earlier:

(3) —$\Gamma_s(\varphi_1 \mathbb{U}_\phi \varphi_2, T^\star, T, x) \triangleq \exists T'.\mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T, T', x) \wedge \Gamma_s(\varphi_2, T^\star, T', x) \wedge \forall T''.\mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T, T'', x) \wedge \mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T'', T', x) \rightarrow \Gamma_s(\varphi_1, T^\star, T'', x)$;
—$\Gamma_s(\varphi_1 \mathbb{R}_\phi \varphi_2, T^\star, T, x) \triangleq \forall T'.\mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T, T', x) \rightarrow \Gamma_s(\varphi_2, T^\star, T', x) \vee \exists T''.\mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T, T'', x) \wedge \mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T'', T', x) \wedge \Gamma_s(\varphi_1, T^\star, T'', x)$;
—$\Gamma_s(\varphi_1 \mathbb{S}_\phi \varphi_2, T^\star, T, x) \triangleq \exists T'.\mathtt{SupTree}_{\phi'(y)}(T^\star, T, T', x) \wedge \Gamma_s(\varphi_2, T^\star, T', x) \wedge \forall T''.\mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T', T'', x) \wedge \mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T'', T, x) \rightarrow \Gamma_s(\varphi_1, T^\star, T'', x)$;
—$\Gamma_s(\varphi_1 \mathbb{B}_\phi \varphi_2, T^\star, T, x) \triangleq \forall T'.\mathtt{SupTree}_{\phi'(y)}(T^\star, T, T', x) \rightarrow \Gamma_s(\varphi_2, T^\star, T', x) \vee \exists T''.\mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T', T'', x) \wedge \mathtt{SubTree}^{\sqsubset}_{\phi'(y)}(T'', T, x) \wedge \Gamma_s(\varphi_1, T^\star, T'', x)$,
where $\phi'(y) \triangleq \Gamma_s(\phi, T^\star, T, y)$.

The last group deals with path quantifiers, using the construct $\mathtt{Path}(T, P, x)$ to restrict the range of the quantified second-order variables to a path P of T, in accordance with Table I:

(4) —$\Gamma_s(\mathrm{E}\psi, T^\star, T, x) \triangleq \exists P \,.\, \mathtt{Path}(T, P, x) \wedge \Gamma_p(\psi, T^\star, T, P, x)$;
—$\Gamma_s(\mathrm{A}\psi, T^\star, T, x) \triangleq \forall P \,.\, \mathtt{Path}(T, P, x) \rightarrow \Gamma_p(\psi, T^\star, T, P, x)$.

Finally, we can encode the semantics of path formulas using the translation functions Γ_p. Once again, the following first two groups take care of atomic propositions and Boolean connectives:

(5) $\Gamma_p(\varphi, T^\star, T, P, x) \triangleq \Gamma_s(\varphi, T^\star, T, x)$.
(6) —$\Gamma_p(\neg\psi, T^\star, T, P, x) \triangleq \neg\Gamma_p(\psi, T, T^\star, P, x)$;
—$\Gamma_p(\psi_1 \wedge \psi_2, T^\star, T, P, x) \triangleq \Gamma_p(\psi_1, T^\star, T, P, x) \wedge \Gamma_p(\psi_2, T^\star, T, P, x)$;
—$\Gamma_p(\psi_1 \vee \psi_2, T^\star, T, P, x) \triangleq \Gamma_p(\psi_1, T^\star, T, P, x) \vee \Gamma_p(\psi_2, T^\star, T, P, x)$.

The last group, instead, deals with temporal operators $\mathtt{X}$, $\mathtt{U}$, and $\mathtt{R}$ and assumes that the argument P is a path of T:

(7) —$\Gamma_p(\mathtt{X}\psi, T^\star, T, P, x) \triangleq \exists y \,.\, y \in P \wedge x \lessdot y \wedge \Gamma_p(\psi, T^\star, T, P, y)$;
—$\Gamma_p(\psi_1 \mathtt{U} \psi_2, T^\star, T, P, x) \triangleq$
$\exists y \,.\, y \in P \wedge x \leq y \wedge \Gamma_p(\psi_2, T^\star, T, P, y) \wedge \forall z \,.\, z \in P \ \wedge z < y \rightarrow \Gamma_p(\psi_1, T^\star, T, P, z)$;
—$\Gamma_p(\psi_1 \mathtt{R} \psi_2, T^\star, T, P, x) \triangleq$
$\forall y \,.\, y \in P \leq y \rightarrow \Gamma_p(\psi_2, T^\star, T, P, y) \vee \exists z \,.\, z \in P \ \wedge z < y \wedge \Gamma_p(\psi_1, T^\star, T, P, z)$.

An easy but tedious induction on the structure of the formula allows one to prove that, for every STL* formula φ and tree $\mathcal{T}$,

$$\mathcal{T} \models_{\text{STL}^\star} \varphi \text{ iff } \mathcal{T} \models_{\text{MSOL}} \exists T^\star . \exists x . \mathtt{Mod}(T^\star, x) \wedge \Gamma_s(\varphi, T^\star, T^\star, x),$$

where $\mathtt{Mod}(T^\star, x) \triangleq \forall y . y \in T^\star \wedge x \leq y$ ensures that $T^\star$ evaluates to tree $\mathcal{T}$ and x to its root. □

6.3. Evaluation of the Model-Theoretic Results

We conclude this section with a brief discussion on the model-theoretic results obtained for STL*. As shown by Theorems 6.4 and 6.9, the logic lacks some model-theoretic properties, which are usually considered as desirable properties for a logic. In particular, STL* does not enjoy invariance under bisimulation both in the Kripke structures' and the regular Kripke trees' interpretations. This seems, however, to be an unavoidable price to pay for the additional expressive power required to model relevant game-theoretic notions such as, for instance, Nash equilibria. Indeed, the same phenomenon occurs in other logics for games comparable to STL* in expressive power, most notably Strategy Logic (SL) [Chatterjee et al. 2007; Mogavero et al. 2010, 2014], which is able to express similar properties and lacks invariance under bisimulation as well.

Similar considerations also apply to the lack of invariance under unwinding when interpreted on Kripke structures. In particular, the absence of this property proves to be fundamental in allowing for modeling the concept of finite-memory strategy, which can be expressed in STL* but cannot even be expressed in SL, unless one directly modifies its semantics. Finally, the fact that STL* does not enjoy the finite-model property is a consequence of the ability of the logic to express properties of infinite-state systems.

Figure 13 summarizes the expressiveness results obtained in this section and relating the fragments of STL* with the classic temporal and monadic second-order logics. The figure clearly shows that STL* subsumes all the classic temporal logics and is subsumed only by MSOL. However, being a natural extension of temporal logics, STL* preserves, differently from MSOL, both the syntactic and the semantic structures of the subsumed logics, making it a natural candidate as a unifying temporal framework for reasoning about games, as shown by the case studies provided in Section 5. Indeed, encoding properties in MSOL most often proves to be a difficult task, as clearly witnessed by the proof of Theorem 6.13. The main reason is that MSOL completely hides under several levels of first- and second-order quantifications the underlying structure of the encoded properties. This is particularly evident in the MSOL encodings of temporal properties, where most of the temporal structure gets lost in the translation. As we will see at the end of the next section, the intrinsic temporal nature of STL* also allows one to exploit standard verification techniques for temporal logics in order to obtain, in most cases, optimal decision procedures for the problems considered in the article.

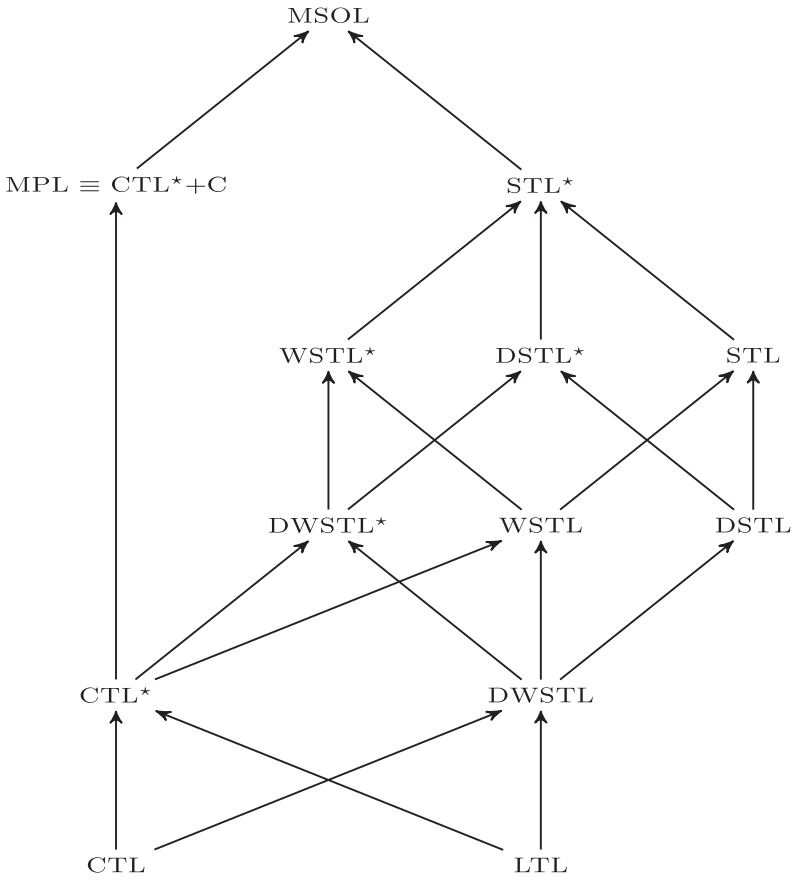

Fig. 13. Expressiveness hierarchy.

7. DECISION PROBLEMS

Depending on the class of models over which the logic is interpreted, complexity results of the standard decision problems, namely, satisfiability and model checking, differ significantly. For instance, when interpreted over arbitrary Kripke structures, satisfiability is undecidable already for DWSTL⋆. However, the problem for the full STL⋆ remains decidable, in nonelementary time, when interpreted on regular Kripke trees, that is, finitely representable trees as those induced by an unwinding of a finite structure. 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 nonelementary time, for regular Kripke trees. The following theorems summarize the results.

7.1. Results on KSs

Undecidability of STL⋆ interpreted in the wider class of KS follows from a reduction from the *recurrent domino problem* [Harel 1984], which is known to be highly undecidable and, in particular, Σ_1^1-COMPLETE, that is, not even computably enumerable.

The *domino problem*, proposed for the first time by Wang [1961], consists of placing a given number of tile types on an infinite grid, satisfying a predetermined set of constraints on adjacent tiles. Its standard version asks for a compatible tiling of the

whole plane $\mathbb{N} \times \mathbb{N}$. The *recurrent domino problem* further requires the existence of a distinguished tile type that occurs infinitely often in the first row of the grid. The formal definition follows.

Definition 7.1 (*Recurrent Domino System*). An $\mathbb{N}\times\mathbb{N}$ *recurrent domino system* $\mathcal{D} = \langle \mathrm{D}, H, V, t\rangle$ consists of a finite nonempty set D of *domino types*; two *horizontal* and *vertical matching relations* $H, V \subseteq \mathrm{D} \times \mathrm{D}$; and a distinguished tile type $t^* \in \mathrm{D}$. The recurrent domino problem asks for an *admissible tiling* of $\mathbb{N} \times \mathbb{N}$, which is a *solution mapping* $\partial : \mathbb{N} \times \mathbb{N} \to \mathrm{D}$ such that, for all $x, y \in \mathbb{N}$, it holds that (1) $(\partial(x, y), \partial(x + 1, y)) \in H$, (2) $(\partial(x, y), \partial(x, y + 1)) \in V$, and (3) $|\{x : \partial(x, 0) = t^*\}| = \infty$.

We show that a recurrent tiling system can be embedded into a model of a particular DWSTL* formula, which is satisfiable if and only if the tiling system allows for an admissible tiling. This suffices to show the undecidability of DWSTL* and, *a fortiori*, of STL*.

Theorem 7.2 (DWSTL*[KS] Undecidable Satisfiability). *The* DWSTL*[KS] *satisfiability problem is highly undecidable; that is, it is* Σ^1_1-hard.

Proof. We embedded a recurrent tiling system $\mathcal{D}$ into a model of a particular DWSTL* formula $\varphi \triangleq \varphi_{grd} \wedge \varphi_{til} \wedge \varphi_{rec}$, which is satisfiable if and only if the tiling system allows for an admissible tiling ∂. The hardest part of the proof consists of the definition of a suitable satisfiable DWSTL* formula φ_{grd}, all of whose models $\mathcal{K}^{grd}$ contain the infinite grid $\mathbb{N} \times \mathbb{N}$ of the tiling problem or, in other words, admit an infinite square grid graph as a minor. Given φ_{grd}, the remaining part of the reduction can easily be completed by using CTL* formulas only, in a way that is similar to the one explained in the undecidability proof of CTL with a minimal model quantifier [Mogavero and Murano 2009]. In particular, φ_{til} ensures that the placing of the domino types is coherent with the horizontal and vertical matching relations H and V, while φ_{rec} forces the distinguished tile type t^* to occur infinitely often on a row of the grid. Hence, in the rest of the proof, we focus on the construction of φ_{grd} only. It is important to observe that our formula φ_{grd} is significantly different from the corresponding one used in Mogavero and Murano [2009], 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 on *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*, for this purpose and require that every world satisfies $\varphi_{flg} \triangleq \mathtt{EX}c \wedge \mathtt{EX}\neg c$. 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 through 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 $\varphi_{num} \triangleq \bigwedge_{i=0}^{3} i \rightarrow \mathtt{AX}((i + 1) \bmod 4)$, 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 a length multiple of four (see Figure 14).

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 correspond 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

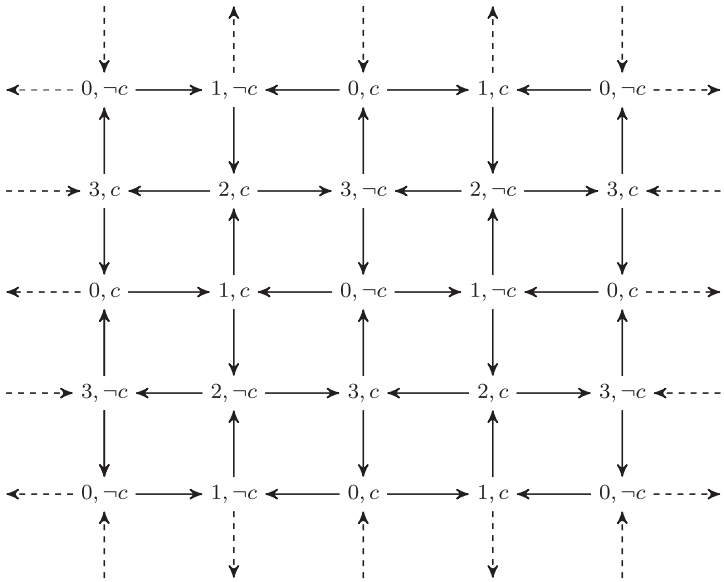

Fig. 14. WSTL*[KS] undecidability ($\mathcal{K}_{grd}$).

not. The formula φ_{num} already guarantees 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 14, there are another 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 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 into two groups, one of which contains only 4-tracks of even parity, that is, tracks where the flag c occurs zero, two, or four times. 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 $\psi_{pth} \triangleq \bigvee_{(\flat_0,\flat_1,\flat_2,\flat_3)\in \mathrm{P}}(\flat_0 \wedge \mathtt{X}(\flat_1 \wedge \mathtt{X}(\flat_2 \wedge \mathtt{X}(\flat_3 \wedge \mathtt{X}\flat_0))))$, 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 $\varphi^{\flat}_{cyc} \triangleq \mathtt{E}(\psi_{pth} \wedge \mathtt{X}\flat \wedge \mathtt{X}^4\mathtt{EX}\neg\flat) \rightarrow \mathtt{EX}\neg\flat$. Indeed, suppose by contradiction that there is a path $\pi \in \mathrm{Pth}_{\mathcal{K}_w}$, with $\mathcal{K}_w, \pi, 0 \models \psi_{pth} \wedge \mathtt{X}\flat \wedge \mathtt{X}^4\mathtt{EX}\neg\flat$, that does not close the cycle on w after 4 steps, that is, $(\pi)_4 \neq (\pi)_0 = w$. Now, let $\mathcal{K}' \in \mathfrak{F}^{\downarrow \mathcal{K}^{grd}_w}_{\mathcal{K}_w}(\mathtt{f})$ be one of the minimal substructures of $\mathcal{K}_w$ such that $\pi \in \mathrm{Pth}_{\mathcal{K}'}$. It can immediately be seen that $\mathcal{K}' \models \mathtt{E}(\psi_{pth} \wedge \mathtt{X}\flat \wedge \mathtt{X}^4\mathtt{EX}\neg\flat)$. However, $\mathcal{K}' \not\models \mathtt{EX}\neg\flat$, 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 that we require $\mathtt{EX}\neg\flat$ on $(\pi)_4$ in $\mathcal{K}'$ but not on $w = (\pi)_0 \neq (\pi)_4$.

Requiring φ_{flg}, φ_{num}, and φ_{cyc} on all worlds of $\mathcal{K}^{grd}$ simply amounts to verifying the formula $\mathtt{AG}(\varphi_{flg} \wedge \varphi_{num} \wedge \varphi_{cyc})$ on $\mathcal{K}^{grd}$. Notice, however, that this formula is also satisfied on the quatrefoil KS partially depicted in Figure 15, 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 the uniqueness of $\flat$-successors and $\flat$-predecessor for each flag $\flat \in \{c, \neg c\}$.

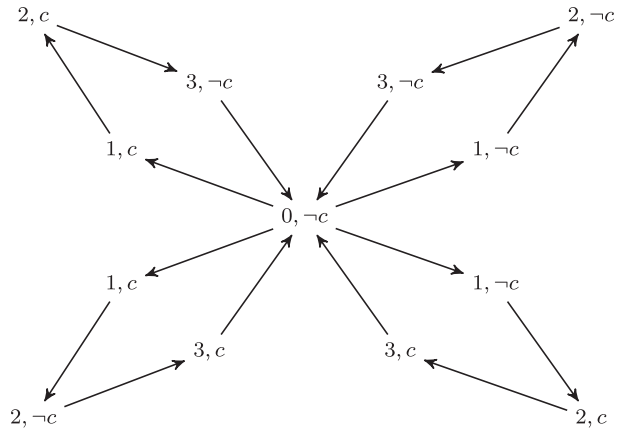

Fig. 15. A quatrefoil KS.

For the uniqueness of $\flat$-successors, it suffices to require the formula $\mathbb{G}\varphi^{\flat}_{suc}$, where $\varphi^{\flat}_{suc} \triangleq \varphi^{\flat}_{s} \rightarrow \mathtt{EX}\varphi_{flg}$ and $\varphi^{\flat}_{s} \triangleq \mathtt{EX}(\flat \wedge \mathtt{EX}c) \wedge \mathtt{EX}(\flat \wedge \mathtt{EX}\neg c)$. 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 $\mathtt{EX}\varphi_{flg}$. 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 $\varphi^{\flat}_{pre} \triangleq \varphi^{\flat}_{p} \rightarrow \mathtt{EX}^3\varphi_{flg}$ and $\varphi^{\flat}_{p} \triangleq \mathtt{E}(\psi_{pth} \wedge \mathtt{X}^3(\flat \wedge \mathtt{EX}c)) \wedge \mathtt{E}(\psi_{pth} \wedge \mathtt{X}^3(\flat \wedge \mathtt{EX}\neg c))$, ensure the uniqueness of $\flat$-predecessors.

We can finally conclude that the DWSTL* formula $\varphi_{grd} \triangleq \mathtt{AG}(\varphi_{flg} \wedge \varphi_{num} \wedge \varphi_{cyc} \wedge \varphi_{suc} \wedge \varphi_{pre})$ has precisely the KS $\mathcal{K}^{grd}$ of Figure 14 as model. □

As opposed to the satisfiability problem, the model-checking problem for STL* formulas against a KS is decidable and, in particular, results to be PSPACE -COMPLETE. The following theorem provides a brute-force recursive algorithm that checks whether a finite KS $\mathcal{K}$ satisfies an STL* formula φ using polynomial space both on the size of the formula and on the KS $\mathcal{K}$. PSPACE hardness is established by a reduction from the satisfiability problem for *Quantified Boolean Formulas* (QBFs), which was proved complete for PSPACE in Stockmeyer and Meyer [1973].

THEOREM 7.3 (STL*[KS] DECIDABLE MODEL CHECKING). *The* STL*[KS] *model-checking problem is* PSPACE -COMPLETE *w.r.t. both the size of the* STL* *formula* φ *and the finite* KS *model* $\mathcal{K}$.

PROOF. Let us first consider the upper bound. 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 $\mathcal{K}$ [Kupferman et al. 2000]. 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 an 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 into the following phases:

(1) Identification of the subset $W' \subseteq W_{\mathcal{K}}$ such that $w \in W'$ if and only if $\mathcal{K}_w \models \phi$, for all $w \in W_{\mathcal{K}}$;
(2) Guess of the substructure $\mathcal{K}' \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(W')$;
(3) Check for $\mathcal{K}' \models \varphi_2$;
(4) Guess of the substructure $\mathcal{K}'' \in \overline{\mathfrak{F}}^{\downarrow}_{\mathcal{K}}(W')$ 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 the model $\mathcal{K}$, and since PSPACE = NPSPACE, the thesis follows for φ too.

As to the lower bound, loosely inspired by the hardness proof of the clique problem, we will make a reduction from the 3-QBF satisfiability problem, which is known to be PSPACE -COMPLETE, to the model checking of DSTL[KS]. Let $\varphi = \wp\psi$ be a QBF formula over the Boolean variables in $X = \{x_1, \ldots, x_n\}$, where $\wp$ is the quantification prefix of the form $\mathtt{Qn}_1 x_1 \cdots \mathtt{Qn}_k x_k$, with $\mathtt{Qn}_i \in \{\exists, \forall\}$ for $i \in [1, k]$, and the matrix $\psi = \bigwedge_{i=1}^{m}(\ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3})$ is a Boolean formula in conjunctive normal form over the variables in X. We transform φ into a suitable STL formula $\widetilde{\varphi}$ over the set of atomic propositions $AP \triangleq \{x, \overline{x} : x \in X\}$, where, for each $x \in X$, we need two atomic propositions, one for the positive literal x and one for the negative one $\neg x$. In the formula $\widetilde{\varphi}$, propositional quantifications are replaced by the corresponding maximal substructure constructs applied to a suitable KS encoding the matrix ψ.

The desired KS $\mathcal{K}_{MC}$, against which we check $\widetilde{\varphi}$, is a graph having a node $(i, p_{i,j})$, with $p_{i,j} \in AP$, for each possible clause i and literal $\ell_{i,j}$. We say that two nodes (i_1, p_{i_1,j_1}) and (i_2, p_{i_2,j_2}) are *complementary* if ℓ_{i_1,j_1} and ℓ_{i_2,j_2} are complementary literals. The idea is that $\mathcal{K}_{MC}$ contains paths having a node $(i, p_{i,j})$ for each clause $i \in [1, m]$. Moreover, any path that does not contain two complementary nodes encodes a witness for the matrix ψ and is called *consistent*. We say that two consistent paths are *complementary* if they contain complementary nodes. Therefore, a node of $\mathcal{K}_{MC}$ associated to the ith clause, with $i \in [1, m[$, is connected to all those of the $(i + 1)$-th clause. In addition, each node of the last clause only has a self-loop. As an example, the four structures depicted in Figure 16 correspond to copies of the $\mathcal{K}_{MC}$ for the matrix $\eta = (p \vee q \vee r) \wedge (p \vee \neg q \vee r) \wedge (\neg p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$, where the additional node 0 is used as the initial world. In general, the KS $\mathcal{K}_{MC}$ is formally defined as follows: $\mathcal{K}_{MC} \triangleq \langle AP, W, R, L, 0\rangle$, where the set of worlds is $W \triangleq \{0\} \cup \{(i, x) \in [1, m] \times AP : \exists j \in [1, 3] \,.\, \ell_{i,j} = x\} \cup \{(i, \overline{x}) \in [1, m] \times AP : \exists j \in [1, 3] \,.\, \ell_{i,j} = \neg x\}$, the transition relation is $R \triangleq \{(0, (1, p))\} \cup \{((i, p'), (i + 1, p'')) : i \in [1, m[\} \cup \{((m, p), (m, p))\}$, and the labeling function L is defined as follows: $L(0) \triangleq \emptyset$ and $L((i, p)) \triangleq \{p\}$, for all $(i, p) \in W$.

Observe that any maximal substructure of $\mathcal{K}_{MC}$ containing only noncomplementary consistent paths corresponds to a single model of the matrix. For instance, each substructure identified by the bold arrows in Figure 16 denotes the four possible models of η. Moreover, a structure only contains noncomplementary consistent paths if and only if it satisfies the CTL formula $\flat_x \triangleq \mathtt{AG}\neg x \vee \mathtt{AG}\neg\overline{x}$, for each $x \in \mathtt{x}$.

Therefore, in order to deal with the Boolean existential (universal, respectively) quantification $\exists x$ ($\forall x$, respectively), we just need to select the desired maximal substructures by means of the existential (universal, respectively) relative downward maximal construct, using $\flat_x$ as the first argument.

Formally, we define a translation function $\widetilde{\cdot}$: 3-QBF $\rightarrow$ STL as follows:

—$\widetilde{\exists x \,.\, \psi} \triangleq \overline{\mathtt{EMax}}^{\downarrow}(\flat_x, \widetilde{\psi})$;
—$\widetilde{\forall x \,.\, \psi} \triangleq \overline{\mathtt{AMax}}^{\downarrow}(\flat_x, \widetilde{\psi}) \wedge \overline{\mathtt{EMax}}^{\downarrow}(\flat_x, \mathtt{t})$;
—$\widetilde{\psi} \triangleq \mathtt{t}$, when ψ is a Boolean formula.

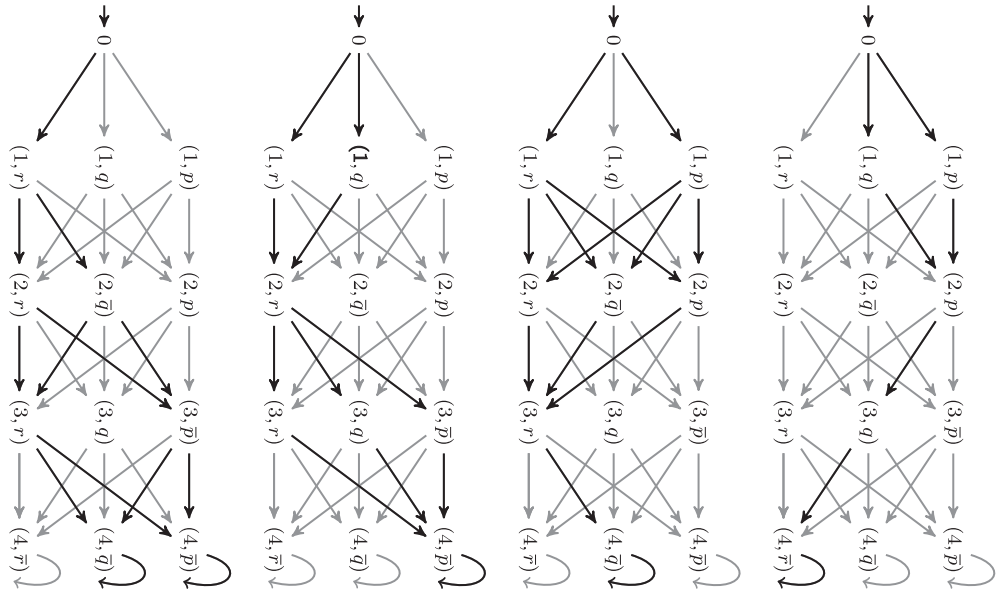

Fig. 16. Reduction from the 3-QBF sentence $\varphi = \forall r \exists p \exists q \psi$, where $\psi = \bigwedge_{i=1}^{4}(\ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}) = (p \vee q \vee r) \wedge (p \vee \neg q \vee r) \wedge (\neg p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$.

Note that the additional conjunct $\overline{\texttt{EMax}}^{\downarrow}(\flat_x, \mathtt{t})$ in the translation of the universal quantification is needed in order to ensure that at least one substructure satisfying the constraint $\flat_x$ actually exists. □

7.2. Results on KTs

When STL⋆ is interpreted over the class KT of Kripke trees, satisfiability becomes decidable with nonelementary complexity. Clearly, the model checking remains decidable, though with a significant increase of its complexity. The two problems have, precisely, $(k+1)$-ExpTime complexity, where k is the number of alternations among the semilattice operators occurring in the formula to decide. Theorem 7.4 formally establishes these results by providing the upper bound for both problems.

Theorem 7.4 (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 $\mathcal{K} \in$ KS(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, recognizing the KTs satisfying it. If the language recognized by $\mathcal{A}_\varphi$ is not empty, then φ is satisfiable. By testing nonemptiness of the one-letter automaton $\mathcal{A}_{\mathcal{K},\varphi}$ resulting from the product of $\mathcal{A}_\varphi$ and the KS $\mathcal{K}$, we obtain a model-checking procedure to test whether $\mathcal{K}^U \overset{\mathcal{K}^U}{\models} \varphi$.

To build the APT $\mathcal{A}_\varphi$, we use an inductive procedure on the structure of the formula φ, which extends the classic construction proposed in Kupferman et al. [2000] to semilattice operators. This is done by means of additional labelings, used to identify the substructures quantified over in the semantics of those operators. Observe that this approach generalizes the one used to solve the module-checking problem for CTL⋆ [Kupferman et al. 2001].

We first recall how to build the automaton $\mathcal{A}_\varphi$, for a given CTL* formula φ. The Boolean cases are dealt with by exploiting the closure properties of alternating automata. Path quantifiers $\mathtt{E}\psi$ and $\mathtt{A}\psi$, where ψ is an LTL formula, are translated into an APT, which runs the Vardi-Wolper linear automaton for ψ on one or all paths of the underlying tree in input. Finally, when ψ contains state formulas of the form $\mathtt{E}\psi'$ or $\mathtt{A}\psi'$, we proceed inductively, first constructing the automaton for those state formulas, then building the automaton for the external formula, where all occurrences of the internal ones are replaced by fresh atomic propositions, and finally intersecting of the resulting automata.

For STL*, we need to extend the previous procedure by providing a construction for the semilattice operators. In order to check the corresponding properties on the subtrees identified by the operators, we also need to slightly modify the cases of path quantifiers.

In the following, $\mathsf{aut}(\varphi', i, k)$ denotes the APT for the STL* formula φ', where $k \in \{1, 2\}$ is a flag used to distinguish two different labelings and $i \in \mathbb{N}$ counts the number of semilattice operators having φ' in their scope, that is, the number of operators traversed to reach the subformula φ', starting from the root of the original formula φ.

Beginning with the STL* formula of interest φ, we inductively build the automaton $\mathcal{A}_\varphi \triangleq \Pi^{\exists}_{l^1_0}(\mathcal{A}_{\mathtt{AG}l^1_0} \cap \mathsf{aut}(\varphi, 0, 1))$ as described in the following, where $\Pi^{\exists}_l\mathcal{A}$ denotes the existential projection operator applied on a given automaton $\mathcal{A}$ (see, e.g., Khoussainov and Nerode [2001]), whose result is the automaton accepting the language obtained by existentially quantifying out the propositional variable l from the language accepted by $\mathcal{A}$. Note that the projection and intersection operations are used to ensure that the initial labeling l^1_0 corresponds to the whole input tree $\mathcal{T}^\star$, namely, the unwinding of the top structure $\mathcal{K}^\star$. Moreover, in the following, we make use of the universal projection $\Pi^{\forall}_l\mathcal{A}$ to denote the automaton that accepts a language obtained by quantifying out in a universal way the proposition l from the language accepted by $\mathcal{A}$.

Intuitively, for each (sub)formula, the corresponding APT reads a subtree of the original input tree, which is identified by a suitable labeling, and checks whether the formula is true in that subtree. In order to check a CTL* (sub)formula φ' on the subtree identified by a labeling l, we need to transform the φ' so that temporal operators are interpreted only over the paths of the input tree labeled by l. This is achieved by relativizing the temporal operators to the desired labeling by means of the following translation function:

—$\mathsf{lab}_l(p) \triangleq p$;
—$\mathsf{lab}_l(\mathtt{Op}\varphi) \triangleq \mathtt{Op}\mathsf{lab}_l(\varphi)$, where $\mathtt{Op} \in \{\neg, \mathtt{X}\}$;
—$\mathsf{lab}_l(\varphi_1 \mathtt{Op} \varphi_2) \triangleq \mathsf{lab}_l(\varphi_1)\,\mathtt{Op}\mathsf{lab}_l(\varphi_2)$, where $\mathtt{Op} \in \{\wedge, \vee, \mathtt{U}, \mathtt{R}\}$;
—$\mathsf{lab}_l(\mathtt{E}\psi) \triangleq \mathtt{E}(\mathtt{G}\,l \wedge \mathsf{lab}_l(\psi))$;
—$\mathsf{lab}_l(\mathtt{A}\psi) \triangleq \mathtt{A}(\mathtt{G}\,l \rightarrow \mathsf{lab}_l(\psi))$.

Given a CTL* (sub)formula φ' and the indexes i and k, identifying the labeling l^k_i of the current subtree, we set $\mathsf{aut}(\varphi', i, k) \triangleq \mathcal{A}_{\mathsf{lab}_{l^k_i}(\varphi)}$, that is, the APT obtained by the classic Kupferman-Vardi-Wolper procedure applied to the CTL* formula $\mathsf{lab}_{l^k_i}(\varphi)$.

For an STL* formula φ' whose outermost operator is not a semilattice operator, we perform the following steps: (1) construct the automata $\mathsf{aut}(\varphi_1 \mathtt{Op}_\phi \varphi_2, i, k)$ for all the subformulas $\varphi_1 \mathtt{Op}_\phi \varphi_2$ with $\mathtt{Op} \in \{\mathbb{U}, \mathbb{S}\}$; (2) build the automaton for the CTL* formula, where each semilattice construct is replaced by a fresh atomic proposition as described earlier; and, finally, (3) compose the resulting automata as in the classic procedure. With more detail, if $\{\varphi^h_1 \mathtt{Op}^h_\phi \varphi^h_2\}$ is the finite set of semilattice constructs occurring in

the formula φ', we first derive all automata $\mathsf{aut}(\varphi_1^h \mathtt{Op}_\phi^h \varphi_2^h, i, k)$. Then, we build the CTL* automaton $\mathcal{A}_{\varphi''}$ for the formula φ'' obtained by replacing in $\mathsf{lab}_{l_i^k}(\varphi')$ all occurrences of $\varphi_1^h \mathtt{Op}_\phi^h \varphi_2^h$ with a fresh atomic proposition p_h. Finally, we define an automaton $\mathcal{A}'_{\varphi'}$ that, in parallel to $\mathcal{A}_{\varphi''}$, starts a copy of $\mathsf{aut}(\varphi_1^h \mathtt{Op}_\phi^h \varphi_2^h, i, k)$ at all those nodes where φ'' requires the proposition p_h to hold.

To deal with the remaining formulas of the forms $\varphi' = \varphi_1 \mathbb{U}_\phi \varphi_2$ and $\varphi' = \varphi_1 \mathbb{S}_\phi \varphi_2$, we will simulate the quantifications contained inside these semilattice operators and ranging over the subtrees of the input tree, by quantifying over the corresponding sublabelings of the top labeling l_0^1. In other words, we encode a quantification over a subtree $\mathcal{T}$ of the original tree $\mathcal{T}^\star$ by means of a sublabeling l of the labeling l_0^1 identifying the whole $\mathcal{T}^\star$. In order to characterize those labelings l that correspond to proper subtrees $\mathcal{T}$ of the input tree and to check inclusion between labelings (the subtree relation), we use the following CTL formulas:

—$\varphi_l \triangleq l \wedge \mathtt{AG}((l \rightarrow \mathtt{EX}l) \wedge (\neg l \rightarrow \mathtt{AX}\neg l))$, which verifies that l correctly identifies a subtree of the original one;
—$\varphi_{l' \sqsubseteq l} \triangleq \mathtt{AG}(l' \rightarrow l)$, which checks that the tree corresponding to l' is a subtree of the one identified by l;
—$\varphi_{l' \sqsubset l} \triangleq \varphi^{l' \sqsubseteq l} \wedge \mathtt{EF}(l \wedge \neg l')$, which ensures that the inclusion between l' and l is strict.

Let $\mathcal{A}_{l' \sqsubset l}$, $\mathcal{A}_{l' \sqsubset l'' \sqsubseteq l}$, and $\mathcal{A}_{\overline{l' \sqsubset l'' \sqsubset l}}$ be the automata for the CTL formulas $(\varphi_l \wedge \varphi_{l' \sqsubset l})$, $(\varphi_{l''} \wedge \varphi_{l' \sqsubset l''} \wedge \varphi_{l'' \sqsubseteq l})$, and $\neg(\varphi_{l''} \wedge \varphi_{l' \sqsubset l''} \wedge \varphi_{l'' \sqsubset l})$, respectively.

Finally, to account for the selector parameter ϕ, we need to enforce that the subtrees quantified over preserve all the edges of the current tree exiting from worlds satisfying ϕ. To this aim, we use the STL* formula $\varphi^{\phi \rightarrow l} \triangleq \mathtt{AG}((l \wedge \phi) \rightarrow \mathtt{AX}l)$. This formula requires that, for any world w of the subtree of l satisfying ϕ, each successor of w in the current tree is labeled by l as well.

At this point, we can define the automaton for the semilattice operators as follows:

$$\mathsf{aut}(\varphi_1 \mathbb{U}_\phi \varphi_2, i, k) \triangleq \Pi^{\exists}_{l^2_{i+1}} \Big[\mathcal{A}_{l^2_{i+1} \sqsubset l^k_i} \cap \mathsf{aut}(\varphi^{\phi \rightarrow l^2_{i+1}}, i, k) \cap \mathsf{aut}(\varphi_2, i+1, 2) \cap \Pi^{\forall}_{l^1_{i+1}} \Big(\mathcal{A}_{\overline{l^2_{i+1} \sqsubset l^1_{i+1} \sqsubset l^k_i}} \cup \mathsf{aut}(\neg \varphi^{\phi \rightarrow l^1_{i+1}}, i, k) \cup \mathsf{aut}(\varphi_1, i+1, 1) \Big) \Big];$$

$$\mathsf{aut}(\varphi_1 \mathbb{S}_\phi \varphi_2, i, k) \triangleq \Pi^{\exists}_{l^2_{i+1}} \Big[\mathcal{A}_{l^k_i \sqsubset l^2_{i+1} \sqsubseteq l^1_0} \cap \mathsf{aut}(\varphi^{\phi \rightarrow l^k_i}, i+1, 2) \cap \mathsf{aut}(\varphi_2, i+1, 2) \cap \Pi^{\forall}_{l^1_{i+1}} \Big(\mathcal{A}_{\overline{l^k_i \sqsubset l^1_{i+1} \sqsubset l^2_{i+1}}} \cup \mathsf{aut}(\neg \varphi^{\phi \rightarrow l^k_i}, i+1, 1) \cup \mathsf{aut}(\varphi_1, i+1, 1) \Big) \Big].$$

The two projection operators on the labelings account for the associated quantification over subtrees in the semantics of the corresponding operator. The automaton for $\varphi_1 \mathbb{U} \varphi_2$ checks the formula on the subtree labeled by l_i^k and is the result of an existential projection of the label l_{i+1}^2 (a possible subtree) of the intersection of four automata. The APT $\mathcal{A}_{l^2_{i+1} \sqsubset l^k_i}$ checks that l_{i+1}^2 is a strict sublabeling of l_i^k. The APT $\mathsf{aut}(\varphi^{\phi \rightarrow l_i^k}, i+1, 2)$ ensures that all the worlds in the subtree identified by l_{i+1}^2 that satisfy the selector ϕ (recall that this may be a STL* formula) have the same outgoing edges as in the subtree labeled l_i^k, and $\mathsf{aut}(\varphi_2, i+1, 2)$ checks that the selected subtree satisfies φ_2. The result of the universal projection takes care of the remaining conditions for the satisfaction of the operator. It requires that all the subtrees between l_i^k and l_{i+1}^2, which preserve the outgoing edges from worlds satisfying ϕ, do satisfy φ_1. The automaton for $\varphi_1 \mathbb{S}_\phi \varphi_2$ is defined similarly. The only relevant difference is that subtree quantifications

range only over the subtrees of the input one (i.e., l_0^1), which are superstructures of the current subtree l_i^k. In addition, $\mathsf{aut}(\varphi^{\phi \to l_i^k}, i+1, 2)$ requires that the worlds satisfying ϕ in the selected supertree l_{i+1}^2 have the same outgoing edges as in the current subtree l_i^k, as required by the semantics.

The construction can be proved correct by induction on the structure of the formula. For all the syntactic cases but the two semilattice operators $\mathbb{U}$ and $\mathbb{S}$, the correctness immediately follows from the correctness of the automata construction for CTL* proposed in Kupferman et al. [2000] and used here as well. Therefore, we will concentrate later on the inductive case for the operator $\mathbb{U}$ (the case for $\mathbb{S}$ follows essentially the same line of reasoning). Let $\mathcal{T}$ be a subtree of $\mathcal{T}^\star$ identified by the labeling l_i^k. Then, the correctness of the construction for $\varphi_1 \mathbb{U}_\phi \varphi_2$ amounts to prove that $\mathcal{T} \models^{\mathcal{T}^\star} \varphi_1 \mathbb{U}_\phi \varphi_2$ if and only if the tree $\mathcal{T}'$, obtained by labeling each node of $\mathcal{T}$ according to the sublabeling l_i^k, belongs to the language accepted by the automaton $\mathsf{aut}(\varphi_1 \mathbb{U}_\phi \varphi_2, i, k)$. Assume that $\mathcal{T} \models^{\mathcal{T}^\star} \varphi_1 \mathbb{U}_\phi \varphi_2$. Then, by definition of the semantics, there exists a strict subtree $\mathcal{T}_2 \in \mathfrak{F}_{\mathcal{T}}^{\downarrow \mathcal{T}^\star}(\phi)$ of $\mathcal{T}$ such that $\mathcal{T}_2 \models^{\mathcal{T}^\star} \varphi_2$, and, for all subtrees $\mathcal{T}_1 \in \mathfrak{F}_{\mathcal{T}}^{\downarrow \mathcal{T}^\star}(\phi)$ of $\mathcal{T}$ that are also strict supertrees of $\mathcal{T}_2$, it holds that $\mathcal{T}_1 \models^{\mathcal{T}^\star} \varphi_1$. Now, let l_{i+1}^2 be the labeling that identifies $\mathcal{T}_2$ as a strict subtree of $\mathcal{T}$. Clearly, this is a strict sublabeling of l_i^k and identifies $\mathcal{T}_2$ as a strict subtree of $\mathcal{T}^\star$ as well. Let the tree $\mathcal{T}_2'$ be obtained by augmenting the labeling of $\mathcal{T}'$ with the labeling l_{i+1}^2. Then, it is necessarily accepted by $\mathcal{A}_{l_{i+1}^2 \sqsubset l_i^k}$. Moreover, as $\mathcal{T}_2$ belongs to the filtering induced by ϕ, we have that it preserves all edges outgoing from a state satisfying this formula. Consequently, $\mathcal{T}_2'$ is also accepted by $\mathsf{aut}(\varphi^{\phi \to l_{i+1}^2}, i, k)$. Finally, since $\mathcal{T}_2 \models^{\mathcal{T}^\star} \varphi_2$, by inductive hypothesis, we have that $\mathcal{T}_2'$ belongs to the language accepted by $\mathsf{aut}(\varphi_2, i+1, 2)$. To conclude this direction of the proof, we need to prove that $\mathcal{T}_2'$ is also accepted by the remaining part $\Pi^\forall_{l_{i+1}^1}(\ldots)$ of the automaton. Consider a strict sublabeling l_{i+1}^1 of l_i^k that is also a strict superlabeling of l_{i+1}^2. Moreover, ensure that l_{i+1}^1 preserves all edges between states labeled by l_i^k that have origin in states that also satisfy ϕ. It is obvious that the strict subtree $\mathcal{T}_1$ of $\mathcal{T}$ identified by l_{i+1}^1 is also a strict supertree of $\mathcal{T}_2$. Moreover, it belongs to the filtering induced by ϕ. Now, due to the particular choice of l_{i+1}^1, it is not hard to see that the tree $\mathcal{T}_1'$, obtained by augmenting the labeling of $\mathcal{T}'$ with the labeling l_{i+1}^1, is accepted neither by $\mathcal{A}_{\overline{l_{i+1}^2 \sqsubset l_{i+1}^1 \sqsubset l_i^k}}$ nor by $\mathsf{aut}(\neg\varphi^{\phi \to l_{i+1}^1}, i, k)$. However, by the definition of the semantics, we have that $\mathcal{T}_1 \models^{\mathcal{T}^\star} \varphi_1$. Hence, by inductive hypothesis, we have that $\mathcal{T}_1'$ belongs to the language accepted by $\mathsf{aut}(\varphi_1, i+1, 1)$. This concludes the *only if* direction of the proof. The proof of the *if* direction is symmetric and is left to the reader.

Observe that, due to the projection operations, the size of the resulting automaton is nonelementary in number k of alternations of semilattice operators occurring in φ. Since solving both satisfiability and model checking reduces to testing emptiness of an APT, which requires time exponential in the size of the APT, we obtain the thesis. Notice that the product APT $\mathcal{A}_{\mathcal{K},\varphi}$ of $\mathcal{A}_\varphi$ and $\mathcal{K}$ is linear in the size of the KS. Hence, the PTIME data complexity of the model checking problem w.r.t. the size of the KS follows. □

Both for satisfiability and for model checking of DSTL* and, *a fortiori*, of STL*, a nonelementary lower bound can also be established, once again w.r.t. the alternation k of semilattice operators. The following theorem proves the result by means of a reduction from the satisfiability problem for the quantified propositional temporal

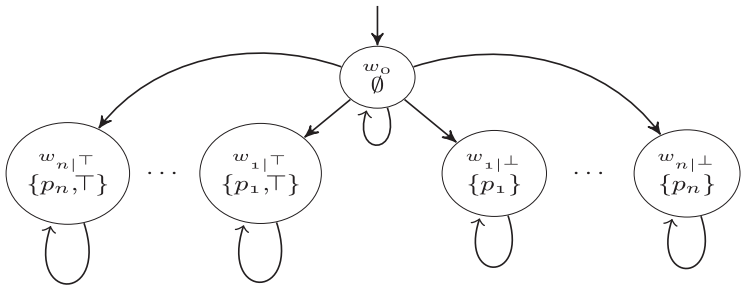

Fig. 17. STL*[KT] model-checking hardness ($\mathcal{K}_{MC}$).

logic QPTL, which is known to have a nonelementary space complexity [Sistla et al. 1987].

THEOREM 7.5 (DSTL*[KT] DECISION PROBLEM HARDNESS). DSTL*[KT] *satisfiability and model-checking problems are* k-EXPSPACE-HARD *w.r.t. the alternation* k *of semilattice operators in the* DSTL* *formula* ψ. *The latter problem is* PTIME-HARD *w.r.t. the size of the finite* KS $\mathcal{K} \in \mathrm{KS(AP)}$ *encoding the* KT *model* $\mathcal{K}^{\mathcal{U}}$.

PROOF OF THEOREM 7.5. To prove the hardness results w.r.t. the formula complexity for both the model checking and satisfiability of DSTL*, we provide a linear reduction from the QPTL satisfiability problem, whose complexity is k-EXPSPACE-HARD in the alternation $\mathsf{alt}(\psi)$ of the QPTL formula ψ.

Let us consider the model-checking problem first. 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 DSTL* formula $\widetilde{\psi}$ against the KS $\mathcal{K}_{MC}$ over $\mathrm{AP}^{\top}$ of Figure 17, 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 one 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, therefore, a single successor between $w_i{}^{\top}$ and $w_i{}^{\bot}$. The formula $\flat_i \triangleq \mathtt{EG}(\flat \wedge \neg \varphi_i)$, with $\varphi_i \triangleq \mathtt{EX}(p_i \wedge \top) \wedge \mathtt{EX}(p_i \wedge \neg\top)$, 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 has at least one of these successors. Therefore, the construct $\overline{\mathtt{EMax}}^{\downarrow}(\flat_i, \varphi)$ selects 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 $\overline{\mathtt{AMax}}^{\downarrow}(\flat_i, \varphi)$ can be used to encode the universal quantification $\forall p_i$. We can now define the following translation function $\widetilde{\cdot}$: QPTL $\rightarrow$ DSTL*:

—$\widetilde{\exists p_i \,.\, \psi'} \triangleq \overline{\mathtt{EMax}}^{\downarrow}(\flat_i, \widetilde{\psi'})$;
—$\widetilde{\forall p_i \,.\, \psi'} \triangleq \overline{\mathtt{AMax}}^{\downarrow}(\flat_i, \widetilde{\psi'})$;

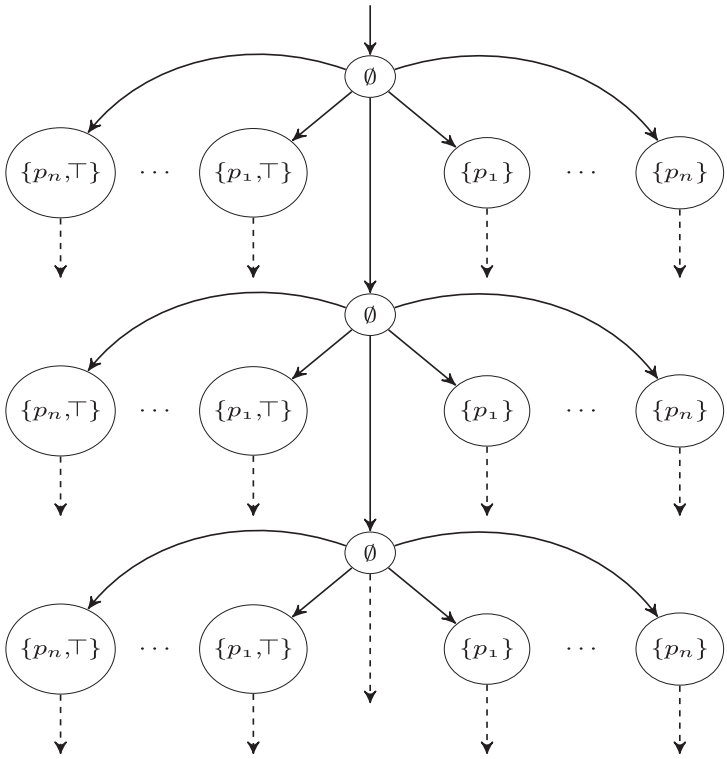

Fig. 18. STL⋆[KT] satisfiability hardness ($\mathcal{T}_{Sat}$).

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

It is easy to see that $|\widetilde{\psi}| = \mathrm{O}(|\psi|)$ and $\mathsf{alt}(\widetilde{\psi}) = \mathsf{alt}(\psi)$. Intuitively, this translation replaces each propositional quantification with a suitable choice of a subtree of $\mathcal{K}^U_{MC}$. Moreover, the verification of the truth value of a proposition p_i, at any given instant of time, is done by checking the existence of a successor of that 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 proves that ψ is satisfiable if and only if $\mathcal{K}^U_{MC} \models \widetilde{\psi}$. Hence, the thesis for the model-checking problem follows.

To obtain the reduction from QPTL satisfiability to STL⋆ satisfiability, we build a DSTL⋆ formula $\varphi_{\mathcal{K}} \wedge \widetilde{\psi}$, where $\varphi_{\mathcal{K}}$ is used to characterize the KTs of the same form of the tree structure $\mathcal{T}_{Sat}$ depicted in Figure 18. This structure corresponds to the unwindings of KSs that are equal to $\mathcal{K}_{MC}$ except, possibly, for the self-loops on the worlds w_i^{ℓ}, with $\ell \in \{\top, \bot\}$. First, we have to ensure that all the models of $\varphi_{\mathcal{K}}$ contain a spine globally satisfying $\flat$, whose worlds have two successors for each proposition p_i, one of which is labeled by the additional atomic proposition $\top$. This can be easily achieved by using the CTL formula $\varphi_{spn} \triangleq \flat \wedge \mathtt{AG}(\flat \rightarrow (\mathtt{EX}\flat \wedge \bigwedge_{i=1}^{n} \varphi_i))$, where φ_i is the same formula used earlier for the hardness of the model-checking problem. Moreover, to enforce that the flow of time is linear, we impose uniqueness of that spine by means of the DWSTL formula $\varphi_{min} \triangleq \mathbb{G}((\mathtt{AG}\flat) \rightarrow \mathtt{Min(t)})$. 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 if and only if $\varphi_{\mathcal{K}} \wedge \widetilde{\psi}$ is. Hence, the thesis for the satisfiability problem follows.

Finally, PTIME hardness of the model checking w.r.t. the size of the finite encoding of the model follows by a reduction from the reachability problem on And-Or graphs [Immerman 1981]. In particular, we check the formula $\overline{\texttt{EMin}}^{\downarrow}_{Or}(\texttt{t}, \overline{\texttt{AMin}}^{\downarrow}_{And}(\texttt{t}, \texttt{EF}\,p))$ 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 (Or, respectively) node is represented by a world labeled by the proposition *And* (*Or*, respectively). □

7.3. Complexity of Reasoning About Games in STL⋆

In light of the complexity results of the satisfiability and the model-checking problems for STL⋆ stated in the previous section, we can now assess the applicability of the logic to the game problems described in Section 5 and show that in most cases optimal complexities can be easily obtained. However, it is worth noticing that the worst-case complexities of the decision problems for STL⋆ under both KSs and KTs are symptomatic of the fact that this logic is able to encode properties that are already hard to verify themselves. In addition, the nonelementary results depend on the number of alternations of the semilattice operators, while all the game problems considered in Section 5 require a number of alternations of the operators, which is at most 2.

Consider, for example, the formula $\varphi_{MC}(\varphi) \triangleq \overline{\mathbb{G}}_1(\varphi)$, used to encode the module checking of a CTL⋆ formula φ. Note that in the formula $\varphi_{MC}(\varphi)$, the alternation of the semilattice operators is equal to 1. Therefore, by applying the STL⋆[KT] model-checking procedure of Theorem 7.4, we immediately obtain a 2EXPTIME solution algorithm, which is known to be optimal as proved in Kupferman et al. [2000]. When, in addition, φ is a CTL formula, the complexity drops down to EXPTIME, as the $+1$ in the tower of exponentials, which comes from the Vardi-Wolper construction of the automata for LTL (sub)formulas, can be avoided when only CTL formulas are involved. Similar considerations also apply to the STL⋆[KT] satisfiability procedure used to solve the synthesis problem. Indeed, the alternation of semilattice operators in the corresponding formula $\varphi_{RS}(\varphi)$ is equal to 1 in this case as well.

The number of alternations in the formulas $\varphi_{TG}(\psi)$, $\varphi_{CG}(\psi)$, and $\varphi_{NE}(\psi_1, \psi_2)$, used for the encodings of turn-based games, concurrent games, and existence of a Nash equilibrium under deterministic strategies, is 2. Therefore, a naive application of the STL⋆[KT] model-checking procedure would lead to a nonoptimal 3EXPTIME algorithm for all these problems. However, one can note that the nesting of the two relative minimality or maximality operators inside those formulas necessarily implies the verification of the LTL temporal goals on a minimal tree, that is, a tree having a single path, which corresponds to the single play resulting from the composition of the deterministic strategies of the agents. Therefore, the application of the exponential Vardi-Wolper construction for LTL can be avoided by replacing it with the equivalent linear alternating word automaton [Vardi 1996]. In this way, we can avoid the $+1$ in the tower of exponential of STL⋆[KT] complexity, obtaining an optimal 2EXPTIME algorithm [Alur et al. 2002; Gutierrez et al. 2014].

Consider now the encoding $\varphi_{NE}(\phi_1, \phi_2)$ used for the verification of the existence of a Nash equilibrium under nondeterministic strategies with branching-time goals ϕ_1 and ϕ_2. Once again, we have a formula with alternation 2, which implies a 3EXPTIME algorithm. Even though, to the best of our knowledge, no optimality result for this problem is known in the literature, this is likely to be the optimal complexity, due to the additional nondeterministic nature of the strategies involved.

Finally, consider the verification of ATL⋆ under memoryless strategies, *a.k.a.* imperfect recall. This problem was already proved to be PSPACE-COMPLETE in the size both of

the specification and of the model [Schobbens 2004; Vester 2013]. As shown in the paragraphs about turn-based and concurrent games in Section 5, STL* can easily encode ATL* formulas, and when interpreted on Kripke structures instead of their unwindings, the corresponding encodings check for the existence of memoryless strategies. Therefore, Theorem 7.3 establishes that the solution of the model-checking problem for STL*[KS] is indeed optimal w.r.t. its expressive power.

8. DISCUSSION

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 for 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. While this approach often allows for the development of efficient solutions to the specific problem at hand, it does not provide a temporal framework that allows one to uniformly reason about these problems.

In this article, we have shown that a suitable concept of substructure does play a crucial role when reasoning about open systems and games. In particular, the notion of substructure (of a Kripke structure) appears to be a viable alternative to the concept of strategy, commonly used in the game-related literature, to capture a wide range of game-related notions in temporal settings, that is, when temporal objectives are the main concern.

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* four semilattice operators used to select and reason about suitable substructures from the upper layer.

The temporal logic proposed is based on notions that are fairly standard in the context of formal verification. The semantic framework is indeed based solely on Kripke structures, while at the syntactic level temporal-logic-like operators suffice to provide the full expressive power of the logic. The semantics of these new modal operators is also defined in temporal logic terms and coincides with the standard temporal future operators "until" and "release" and the past operators "since" and "back to," interpreted on the lattice of substructures of a Kripke structure, instead of the Kripke structure itself. Therefore, STL* remains well within the realm of temporal logic, without the need to resort to external, and more complex, concepts like that of strategy and of game structure, typically introduced in logics for games, such as ATL* and Strategy Logic.

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 previously. In this sense, STL* serves as a purely temporal framework to reason about games in a general sense.

On a practical side, a wide range of techniques and tools have been devised by the formal verification community, which rely on temporal operators and on the notion of Kripke structure as their semantic model. An immediate consequence of the intrinsic temporal nature of the logic and of the results presented in this article is that well-known temporal verification techniques and tools can, at least in principle, be lifted with little effort to deal with STL* formulas. In addition, we have shown that in many cases of interest, the complexities of the resulting decision algorithms match the

optimal ones. This paves the way for a direct and easy way to uniformly extend such tools to reasoning about games with temporal objectives.

We investigated 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 becomes decidable in nonelementary time when interpreted over infinite regular trees. On the other hand, the model-checking problem is decidable under both interpretations, in PSPACE and in nonelementary time, respectively. The logic also enjoys favorable succinctness properties. In particular, the weakest fragment DWSTL, obtained by adding the weak downward operators $\mathbb{G}$ and $\mathbb{F}$ to CTL, is already exponentially more succinct than CTL.

Future work may proceed along various directions. Open problems concerning expressiveness and succinctness still remain to be settled. In particular, while we could prove that CTL* can be uniformly expressed in the weak fragment WSTL, it is not clear whether a translation exists into its downward fragment DWSTL or even into DSTL. Similarly, the succinctness relation between CTL* and WSTL is still open. We have reason to believe that WSTL is enough to express the whole logic STL*, while a similar relation between the downward fragments DSTL and DSTL* or between DWSTL in DWSTL* seems less likely to hold. We also think that both downward fragments DWSTL and DWSTL* can be decided in elementary time.

While, for the sake of space, we had to confine the analysis of STL* properties to its expressiveness and succinctness 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 ATL, Strategy Logic, and Sabotage Logic. Some of these analyses are currently underway. 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

C. Alos-Ferrer and A. B. Ania. 2001. Local equilibria in economic games. *Economics Letters* 70, 2 (2001), 165–173.

R. Alur, T. A. Henzinger, and O. Kupferman. 2002. Alternating-time temporal logic. *Journal of the ACM* 49, 5 (2002), 672–713.

M. Benerecetti, F. Mogavero, and A. Murano. 2013. Substructure temporal logic. In *Logic in Computer Science '13*. IEEE Computer Society, 368–377.

P. A. Bonatti, C. Lutz, A. Murano, and M. Y. Vardi. 2008. The complexity of enriched muCalculi. *Logical Methods in Computer Science* 4, 3 (2008), 1–27.

K. Chatterjee, T. A. Henzinger, and N. Piterman. 2007. Strategy logic. In *Concurrency Theory '07 (LNCS 4703)*. Springer, 59–73.

A. Church. 1963. Logic, arithmetics, and automata. In *International Congress of Mathematicians '62*. 23–35.

E. M. Clarke and E. A. Emerson. 1981. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs '81 (LNCS 131)*. Springer, 52–71.

E. M. Clarke, E. A. Emerson, and A. P. Sistla. 1986. Automatic verification of finite-state concurrent systems using temporal logic specifications. *Transactions on Programming Languages and Systems* 8, 2 (1986), 244–263.

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

R. Diestel. 2012. *Graph Theory* (4th ed.). Graduate Texts in Mathematics, Vol. 173. Springer.

E. A. Emerson and J. Y. Halpern. 1985. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Science* 30, 1 (1985), 1–24.

E. A. Emerson and J. Y. Halpern. 1986. "Sometimes" and "Not Never" revisited: On branching versus linear time. *Journal of the ACM* 33, 1 (1986), 151–178.

E. A. Emerson and C.-L. Lei. 1986. Temporal reasoning under generalized fairness constraints. In *Symposium on Theoretical Aspects of Computer Science '86 (LNCS 210)*. Springer, 267–278.

E. A. Emerson and A. P. Sistla. 1983. Deciding branching time logic: A triple exponential decision procedure for CTL*. In *Logic of Programs '83 (LNCS 164)*. Springer, 176–192.

N. Francez. 1986. *Fairness*. Springer.

J. Gerbrandy and W. Groeneveld. 1997. Reasoning about information change. *Journal of Logic, Language, and Information* 6, 2 (1997), 147–169.

J. Gutierrez, P. Harrenstein, and M. Wooldridge. 2014. Reasoning about equilibria in game-like concurrent systems. In *Knowledge Representation and Reasoning '14*. AAAI Press.

T. Hafer and W. Thomas. 1987. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *International Colloquium on Automata, Languages, and Programming '87 (LNCS 267)*. Springer, 269–279.

D. Harel. 1984. A simple highly undecidable domino problem. In *Logic and Computation Conference '84*. North-Holland, 177–194.

N. Immerman. 1981. Number of quantifiers is better than number of tape cells. *Journal of Computer and System Science* 22, 3 (1981), 384–406.

R. M. Keller. 1976. Formal verification of parallel programs. *Communication of the ACM* 19, 7 (1976), 371–384.

B. Khoussainov and A. Nerode. 2001. *Automata Theory and Its Applications*. Birkhauser.

S. A. Kripke. 1963. Semantical considerations on modal logic. *Acta Philosophica Fennica* 16 (1963), 83–94.

O. Kupferman, U. Sattler, and M. Y. Vardi. 2002. The complexity of the graded muCalculus. In *Conference on Automated Deduction '02 (LNCS 2392)*. Springer, 423–437.

O. Kupferman, M. Y. Vardi, and P. Wolper. 2000. An automata theoretic approach to branching-time model checking. *Journal of the ACM* 47, 2 (2000), 312–360.

O. Kupferman, M. Y. Vardi, and P. Wolper. 2001. Module checking. *Information and Computation* 164, 2 (2001), 322–344.

L. Lamport. 1980. "Sometime" is sometimes "Not Never": On the temporal logic of programs. In *Principles of Programming Languages '80*. Association for Computing Machinery, 174–185.

C. Löding and P. Rohde. 2003. Model checking and satisfiability for sabotage modal logic. In *Foundations of Software Technology and Theoretical Computer Science '03 (LNCS 2914)*. Springer, 302–313.

F. Mogavero and A. Murano. 2009. Branching-time temporal logics with minimal model quantifiers. In *Developments in Language Theory '09 (LNCS 5583)*. Springer, 396–409.

F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. 2012. What makes ATL* decidable? A decidable fragment of strategy logic. In *Concurrency Theory '12 (LNCS 7454)*. Springer, 193–208.

F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. 2014. Reasoning about strategies: On the model-checking problem. *Transactions on Computational Logic* 15, 4 (2014), 34:1–42.

F. Mogavero, A. Murano, and L. Sauro. 2013. On the boundary of behavioral strategies. In *Logic in Computer Science '13*. IEEE Computer Society, 263–272.

F. Mogavero, A. Murano, and L. Sauro. 2014. A behavioral hierarchy of strategy logic. In *Computational Logic in Multi-Agent Systems '14 (LNCS 8624)*. Springer, 148–165.

F. Mogavero, A. Murano, and M. Y. Vardi. 2010. Reasoning about strategies. In *Foundations of Software Technology and Theoretical Computer Science '10 (LIPIcs 8)*. Leibniz-Zentrum fuer Informatik, 133–144.

F. Moller and A. M. Rabinovich. 2003. Counting on CTL*: On the expressive power of monadic path logic. *Information and Computation* 184, 1 (2003), 147–159.

M. J. Osborne and A. Rubinstein. 1994. *A Course in Game Theory*. MIT Press.

J. A. Plaza. 2007. Logics of public communications. *Synthese* 158, 2 (2007), 165–179.

A. Pnueli. 1977. The temporal logic of programs. In *Foundation of Computer Science '77*. IEEE Computer Society, 46–57.

A. Pnueli. 1981. The temporal semantics of concurrent programs. *Theoretical Computer Science* 13 (1981), 45–60.

A. Pnueli and R. Rosner. 1989. On the synthesis of a reactive module. In *Principles of Programming Languages '89*. Association for Computing Machinery, 179–190.

J. P. Queille and J. Sifakis. 1981. Specification and verification of concurrent programs in Cesar. In *Symposium on Programming '81 (LNCS 137)*. Springer, 337–351.

M. O. Rabin. 1969. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematics Society* 141 (1969), 1–35.

R. Rosner. 1992. *Modular Synthesis of Reactive Systems*. Ph.D. Dissertation. Weizmann Institute of Science, Rehovot, Israel.

D. Sangiorgi. 2009. On the origins of bisimulation and coinduction. *Transactions on Programming Languages and Systems* 31, 4 (2009), 111–151.

P. Y. Schobbens. 2004. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science* 85, 2 (2004), 82–93.

A. P. Sistla, M. Y. Vardi, and P. Wolper. 1987. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science* 49 (1987), 217–237.

L. J. Stockmeyer and A. R. Meyer. 1973. Word problems requiring exponential time (preliminary report). In *Symposium on Theory of Computing '73*. Association for Computing Machinery, 1–9.

J. van Benthem. 2005. An essay on sabotage and obstruction. In *Mechanizing Mathematical Reasoning '05 (LNCS 2605)*. Springer, 268–276.

M. Y. Vardi. 1996. Why is modal logic so robustly decidable? In *Descriptive Complexity and Finite Models '96*. American Mathematical Society, 149–184.

S. Vester. 2013. Alternating-time temporal logic with finite-memory strategies. In *Games, Automata, Logics, and Formal Verification '13 (EPTCS 119)*. 194–207.

H. Wang. 1961. Proving theorems by pattern recognition II. *Bell System Technical Journal* 40 (1961), 1–41.

Received April 2014; revised December 2014; accepted April 2015