Reasoning about CTL* with Graded Path Modalities

Benjamin Aminof  
TU Wien, Austria  
Email: benj@forsyte.at

Aniello Murano  
Università di Napoli "Federico II", Italy  
Email: murano@na.infn.it

Sasha Rubin  
TU Wien, Austria  
Email: rubin@forsyte.at

Abstract—Graded path modalities count the number of paths satisfying a property, and generalize the existential (E) and universal (A) path modalities of CTL*. The resulting logic is denoted GCTL*, and is a very powerful logic since (as we show) it is equivalent to monadic path logic. We settle the complexity of the satisfiability problem of GCTL*, i.e., 2EXPSPACE-COMPLETE, and the complexity of the model checking problem of GCTL*, i.e., PSPACE-COMPLETE. The lower bounds already hold for CTL*, and so we supply the upper bounds. The significance of this work is two-fold: GCTL* is much more expressive than CTL* as it adds to it a form of quantitative reasoning, and this is done at no extra cost in computational complexity.

I. INTRODUCTION

Quantitative Verification. Temporal logics are the cornerstone of the field of formal verification [1]. Traditional Temporal Logics such as CTL* and LTL allow one to express Boolean properties of reactive systems, e.g., “it is always the case that if a print job is sent then it is eventually printed”. In recent years, a lot of attention has been given to the need to extend such specification formalisms by quantitative measures of function and robustness (see for example [2]–[4]). Unfortunately, though the need is great, these extensions many times come at a high cost. For example, while in the Boolean setting almost all problems can be addressed very elegantly by the automata-theoretic approach [5]–[8], which at its core translates questions about logical formulas to questions about automata accepting the models of these formulas; in the quantitative setting this approach naturally yields weighted automata [3], [9], [10]. Unfortunately, these are powerful machines for which much is undecidable [11], [12]. Other approaches usually encounter similar barriers as it turns out that some natural questions in the quantitative setting are undecidable, leading one to focus on restricted versions which are more tractable [3], [13], [14]. One possible way to extend classical temporal logics at a lower cost is by counting quantifiers, also known as graded modalities.

Temporal Logics with Graded Modalities. Graded world modalities were introduced in formal system specification and design as a useful extension of the standard existential and universal quantifiers in branching-time modal logics [15]–[18]. These modalities allow one to express properties such as “there exist at least n successors satisfying a formula” or “all but n successors satisfy a formula”. A prominent example is the extension of μ-calculus called Gμ-calculus [15], [17].

Despite its high expressive power, the μ-calculus (which extends modal logic by least and greatest fixpoint operators) is a low-level logic, making it “unfriendly” for users, who usually find it very hard to understand, let alone write, formulas involving even very modest nesting of fixed points. In contrast, CTL and CTL* are much more intuitive and user-friendly. An extension of CTL with graded path modalities called GCTL was defined in [19], [20]. Although there are several positive results about GCTL this logic suffers from similar limitations as CTL, i.e., it cannot nest successive temporal operators and so cannot express fairness constraints. This dramatically limits the usefulness of GCTL and so we turn instead to GCTL* in which one can naturally and comprehensibly express complex properties of systems. Although the syntax and semantics of GCTL* were defined and well justified in [20], only a very rudimentary study of it was made. In particular, the complexity of the satisfiability and model checking problem for this logic was never established, and remained open since its introduction in 2009. Instead, research has focused on the much simpler fragment of GCTL.

Our results. In this article we establish the exact complexity of the satisfiability and model checking problems for GCTL*. Our main result is that satisfiability is 2EXPSPACE-COMPLETE and that model checking is PSPACE-COMPLETE. Thus, in both cases, the problems for GCTL* are not harder than for CTL*. This is very good news indeed since, as we also show, GCTL* is expressively equivalent to monadic path logic, and is thus an extremely powerful, yet relatively friendly logic.

Finally, along the way, and in order to prove our main result, we prove that GCTL* has the bounded-degree tree model property, i.e., a satisfiable formula is satisfied in a tree whose branching degree is at most exponential in the size of the formula.

The importance of our results. Overall, we obtain that GCTL* has the following desirable attributes: a) it can naturally express properties of paths as well as count them, b) it is extremely expressive, and c) it has relatively low complexity of satisfiability. Below we briefly elaborate on these attributes in turn.

a) GCTL* can naturally express properties of paths as well as count them: For example, in a multitasking scheduling design, one can easily express the property “there are at least two ways to schedule the computation such that every request is eventually granted” by means of the GCTL* formula $E^{2}G(request \rightarrow (request \cup granted))$. Clearly this property cannot be expressed in CTL* nor in GCTL. As a more interesting example, consider a faulty program in which requests
are sometimes not granted due to the communication channel closing unexpectedly. The semantics of GCTL* (which is formally defined in Section II) is such that the formula
\[ \exists \exists \exists [F(request \land \neg F granted)] \]
implies that there are two incomparable sequences of operations, each causing the communication channel to close. Hence, this formula helps one to determine in one shot whether or not the faulty behaviour is the result of a single bug or multiple underlying problems.

It is important to note that the naive semantics for \( E \exists \exists \exists \psi \) which states that “there are at least \( n \) different paths satisfying \( \psi \)” while at first glance may seem natural and desirable, when examined more carefully turns out to be undesirable, since it very quickly leads to unnatural interpretations:\[1\]

b) GCTL* is extremely expressive: Not only does GCTL* extend CTL* (and thus, unlike CTL, it can reason about fairness), we prove that it is expressively equivalent, over trees, to Monadic Path Logic (MPL) which is Monadic Second-Order Logic (MSOL) with set quantification restricted to branches. More than a half-century of research has established MSOL as the cornerstone logic of formal verification.

c) GCTL* has relatively low complexity of satisfiability: Unfortunately, the complexity of satisfiability of MPL is non-elementary (this is already true for FOL). In sharp contrast, we prove that the complexity of satisfiability of GCTL* is 2ExpTIME, and thus is no harder than for CTL*.

**Technical Contributions.** The upper bounds are obtained by exploiting an automata-theoretic approach for branching-time logics, following [5]: combined with game theoretic reasoning at a crucial point. The automata theoretic approach is suitable because GCTL* turns out to have the tree model property. On the other hand, it is very hard to see how older techniques for deciding questions in logic (e.g. effective quantifier elimination, the tableaux method, the composition method) can be used to achieve optimal complexity results for GCTL*. Our proof is not just an easy adaptation of the classical decision procedure since considerable work has to be done relating GCTL* to a new model of automata, i.e., Graded Hesitant Tree Automata (GHTA). These automata, which we believe may be of interest in their own right, can work on finitely-branching trees (not just \( k \)-ary trees) and their transition relations can count up to a given number (standard alternating automata can only count up to 1).

In broad outline, our proof that satisfiability of GCTL* is in \( 2\text{ExpTIME} \) is in three steps. In the first step we reduce the satisfiability problem of GCTL* over arbitrary trees to the non-emptiness problem of an exponentially larger GHTA. The non-trivial part of this reduction is dealing with the graded path modalities \( E \exists \exists \exists \psi \), which generalises the case that deals with graded world modalities (as in \( G\mu \)-calculus \[17\]) as well as the case that the temporal operators in \( \psi \) are always coupled with path quantifiers (as in GCTL \[19\]).

In the second step, using a game theoretic approach, we show that GCTL* has the bounded-tree model property — in particular, a satisfiable formula is satisfied in a tree whose branching degree is at most exponential in the size of the formula. These two steps form the bulk of the work and can be summarised as follows:

\[ \text{GCTL}^* \xrightarrow{exp} \text{GHTA over } exp-ary \text{ trees} \]

In the third step we show that the emptiness problem for GHTA A over \( d \)-ary trees is decidable in time \( 2^{d(|Q|^2)} \), where \( Q \) is the state set of A.

**Comparison with naive approaches.** Although showing that satisfiability of GCTL* is decidable is not hard (for example, by reducing to MSOL), identifying the exact complexity is much harder. Indeed, there is no known satisfiability-preserving translation of GCTL* to a logic with decidable satisfiability over trees that would yield the optimal 2ExpTIME upper bound. We discuss two naive approaches along this line of attack. First, in this article we show a translation from GCTL* to MPL — unfortunately the complexity of satisfiability of MPL is non-elementary. Second, there is no reason to be optimistic that a translation from GCTL* to \( G\mu \)-calculus (whose satisfiability is \( \text{ExpTIME-COMPLETE} \)) would yield the optimal complexity since a) already the usual translation from CTL* to \( \mu \)-calculus does not yield optimal complexity \[21\], and b) the translation given in \[20\] from CTL to \( G\mu \)-calculus does not yield optimal complexity. Moreover, the usual translation from CTL* to \( \mu \)-calculus goes through automata which means that automata machinery for GCTL* would have to be developed, which is exactly what we do in this article. Thus, we do not further pursue this translation since we get optimal and elegant results using our automata directly.

**Comparison with Related Work.** Counting modalities were first introduced by Fine \[16\] under the name graded world modalities. A systematic treatment of the complexity of various graded modal logics followed, see for instance \[18\], \[22\]–\[26\]. The extension of \( \mu \)-calculus by graded world modalities was investigated in \[15\], \[17\]. Although these articles introduce automata that can count, our GHTA are more complicated since they have to deal with graded path modalities and not just graded world modalities. As discussed in \[20\], \( G\mu \)-calculus cannot succinctly reason about paths (or even grandchildren of a given node). The extension of CTL* by the ability to say “there exist at least \( n \) successors satisfying a certain formula”, called counting-CTL*, was studied in \[27\] using the composition method. It is unclear if that method, although elegant, can yield the complexity bounds we achieve. The first work to deal with graded path modalities is \[19\] that introduced GCTL, the extension of CTL by these modalities.

\[1\]The interested reader should consult \[20\] who devote countless pages to show that the semantics of GCTL* satisfies a variety of natural conditions that one would associate with the ability to count paths.

\[2\]Although GCTL does consider path modalities, as in \( G\mu \)-calculus, it only talks about state formulas. In comparison, GCTL* applies graded modalities to path formulas, which is a serious complication.
However, in GCTL the graded modalities are combined with state-formulas, while in GCTL* the graded modalities are combined with path-formulas, so it is impossible to inherit their techniques of [19], [20].

Graded path modalities over CTL were also studied in [28], but under a different semantics than GCTL. However, the semantics given in [28] is tailored for extending CTL, and it is unclear how one can extend their work to CTL*.

II. THE GCTL* TEMPORAL LOGIC

In this section we define the syntax and semantics of the main object of study, GCTL*, an extension of the classical branching-time temporal logic CTL* by graded path quantifiers of the form $E^{g}\varphi$. We follow the definition of GCTL* from [20] Section 3, but give a slightly simpler syntax. We assume the reader is familiar with the syntax and semantics of the classic temporal logics CTL*, LTL, and CTL. We first introduce some notation.

Models of GCTL* are Labelled Transition Systems (LTS) (alternatively, Kripke Structures). An LTS is a tuple $S = (\Sigma, S, E, \lambda)$, where $\Sigma$ is a set of labels, $S$ is a countable set of states, $E \subseteq S \times S$ is the transition relation, and $\lambda : S \rightarrow \Sigma$ is the labeling function. Typically, $\Sigma = 2^A$ where $A$ is a set of atomic propositions (also called atoms) and every $\lambda(s)$ is a finite subset of $A P$. If an atomic proposition $p$ is in $\lambda(s)$, then we say that $p$ holds in $s$. The degree of a state $s$ is the number of its successors, i.e., the cardinality of the set \{ $t \in S : (s, t) \in E$ \}. For simplicity of presentation, we assume that $E$ is total, i.e., that for every $q \in S$ there exists $q' \in S$ such that $(q, q') \in E$.

A path in a finite or infinite sequence $\pi_0\pi_1\ldots \in (S^*) \cup (S^\omega)$ such that $(\pi_i, \pi_{i+1}) \in E$ for all $i < |\pi|$. Here $|\pi|$, the length of the path $\pi$, is defined to be the cardinality of the sequence $\pi$, and in particular it is equal to $\omega$ if the path is infinite. Note that we count positions in the sequence starting with 0. If $0 \leq i < |\pi|$ then write $\pi_{i+1}$ for the suffix of $\pi$ starting at position $i$, namely the path $\pi_i\pi_{i+1}\ldots \in (S^*) \cup (S^\omega)$.

The set of (finite and infinite) paths in $S$ is written $pth(S)$, and the set of (finite and infinite) paths in $S$ that start in a given state $q \in S$ is written $pth(q, S)$. We write $\preceq$ for the prefix ordering on paths, and if $\pi \preceq \pi'$ say that $\pi'$ is an extension of $\pi$. Note that if $\pi$ is infinite then $\pi \preceq \pi'$ implies $\pi = \pi'$. If $X \subseteq pth(S, q)$ is a set of paths with the same starting element, define $\min(X)$ as the set of minimal elements in $X$ with respect to the prefix ordering $\preceq$.

A $\Sigma$-labeled tree $T$ is a pair $(T, V)$ where $T \subseteq N^*$ is a $\prec$-downward closed set of strings over $N$ (i.e., here $\prec$ is the prefix ordering on $N^*$), and $V : T \rightarrow \Sigma$ is a labeling, where $\Sigma$ is a set of labels. In particular the empty string $\epsilon$ is the root of the tree. For $t \in T$, we write $sons(t) \subset T$ for the set \{ $s \in T : t < s \land \pi_2z, t \prec z \prec s$ \}. The degree of a node $v$ is the cardinality of the set $sons(v)$. If every node $v$ of tree $T$ has finite degree, then we say that $T$ is finitely branching. If there is a $k \in N$ such that every node of $T$ has degree at most $k$, then say that $T$ is boundedly branching or has branching degree $k$. We implicitly view a tree $T = (T, V)$ as the LTS $(\Sigma, T, E, V)$ where $(t, s) \in E$ iff $s \in sons(t)$. As for LTSs, we assume for simplicity of presentation that trees are total (i.e., $sons(t) \neq \emptyset$ for every $t \in T$).

A. Syntax and Semantics of GCTL*

The semantics of GCTL* are defined for labelled transition systems $S$. The GCTL* formula $E^{g}\varphi$, for GCTL* path formula $\varphi$, can be read as “there exist at least $g$ (minimal $\psi$-conservative) paths”. Minimality was defined above, and so we now say, informally, what it means for a path to be $\psi$-conservative. An infinite path of $S$ is $\psi$-conservative if it satisfies $\varphi$, and a finite path of $S$ is $\psi$-conservative if all its (finite and infinite) extensions in $S$ satisfy $\varphi$. Note that this notion uses a semantics of GCTL* over finite paths, and thus the semantics of GCTL* needs to be defined for finite paths (as well as infinite paths). As in [20], we use the weak-version of semantics of temporal operators for finite paths (defined in [29]). Intuitively, temporal operators are interpreted pessimistically (with respect to possible extensions of the path), e.g., $(S, \pi) \models X\varphi$ if $|\pi| \geq 2$ and $(S, \pi_{\geq 2}) \models \varphi$. We now formally define the syntax and semantics of GCTL*.

Syntax of GCTL*: Fix a set of atoms AP. The GCTL* state $(\varphi)$ and path $(\psi)$ formulas are built inductively from AP using the following grammar:

$$
\varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid E^{g}\varphi
$$

$$
\psi ::= \varphi \mid \neg \varphi \mid \psi \lor \psi \mid X\psi \mid \psi U\psi \mid \psi R\psi
$$

In the first item, $p$ varies over AP and $g$ varies over $N$ (and thus, technically, there are infinitely many rules in this grammar). As usual, $X, U$ and $R$ are called temporal operators and $E^{g}$ (for $g \in N$) are called path modalities (also called path quantifiers). The class of GCTL* formulas is the set of state formulas generated by the above grammar. The simpler class of Graded CTL formulas (GCTL) is obtained by requiring each temporal operator occurring in a formula to be immediately preceded by a path quantifier, and each path quantifier to be immediately succeeded by a temporal operator, as in the classical definition of CTL. The class of LTL formulas is obtained as the path formulas in which no path quantifier appears. The degree of the quantifier $E^{g}$ is the number $g$. For a state formula $\varphi$, we define the degree $deg(\varphi)$ of $\varphi$ as the maximum natural number $g$ occurring among the degrees of all its path quantifiers. The length of a formula $\varphi$, denoted by $|\varphi|$, is defined inductively on the structure of $\varphi$ as usual, and using $|E^{g}\varphi|$ equal to $g + 1 + |\psi|$.

Semantics of GCTL*: Fix a LTS $S$. If $\varphi$ is a GCTL* state formula and $s \in S$, then define $(S, s) \models \varphi$ inductively:

1. $(S, s) \models p$, for $p \in AP$, iff $p \in \lambda(s)$.
2. $(S, s) \models \neg \varphi$, iff $(S, s) \not\models \varphi$.
3. $(S, s) \models E^{g}\varphi$, for $\psi$ a GCTL* path formula, iff the cardinality of the set $\min(Con(S, s, \psi))$ is at least $g$, where $Con(S, s, \psi) = \{ \pi \in pth(S, s) : \forall \pi' \in pth(S, s) : \pi \preceq \pi' \implies (S, \pi') \models \psi \}$. The paths in $Con(S, s, \psi)$ are called $\psi$-conservative (in $S$ starting at
If $\psi$ is a GCTL* path formula and $\pi = \pi_0, \pi_1, \ldots \in \text{pth}(S)$ is a finite or infinite path in $S$, define $(S, \pi) \models \psi$ inductively:

1. $(S, \pi) \models \varphi$, for $\varphi$ a state formula, iff $(S, \pi_0) \models \varphi$.
2. $(S, \pi) \not\models -\psi$ iff $(S, \pi) \not\models \psi$.
3. $(S, \pi) \models \psi_1 \lor \psi_2$ iff $(S, \pi) \models \psi_1$ or $(S, \pi) \models \psi_2$.
4. $(S, \pi) \models X\psi$ iff $|\pi| \geq 2$ and $(S, \pi_{|\pi|}) \models \psi$.
5. $(S, \pi) \models \psi_1 U \psi_2$ iff there exists $i$ with $0 \leq i < |\pi|$ such that $(S, \pi_{|\pi|-i}) \models \psi_2$ and for all $j$ with $0 \leq j < i$, $(S, \pi_{|\pi|-j}) \models \psi_1$.
6. $(S, \pi) \models \psi_1 R \psi_2$ iff for all $i$ with $0 \leq i < |\pi|$, either $(S, \pi_{|\pi|-i}) \models \psi_2$ or there exists $j$ with $0 \leq j < i$ such that $(S, \pi_{|\pi|-j}) \models \psi_1$; and moreover ii) if $\pi$ is finite then there is some $j < |\pi|$ such that $(S, \pi_{|\pi|-j}) \models \psi_1$.

If $\psi$ is an LTL formula, then we may write $\pi \models \psi$ instead of $(S, \pi) \models \psi$. This notation is justifiable since the truth of an LTL formula $\psi$ depends only on the path $\pi$ and not on the rest of the structure $S$.

Two state formulas $\phi, \phi'$ are equivalent if for all $s$ in $S$, we have $(S, s) \models \phi$ iff $(S, s) \models \phi'$. Similarly, two path formulas $\psi, \psi'$ are equivalent if for all $s$ and $\pi \in \text{pth}(S)$, we have that $(S, \pi) \models \psi$ if and only if $(S, \pi) \models \psi'$.

An LTS $S$ with a designated state $q \in S$ is a model of a GCTL* formula $\varphi$, sometimes denoted $S \models \varphi$, if $(S, q) \models \varphi$. For a labeled tree $T$, the designated node is by default the root, and thus, $T \models \varphi$ means that $(T, \epsilon) \models \varphi$ (recall that $\epsilon$ designates the root of $T$). A GCTL* formula $\varphi$ is satisfiable iff there exists a model of it.

**Example 1:** We unpack the meaning of the GCTL* formula from the introduction:

$$\exists \exists^2 [F(\text{request} \land -\text{F\text{grandted}})].$$

Let $\psi$ denote the path formula $F(\text{request} \land -\text{F\text{grandted}})$. First, a finite or infinite path satisfies $\psi$ if at some point $t$ the atom request holds, and from this point (to the end of the path if the path is finite) the atom granted does not hold. Second, a finite path is $\psi$-conservative if and only if at some point $t$ the atom request holds and the atom granted does not hold in any node of the subtree rooted at $t$ (an infinite path is $\psi$-conservative if and only if it satisfies $\psi$). Thus, $\exists \exists^2 \psi$ holds if and only if there exist two possibly finite paths, say $\pi^1$ and $\pi^2$, neither one a prefix of the other, both satisfying $\psi$ (i.e., $\pi^1$ has a request that is never granted on $\pi^1$), and such that if $\pi^1$ is finite then that path has request that is not granted in any possible future.

**Remark 1:** The additional operators present in the syntax of GCTL* in $[20]$, namely $\land, \lor, \neg, X, R, U$ and $A^g$ are dual to the operators $\land, \lor, R, U$ and $E^g$ respectively. This point is a little subtle for temporal operators, but easily follows from the fact that the tilled operators are defined like the un-tilled operators on infinite paths, but with ‘optimistic semantics’ on finite paths; e.g., $(S, \pi) \models X\psi$ if $|\pi| = 1$ or $(S, \pi_{|\pi|}) \models \psi$.

**Remark 2:** The GCTL* formula $\exists \exists^1 \psi$ means that there exists a $\psi$-conservative path, i.e., either there exists an infinite path satisfying $\psi$, or there exists a finite path $\pi$ satisfying $\psi$ such that every (finite and infinite) extension of $\pi$ satisfies $\psi$. Thus, under the assumption that $S$ is total, we see that $S \models \exists \exists^1 \psi$ if and only if $S$ has an infinite path satisfying $\psi$.

Thus, the fragment of GCTL* in which the degree $g$ of all quantifiers $\exists g$ is 1 coincides with the classical logic CTL* on total labeled transitions systems.

**Remark 3:** For a state formula $\phi$, the GCTL* formula $E^n \exists \phi$ expresses that there exist at least $n$ immediate successors of the current node satisfying $\phi$. This uses the following facts: i) a path of length 1 does not satisfy the path formula $X\phi$, and thus is not $X\phi$-conservative; ii) if $(S, \pi) \models X\phi$, then $\phi$ holds on the second state of $\pi$ (recall that $\phi$ is a state formula), and thus every extension $\pi'$, of the prefix of $\pi$ of length 2, satisfies $X\phi$, and thus $\pi$ is minimal $X\phi$-conservative iff $|\pi| = 2$. Hence, $(S, s) \models E^n \exists \phi$ if and only if there are at least $n$ minimal $X\phi$-conservative paths, which by the facts must all be of length 2, if there are at least $n$ immediate successors of $s$ satisfying $\phi$.

**B. Important Properties of GCTL**

We begin by explaining how to think of a GCTL* path formula $\psi$ over atoms AP as an LTL formula $\Psi$ over atoms which themselves are GCTL* state formulas. This idea, in the context of CTL*, can be found in $[5]$ and will be used in our subsequent proofs.

A formula $\varphi$ is a state sub-formula of $\psi$ if i) $\varphi$ is a state formula, and ii) $\varphi$ is a sub-formula of $\psi$. A formula $\varphi$ is a maximal state sub-formula of $\psi$ if $\varphi$ is a state sub-formula of $\psi$, and $\varphi$ is not a proper sub-formula of any other state sub-formula of $\psi$. Let $\text{max}(\psi) \equiv \{ \varphi \mid \varphi$ is a maximal state sub-formula of $\psi \}$, and let $\text{let}(\psi) = \bigcup_{\varphi \in \text{max}(\psi)} \{ \varphi, \neg \varphi \}$ be the set of all maximal state sub-formulas of $\psi$ and their negations. Every GCTL* path formula $\psi$ can be viewed as the formula $\Psi$ whose atoms are elements of $\text{let}(\psi)$. Note that $\Psi$ is an LTL formula.

**Proof:** Consider the formula $E^n Xp$. It is false in a tree whose root has exactly one successor $x$ satisfying $p$, but true in the bisimilar tree obtained by adding to the root another subtree which is identical to the one rooted at $x$.

**Lemma 3:** [20] GCTL* is invariant under unwinding.

**Proof Sketch:** The proof is standard: to treat $E^n g$ instead of $E$ use the standard $\prec$-preserving bijection between paths in
an LTS and paths in its unwinding, and note that the semantics of \( E \) involve reasoning about \( \preceq \). For completeness, full details are given in Appendix [B] or see [20].

The next Theorem shows that GCTL* is an extremely powerful logic. Indeed, it is equivalent, over trees, to Monadic Path Logic (MPL) which is MSO with quantification restricted to branches.

We briefly summarise the syntax and semantics of MPL [27], [30]. For a tree T write branches(T), the branches of T, for those finite or infinite paths of T, starting from the root, that are maximal (i.e., have no proper extensions in T). The syntax of MPL has logical symbols for the Boolean operations, first-order variables \( x, y, \cdots \), path variables \( X, Y, \cdots \), quantification over these variables, and non-logical symbols \( =, \prec, \epsilon \) and \( L_p \) for atoms \( p \in AP \). The semantics are defined for labeled trees \( T = (T, V) \) where \( V : T \to 2^{AP} \). The interpretation of variables \( x \) are over elements of \( T \), the interpretation of variables \( X \) are over branches of \( T \), the interpretation of \( = \) is the usual equality of variables, the interpretation of \( \prec \) is as the ancestor relation of \( T \), the interpretation of \( \epsilon \) is that node \( x \) is on the branch \( X \), and the interpretation of \( L_p \) is as the set of nodes \( t \) of \( T \) such that \( p \in V(t) \).

**Theorem 1:** GCTL* is equivalent, over trees, to Monadic Path Logic. In particular, GCTL* is more expressive than CTL*.

**Proof Sketch:** Introduce an auxiliary logic, called counting-CTL*, by adding to CTL* the state formulas \( D_n \phi \) (where \( \phi \) is a state formula and \( n \in \mathbb{N} \)) which are interpreted as saying that at least \( n \) elements of the current node satisfy \( \phi \). By Remark 3 GCTL* is at least as expressive, over trees, as Counting-CTL*. But the main result in [27] is that counting-CTL* is at least as expressive, over trees, as MPL. To finish, we sketch the relatively easy fact that MPL is at least as expressive, over trees, as GCTL*.

We show: (†) for every GCTL* state formula \( \phi \) there exists an MPL formula \( \hat{\phi}(x) \) such that for all trees \( T \), and all \( t \in T \): \( (T, t) \models \phi \) if and only if \( T \models \hat{\phi}(t) \).

In this proof we freely switch between viewing T as a tree and as an LTS. We start with some notation and three facts (whose proofs can be found in Appendix [C]).

We begin with some notation. For \( \pi \in pth(T) \) and \( a \in \pi \) write \( \pi_{a, \infty} \in pth(T, a) \) for the tail of \( \pi \) starting at \( a \). Also, for \( a, b \in T \) with \( a \leq \prec b \), write \( \pi_{a,[b]} \in pth(T, a) \) for the subpath of \( \pi \) starting at \( a \) and ending at \( b \).

Fact 1. For every LTL formula \( \Psi \) over atoms \( AP \) there is an MPL formula \( \Psi'([x, X]) \) (whose atomic relations are of the form \( L_p \) for \( p \in AP \)) such that for all trees \( T \) and all \( a \in T \) and all \( \pi \in branches(T) \) with \( a \in \pi \): \( T \models \Psi'(a, \pi) \) if and only if \( (T, \pi_{a, \infty}) \models \Psi \).

Fact 2. For every LTL formula \( \Psi \) over atoms \( AP \) there is an MPL formula \( \Psi'([x, y]) \) (whose atomic relations are of the form \( L_p \) for \( p \in AP \)) such that for all trees \( T \) and all \( a, b \in T \) with \( a \leq \prec b \): \( T \models \Psi'(a, b) \) if and only if \( (T, \pi_{a,b}) \models \Psi \).

Fact 3. For every LTL formula \( \Psi \) over atoms \( AP \) there is an MPL formula \( mincon_{\Psi}(x, X) \) (whose atomic relations are of the form \( L_p \) for \( p \in AP \)) such that for all trees \( T \), and all \( a \in T \) and all branches \( \pi \in T \) with \( a \in \pi \): \( T \models mincon_{\Psi}(a, \pi) \) if and only if the tail of \( \pi \) starting at \( a \) is minimal \( \Psi \)-conservative in \( (T, a) \). Similarly, there is a formula \( mincon_{\Psi}(x, y) \) such that for all trees \( T \) and all \( a, b \in T \) with \( a \leq \prec b \): \( T \models mincon_{\Psi}(a, b) \) if and only if the path between \( a \) and \( b \) is minimal \( \Psi \)-conservative in \( (T, a) \).

We now show how to inductively define the formula \( \hat{\phi}(x) \) in (†):

- If \( \phi \) is an atom, say \( p \), then \( \hat{\phi}(x) \) is defined as the unary predicate \( L_p(x) \).
- If \( \phi \) is of the form \( \neg \phi' \), then \( \hat{\phi} \) is defined as \( \neg \hat{\phi'} \); and similarly for \( \vee \) and \( \wedge \).
- If \( \phi \) is of the form \( E \psi \), then let \( \Psi \) be the LTL formula corresponding to \( \psi \) from Lemma 1 over atoms \( max(\psi) \). For each atom \( \theta \in max(\psi) \), let \( \hat{\theta}(x) \) be the corresponding MPL formula (which exists by induction) whose atoms are of the form \( L_\theta(x) \) for \( \theta \in max(\psi) \).

The formula \( \hat{\phi}(x) \) is defined as:

\[
\bigvee_{h \in [0, g]} \exists x_1, \cdots, x_h. \exists x_1, \cdots, x_{g-h}. \left[ \bigwedge_{i \in [1, h]} mincon_{\Psi}(x_i, x_i_{\lceil \theta(z)/\hat{\theta}(z) \rceil}) \wedge \right.
\]

\[
\left. \bigwedge_{i \in [1, g-h]} mincon_{\Psi}(x_i, x_i_{\lceil \theta(z)/\hat{\theta}(z) \rceil}) \right],
\]

where \( mincon_{\Psi}(x, x_{\lceil \theta(z)/\hat{\theta}(z) \rceil}) \) is the MPL formula \( mincon_{\Psi}(x, x_{\lceil \theta(z)/\hat{\theta}(z) \rceil}) \) in which every occurrence of a sub-formula of the form \( L_\theta(z) \) (for \( \theta \in max(\psi) \) and \( z \) a variable) is replaced by the formula \( \hat{\theta}(z) \).

This completes the proof sketch.

**Corollary 1:** GCTL* has the finitely-branching tree model property, i.e., if a GCTL* formula \( \varphi \) is satisfiable then it is satisfiable in a finitely-branching tree.

**Proof:** This follows immediately from Theorem 1 and the fact that the logic MPL has the finitely-branching tree model property.

Note that the Corollary does not state that there is a bound on the number of children of every node. However, we later prove that this is indeed the case, and that in all cases this bound is at most exponential in the size of the formula.

### III. GRADED HESITANT TREE AUTOMATA

In this section we define a new kind of automaton that is used in this work, namely Graded Hesitant Tree Automata. We also make use of classical automata, namely non-deterministic finite word automata (NFW) and non-deterministic Büchi word automata (NBW) (for definitions see for instance [30]), alternating parity tree automata (APTA) (for a definition see for instance [5]), and alternating hesitant tree automata (AHTA) (see for instance [5]).

We fix some notation. We write \( (\Sigma, Q, \delta, \delta') \) for NFWs and \((\Sigma, Q, q_0, \delta, F)\) for NBWs where \( \Sigma \) is the input alphabet.

\[3\text{We thank Igor Walukiewicz for pointing out to us that the fact that MPL has the finitely-branching tree model property, although folklore, immediately follows from a result in [31].}\]
is the set of states, \( q_0 \) is the initial state, \( \delta \subseteq Q \times \Sigma \times Q \) is the transition relation, \( G \subseteq Q \) is the set of accepting states and \( F \subseteq Q \) the set of final states.

Alternating tree automata (see [2]) are a generalization of nondeterministic tree automata. Intuitively, a nondeterministic tree automaton visiting a node of the input tree sends at most one copy of itself to each of the sons of the node, whereas an alternating automaton can send several copies of itself to the same son. Alternating hesitant tree automata (AHTA) first appeared in [3]. These are alternating tree automata with an acceptance condition that mixes both a Büchi condition and a co-Büchi condition. Furthermore, the automaton has a structural restriction imposed on its transitions that ensures that every path in a run-tree of the automaton can be uniquely assigned either the Büchi condition or the co-Büchi condition. Graded hesitant tree automata (GHTA), which we introduce below, generalise AHTA by a) they can work on finitely-branching trees (not just \( k \)-ary branching trees), and b) their transition relation allows the automaton to send multiple copies into the successors of the current node in a much more flexible way. Note that the combination of a Büchi and co-Büchi condition can be thought of as a special case of the parity condition with 3 colors, thus we could have defined Graded Parity Tree Automata instead.\(^4\)

However, we do not need the full power of the parity acceptance condition, and moreover, in order to achieve optimal complexity for model checking of GCTL\(^5\) we need to be able to decide membership of our automata in a space efficient way, which the Büchi and co-Büchi condition can be thought of as a special case of the parity condition with 3 colors, thus we could have defined Graded Parity Tree Automata instead.\(^4\)

Below we formally define AHTA and GHTA.

For a set \( X \), let \( B^+(X) \) be the set of positive Boolean formulas over \( X \), including the constants \texttt{true} and \texttt{false}. A set \( Y \subseteq X \) satisfies a formula \( \theta \in B^+(X) \), written \( Y \models \theta \), if assigning \texttt{true} to elements in \( Y \) and \texttt{false} to elements in \( X \setminus Y \) makes \( \theta \) true.

**A. Definition of AHTA**

An Alternating Hesitant Tree Automaton (AHTA) is a tuple

\[
A = (\Sigma, D, Q, q_0, \delta, (G, B), (\text{part}, \text{type}, \preceq))
\]

where \( \Sigma \) is a non-empty finite set of input letters; \( D \subseteq \mathbb{N} \) is a finite non-empty set of directions, \( Q \) is the non-empty finite set of states, \( q_0 \in Q \) is the initial state; the pair \((G, B)\) is \( 2^Q \times 2^Q \) the acceptance condition (we sometimes call the states in \( G \) good states and the states in \( B \) bad states); \( \delta : Q \times \Sigma \rightarrow B^+(D \times Q) \) is the alternating transition function; \( \text{part} \subseteq 2^Q \) is a partition of \( Q \), \( \text{type} : \text{part} \rightarrow \{\text{trans}, \text{exist}, \text{univ}\} \) is a function assigning the label transient, existential or universal to each element of the partition, and \( \preceq \subseteq 2^Q \times 2^Q \) is a partial order on \text{part}. Moreover, the transition function \( \delta \) is required to satisfy the following hesitancy condition: for every \( Q \in \text{part} \), every \( q \in Q \), and every \( \sigma \in \Sigma \):

(i) for every \( Q' \in \text{part} \) and \( q' \in Q' \), if \( q' \) occurs in \( \delta(q, \sigma) \) then \( Q' \preceq Q \).

(ii) if \( \text{type}(Q) \in \text{trans} \) then no state of \( Q \) occurs in the formula \( \delta(q, \sigma) \).

(iii) if \( \text{type}(Q) \in \text{exist} \) (resp., \( \text{type}(Q) \in \text{univ} \)) then there is at most one element of \( Q \) in each disjunct of the DNF (resp., conjunction of CNF) of \( \delta(q, \sigma) \).

An input tree (for AHTA) is a \( \Sigma \)-labeled tree \( T = \langle T, V \rangle \) with \( T \subseteq D^* \). Since \( D \) is finite, such trees have fixed finite branching degree. A run (or run tree) of an alternating tree automaton \( A \) on input tree \( T = \langle T, V \rangle \) is a \((T \times Q)\)-labeled tree \( (T_r, r) \), such that (i) \( r(\varepsilon) = (q_0, q_0) \) and (ii) for all \( y \in T_r \), with \( r(y) = (x, q) \), there exists a minimal set \( S \subseteq D \times Q \), such that \( S \models \delta(q, V(x)) \), and for every \((d, q') \in S \), it is the case that \( x \cdot d \) is a son of \( x \), and there exists a son \( y' \) of \( y \), so that \( r(y') = (x \cdot d, q') \).

Note that if \( \delta(q, V(x)) = \texttt{true} \) then \( S = \emptyset \) and the node \( y \) has no children; and if there is no \( S \) as required (for example if \( x \) does not have the required sons) then there is no run-tree with \( r(y) = (x, q) \). Observe that disjunctions in the transition relation are resolved into different run trees, while conjunctions give rise to different sons of a node in a run tree.

If \( v \) is a node of the run tree, and \( r(v) = (u, q) \), call \( u \) the location associated with \( v \), denoted \( \text{loc}(v) \), and call \( q \) the state associated with \( v \), denoted \( \text{state}(v) \).

We now discuss the acceptance condition. Fix a run tree \( \langle T_r, r \rangle \) and an infinite path \( \pi \) in the run tree. Say that the path \textit{visits} a state \( q \) at time \( i \) if \( \text{state}(\pi_i) = q \). The structural restriction (i) guarantees that the path \( \pi \) eventually gets trapped and visits only states in some element of the partition, i.e., there exists \( Q \in \text{part} \) such that from a certain time \( i \) on, \( \text{state}(\pi_j) \in Q \) for all \( j \geq i \). The condition (ii) ensures that this set is either existential or universal, i.e., \( \text{type}(Q) \in \{\text{exist}, \text{univ}\} \). Thus, we say that the path \( \pi \) gets trapped in an existential set if \( \text{type}(Q) = \text{exist} \), and otherwise we say that it gets trapped in a universal set. We can now define what it means for a path in a run tree to be accepting.

A path that gets trapped in an existential set is \textit{accepting} iff it visits some state of \( G \) infinitely often, and a path that gets trapped in a universal set is \textit{accepting} iff it visits every state of \( B \) finitely often. A run \( \langle T_r, r \rangle \) of an AHTA is \textit{accepting} iff all its infinite paths are accepting. An automaton \( A \) accepts an input tree \( \langle T, V \rangle \) iff there is an accepting run of \( A \) on \( (T, V) \). The \textit{language} of \( A \), denoted \( \mathcal{L}(A) \), is the set of \( \Sigma \)-labeled \( D \)-trees accepted by \( A \). We say that an automaton \( A \) is nonempty iff \( \mathcal{L}(A) \neq \emptyset \).

The \textit{membership problem} of AHTA is the following decision problem: given an AHTA \( A \) with direction set \( D \), and a finite LTS \( S \) in which the degree of each node is at most \( |D| \), decide whether or not \( A \) accepts \( S \). The \textit{depth} of the AHTA is the size of the longest chain in the partial order \( \preceq \). The size \( ||\delta|| \) of the transition function is the sum of the lengths of the formulas it contains. The size \( ||A|| \) of the AHTA \( A \) is \( |D| + |Q| + ||\delta|| \).

Note that the partition, partial order and type function are not \(\^3\)

\(^4\)Strictly speaking, GHTA generalise the symmetric variant of AHTA. That is, for every language accepted by an AHTA and that is closed under the operation of permuting siblings, there is a GHTA that accepts the same language.

\(^5\)Using the parity condition, our automata strictly generalise also the automata defined in [7, 19].
counted in the size of the automaton. The following is implicit in [5].

**Theorem 2:** The membership problem for AHTA can be solved in space $O(\partial \log^2(|S| \cdot |A|))$ where $\partial$ is the depth of $A$, $|A|$ is the size of $A$, and $S$ is the state set of $S$.

**B. Definition of GHTA**

We now introduce Graded Hesitant Tree Automata (GHTA). These can run on finitely-branching trees (not just trees of a fixed finite degree), and the transition function is graded, i.e., instead of a Boolean combination of direction-state pairs, it specifies a Boolean combination of distribution operations. There are two distribution operations: $\diamond(q_1,...,q_k)$ and its dual $\square(q_1,...,q_k)$. Intuitively, $\diamond(q_1,...,q_k)$ specifies that the automaton picks $k$ different sons $s_1,...,s_k$ of the current node and, for each $i \leq k$, sends a copy in state $q_i$ to son $s_i$. Note that the states $q_1,...,q_k$ are not necessarily all different.

Formally, a graded hesitate tree automaton (GHTA) is a tuple $A = (\Sigma, Q, q_0, \delta, (G, B), (\text{part}, \text{type}, \preceq))$ where all elements but $\delta$ are defined as for AHTA, and $\delta : Q \times \Sigma \rightarrow B^+([ \diamond \cup \square ]) = \text{a transition function that maps a state and an input letter to a positive Boolean combination of elements in } \diamond = \{ \diamond(q_1,...,q_k) \mid (q_1,...,q_k) \in Q^k, k \in \mathbb{N} \}$ and $\square = \{ \square(q_1,...,q_k) \mid (q_1,...,q_k) \in Q^k, k \in \mathbb{N} \}$.

We now show how to define the run of a GHTA $A$ on a $\Sigma$-labeled finitely-branching tree $T = \langle T, V \rangle$ by (locally) unfolding every $\diamond \cup \square$ in $\delta(q, V(t))$ into a formula in $B^+([d \times Q])$ where $d$ is the branching-degree of node $t$. For $k,d \in \mathbb{N}$, let $S(k,d)$ be the set of all ordered different $k$ elements in $[d]$, i.e., $(s_1,...,s_k) \in S(k,d)$ iff for every $i \in [k]$ we have that $s_i \in [d]$, and that if $i \neq j$ then $s_i \neq s_j$.

Observe that if $k > d$ then $S(k,d) = \emptyset$. For every $d \in \mathbb{N}$, define the function $\text{expand}_d : B^+([\diamond \cup \square]) \rightarrow B^+([d \times Q])$ that maps formula $\phi$ to the formula formed from $\phi$ by replacing every occurrence of a sub-formula of the form $\diamond(q_1,...,q_k)$ by the formula $\bigvee_{s_1,...,s_k \in S(k,d)} \land (s_i,q_i)$, and every occurrence of a sub-formula of the form $\square(q_1,...,q_k)$ by the formula $\bigwedge_{s_1,...,s_k \in S(k,d)} (s_i,q_i)$. Observe that if $k > d$ then $\diamond(q_1,...,q_k)$ becomes the constant formula false, and $\square(q_1,...,q_k)$ becomes the constant formula true.

The *run* of a GHTA $A$ is defined as for an alternating tree automaton, except that one uses $\text{expand}_d(\delta(q, V(x)))$ instead of $\delta(q, V(x))$ for nodes $x$ of $T$ of degree $n$.

Finally, the hesitancy condition defined above for AHTA are required to apply to the expanded transition function, i.e., insert the phrase “every $n \in \mathbb{N}$,” before the phrase “and every $\sigma \in \Sigma$,” and in items (i)-(iii) replace $\delta(q,\sigma)$ by $\text{expand}_d(\delta(q,\sigma))$. Acceptance is defined as for AHTA.

**Lemma 4:** The emptiness problem for GHTA $A$ over trees of branching degree at most $d$ is decidable in time $2^{O(d \cdot |Q|^2)}$, where $Q$ is the state set of $A$.

**Proof:** Given a GHTA $A$ with state set $Q$, convert it into an AHTA $A'$ with the same state space by using the function $\text{expand}_d$ defined above to transform its transition relation into a non-graded one. Note that this is possible since we assumed a bound $d$ on the branching degree of the input trees, and thus the transformation $\text{expand}_d$ can be used in advance. This construction takes time that is $2^{O(d \cdot |Q|^2)}$. Recall that AHTA are a special case of alternating parity tree automata (APTA) with 3 priorities. Now apply the fact that the emptiness problem for APTA with $p$ priorities over $d$-ary trees can be solved in time $2^{O(d \cdot |Q|^p)}$. $\square$

**IV. FROM GCTL* TO GRADED HESITANT AUTOMATA**

Elegant and optimal algorithms for solving the satisfiability and model-checking problems of CTL* were given using the automata theoretic approach for branching-time temporal logics [5]. Using this approach, one reduces satisfiability to the non-emptiness problem of a suitable tree automaton accepting all tree models of a given temporal logic formula. We follow the same approach here, by reducing the satisfiability problem of GCTL* to the non-emptiness problem of GHTA. By Corollary 1 a GCTL* formula is satisfiable (in some, possibly infinite, labeled transition system) iff it has a finitely branching (though possibly unboundedly branching) tree model, which exactly falls within the abilities of GHTA. Our main technical result states that every GCTL* formula can be compiled into an exponentially larger GHTA (the rest of this section provides a proof of this result):

**Theorem 3:** Given a GCTL* formula $\varphi$, one can build a GHTA $A_\varphi$ that accepts all the finitely-branching tree models of $\varphi$. Moreover, $A_\varphi$ has $2^{O(|\varphi| \cdot \deg(|\varphi|))}$ states, depth $O(|\varphi|)$, and transition function of size $2^{O(|\varphi| \cdot \deg(|\varphi|))}$.

An important observation that allows us to achieve an optimal construction is the following. Suppose that the formula $\mathbb{E}^2_s \varphi$ holds at some node $w$ of a tree. Then, by definition, there are at least $g$ different paths $\rho^1,\ldots,\rho^g$ in $\min(\text{Con}(S,w,\varphi))$. Look at any $g$ infinite extensions $\rho^1,\ldots,\rho^g$ of these paths in the tree, and note that by the definition of $\varphi$-conservativeness all these extensions must satisfy $\varphi$. Also observe that for every $i \neq j$, the fact that $\rho^i,\rho^j$ are different and minimal implies that the longest common prefix $\rho^{i,j}$ of $\rho^i$ and $\rho^j$ is not $\varphi$-conservative. As it turns out, the other direction is also true, i.e., if there are $g$ infinite paths $\rho^1,\ldots,\rho^g$ satisfying $\varphi$, such that for every $i \neq j$ the common prefix $\rho^{i,j}$ is not $\varphi$-conservative, then there are $g$ prefixes $\rho^{1,i},\ldots,\rho^{g,i}$ of $\rho^1,\ldots,\rho^g$ respectively, such that $\rho^{1,i},\ldots,\rho^{g,i} \in \min(\text{Con}(S,w,\varphi))$. Note that this allows us to reason about the cardinality of the set $\min(\text{Con}(S,w,\varphi))$, by considering only the infinite paths $\rho^1,\rho^g$ and their common prefixes, without actually looking at the minimal $\varphi$-conservative paths $\rho^{1,i},\ldots,\rho^{g,i}$ in $\min(\text{Con}(S,w,\varphi))$. In reality, we do not even have to directly consider the common prefixes $\rho^{i,j}$.

Indeed, since the property of being $\varphi$-conservative is upward closed (with respect to the prefix ordering of paths), showing that $\rho^{i,j}$ is not $\varphi$-conservative can be done by finding any extension of $\rho^{i,j}$ that is not $\varphi$-conservative. The following Proposition formally captures the above.

**Proposition 1:** Given a GCTL* path formula $\varphi$ and a $\Delta\mathcal{P}$-labeled tree $T = \langle T, V \rangle$, then $T \models \mathbb{E}^2_s \varphi$ iff there are $g$ distinct nodes $y_1,\ldots,y_g \in T$ (called breakpoints) such that for every $1 \leq i,j \leq g$ we have: (i) if $i \neq j$ then $y_i$ is not a descendant of $y_j$; (ii) the path from the root to the father $x_i$.
of \(y_i\) is not \(\psi\)-conservative; (iii) there is an infinite path \(\rho^i\) in \(T\), starting at the root and going through \(y_i\), such that \(\rho^i \models \psi\).

**Proof:** Assume first that \(T \models E^{g}\psi\), and let \(\rho^1, \ldots, \rho^g\) be \(g\) different paths in the set \(\min(\text{Con}(S, e, \psi))\). For \(1 \leq i \leq g\), let \(y_i\) be an arbitrarily chosen point on the path \(\rho^i\) satisfying, for every \(j \neq i\), that \(y_i\) is not on the path \(\rho^j\). Observe that such a point exists since, by minimality, \(\rho^i \not\sqsubseteq \rho^j\) for every \(j \neq i\). We thus have that property (i) in the statement of the lemma holds. Property (ii) holds by the minimality of \(\rho^i\). Indeed, the path from the root to the father of \(y_i\) is a proper prefix of \(\rho^i\), and is thus not in \(\text{Con}(S, e, \psi)\). By the definition of \(\psi\)-conservativeness, we have that every path \(\rho^i\) in \(T\) such that \(\rho^i \not\sqsubseteq \rho^j\) satisfies \(\psi\). Recall that we assume that trees are total, i.e., that they contain no leaves, and thus property (iii) holds by simply taking \(\rho^i\) to be any infinite extension of \(\rho^i\).

For the other direction, let \(y_1, \ldots, y_g \in T\), be breakpoints satisfying properties (i), (ii), (iii), and consider the paths \(\rho^1, \ldots, \rho^g\) through these breakpoints guaranteed by property (iii). For every \(1 \leq i \leq g\), let \(\rho^i\) be the shortest prefix of \(\rho^i\) such that \(\rho^i\) is \(\psi\)-conservative, and note that \(\rho^i \in \min(\text{Con}(S, e, \psi))\). The path \(\rho^i\) is well defined since \(\rho^i\) is infinite and satisfies \(\psi\) and thus, by definition, it is \(\psi\)-conservative. In order to prove that \(T \models E^{g}\psi\), it remains to show that for every \(i \neq j\) we have that \(\rho^i \not\sqsubseteq \rho^j\). To see that, observe that for every \(1 \leq i \leq g\), property (ii) together with the fact that the property of being \(\psi\)-conservative is upward closed (with respect to the prefix ordering \(\sqsubseteq\) of paths), imply that the path from the root to the father of \(y_i\) is a proper prefix of \(\rho^i\) and thus, \(\rho^i\) goes through \(y_i\). By property (i), if \(i \neq j\) then there is no path that goes through both \(y_i\) and \(y_j\). Combining the last two facts we get that if \(i \neq j\) then \(\rho^i \not\sqsubseteq \rho^j\), which completes the proof.

We are now in a position to describe our construction of a GHTA accepting all finitely branching tree models of a given GCTL* formula. Naturally, the main difficulty lies in handling the graded modalities. The basic intuition behind the way our construction handles formulas of the form \(\varphi = E^{g}\psi\) is the following. Given an input tree, the automaton \(A_\varphi\) for this formula has to find at least \(g\) minimal \(\psi\)-conservative paths. At its core, \(A_\varphi\) runs \(g\) pairs of copies of itself in parallel. The reason that these copies are not run independently is to ensure that the two members of each pair are kept coordinated, and that different pairs do not end up making the same guesses (and thus over-counting the number of minimal \(\psi\)-conservative paths). The task of each of the \(g\) pairs is to detect some minimal \(\psi\)-conservative path that contributes 1 to the count towards \(g\). This is done indirectly by using the characterization given by Proposition 1. Since this proposition requires checking whether or not certain paths satisfy \(\psi\), the automaton \(A_\varphi\) will access certain classic NBWs. And we begin by establishing the existence of these.

**Theorem 4:** Given an LTL formula \(\Psi\), there is an NBW \(A_\Psi\) and an NFW \(B_\Psi\) (both of size \(2^{|\varphi|}\)) accepting exactly all infinite and finite words that satisfy \(\Psi\), respectively.

**Proof:** The existence of \(A_\Psi\) is shown in [8], [32]. The existence of \(B_\Psi\) is proved by adapting the construction in [8] to handle finite paths (for details see Appendix [1]).

The following Lemma is now an easy consequence:

**Lemma 5:** Given an LTL formula \(\Psi\), there is an NBW \(A_\Psi^0\) (of size \(2^{|\varphi|}\)) such that \(A_\Psi^0\) accepts a word \(w\) iff \(w \models \Psi\), or \(u \models \Psi\) for a prefix \(u\) of \(w\). Moreover, \(A_\Psi^0\) has an accepting sink \(T\), such that if \(r_0, r_1, \ldots\) is an accepting run of \(A_\Psi^0\) on \(w\), and \(i \geq 0\) satisfies \(r_i \not\in T\), then a (finite or infinite) prefix \(u\) of \(w\), of length \(|u| > i\), satisfies \(\Psi\), and vice-versa (i.e., if \(r_i \not\in T\) on \(w\), there is an accepting run on \(w\) with \(r_i \not\in T\) for all \(i < |w|\)).

**Proof:** Consider the NBW \(A_\Psi = (\Sigma, Q_\Psi, q_0, \delta, G)\) and the NFW \(B_\Psi = (\Sigma, Q', q_0', \delta', F)\) from Theorem 4. Assume w.l.o.g. that \(Q, Q'\) are disjoint (and do not contain \(T, q_0\)) and construct from them a single NBW

\[A_\Psi^0 = (\Sigma, Q \cup Q' \cup \{\top\}, q_0, \delta, G \cup \{\top\}),\]

where \(\delta'\) is the union of \(\delta\) and \(\delta'\) as well as the transitions \((q_0, q, \sigma, q')\) for every \(\sigma\) and \(q\) such that \((q_0, q, \sigma, q') \in \delta\) or \((q_0', q, \sigma, q') \in \delta'\); \((\top, \sigma, \top)\) for every letter \(\sigma \in \Sigma\), and the transitions \((q, \sigma, \top)\) for every \((q, \sigma, q') \in \delta'\) for which \(q' \in F\).

I.e., by taking the union of \(A_\Psi\) and \(B_\Psi\), adding a new accepting sink state \(\top\), and matching any transition that goes to a final state of \(B_\Psi\) with a transition that goes to the accepting sink \(\top\). It is not hard to see that this construction yields the desired automaton.

We can now finish the intuitive description of the construction of the automaton \(A_\varphi\) associated with a formula \(\varphi = E^{g}\psi\). In essence, \(A_\varphi\) guesses the \(g\) descendants \(y_1, \ldots, y_g\) of the root of the input tree as given in Proposition 1. For every \(1 \leq i \leq g\), the automaton uses one copy of \(A_\Psi^0\) to verify that the path \(\pi\), from the root to the father of \(y_i\), is not \(\psi\)-conservative (by guessing some finite or infinite extension \(\pi \preceq \pi'\) of it such that \(\pi' \models \neg \psi\)), and one copy of \(A_\Psi\) to guess an infinite path \(\pi''\) from the root through \(y_i\) such that \(\pi'' \models \Psi\) (and is thus \(\psi\)-conservative).

The construction of GHTA \(A_\varphi\) for a GCTL* formula \(\varphi\).

We proceed by induction on the structure of \(\varphi\). Given a state sub-formula \(\phi\) of \(\varphi\) (possibly including \(\theta\)), for every formula \(\theta \in \max(\phi)\), let \(A_\theta = (\Sigma, Q_\theta, q_0, \delta, (G^0, G^0), \langle \text{part}\,^\theta, \text{type}\,^\theta, \preceq^\theta \rangle)\) be a GHTA accepting the finitely branching tree models of \(\theta\). We build the GHTA \(A_\varphi\) accepting all finitely branching tree models of \(\phi\) by suitably composing the automata of its maximal sub-formulas and their negations. Note that when composing these automata, we assume w.l.o.g. that the states of any occurrence of a constituent automaton of a sub-formula are disjoint from the states of any other occurrence of a constituent automaton (of the same or of a different sub-formula), as well as from any newly introduced states.

Formally, \(A_\varphi\) is constructed as follows:

If \(\phi = p \in AP\), then \(A_\phi = (\Sigma, q, q_0, \delta, (\emptyset, \emptyset), (\text{part}, \text{type}, \preceq))\) where \(\delta(q, \sigma) = \text{true}\) if \(p \in \sigma\) and false

For example, when building an automaton for \(\phi = \varphi_0 \lor \varphi_1\), in the degenerate case that \(\varphi_0 = \varphi_1\) then \(A_{\varphi_0}\) is taken to be a copy of \(A_{\varphi_1}\) with its states renamed to be disjoint from those of \(A_{\varphi_1}\). Also, the new state \(q_0\) may be renamed to avoid a collision with any of the other states.
otherwise, part = \{\{q\}\}, type(\{q\}) = \text{trans}, and ≤ is the empty relation.

If φ = ψ ∨ ϕ then Aφ is obtained by nondeterministically launching either Aψ or Aϕ. Formally, Aφ = \(\langle Σ, Q^φ, Ψ^φ, q_0, δ^φ, \langle \text{part, type, ≤} \rangle \rangle\), where for every \(i \in \{0, 1\}\), every \(σ ∈ Σ\), and every \(q ∈ Q^φ\) we have that:

\[ δ^φ(i, q, σ) = δ^φ_{\text{part}}(q, σ) \uplus δ^φ_{\text{trans}}(q_0, σ) \]

part = \{\{q_0\}\} ∪ \text{part}^φ ∪ \text{part}^\text{trans}; type(\{q_0\}) = \text{trans}, and for \(i = 1\) and \(q ∈ \text{part}^φ\), we have type(\(Q\)) = type^φ(\(Q\)); and ≤ is the union of the relations ≤\(^\text{part}\) as well as the inequalities \(≤\) for every \(q \in \text{part}\). In words, we maintain the partitioning of the states of Q^φ and Q^\text{trans} and their types and order, and add the transient set \{q_0\} making it larger then all other sets.

If φ = ¬ψ, then Aφ is obtained by dualizing the automaton Aψ. Formally, the dual of a GHTA A is the GHTA obtained by dualizing the transition function of A (i.e., switch ∨ and ∧, switch T and ⊥, and switch ∅ and ⊤), replacing the acceptance condition \(⟨B, G⟩\) with \(⟨B, G⟩\), and dualizing the types (i.e., every existential set becomes a universal set, and vice versa).

If φ = E^2gψ, then Aφ = \(\langle Σ, Q, q_0, δ, \langle B, G⟩, \langle \text{part, type, ≤} \rangle \rangle\) and its structure is detailed below. Observe that ψ is a path formula and, by Lemma 4 reasoning about ψ can be reduced to reasoning about the LTL formula Ψ whose atoms are elements of max(ψ). Let \(Σ' = 2^{\max(ψ)}\).

By Theorem 4 there is an NBW A^Ψ = \(\langle Σ', Q^+, \{q_0\}, δ^+, G^+\rangle\) accepting all infinite words in Σ^Ψ satisfying Ψ. By Lemma 3 there is an NBW A^Ψ = \(\langle Σ', Q^-, \{q_0\}, δ^-, G^-\rangle\) accepting all infinite words in Σ^Ψ that either satisfy ¬Ψ or have some prefix that satisfies ¬Ψ. Note that the states of these automata are denoted Q^+ and Q^-.

The set of states: \(Q = Q_1 ∪ Q_2\), where \(Q_1 = (Q^+ ∪ \{⊥\})^g \times (Q^- ∪ \{⊥\})^g \setminus \{⊥\}^2g\), and \(Q_2 = \bigcup_{q ∈ \text{max}(ψ)} Q^q\). The Q_1 states are used to run copies of A^Ψ and A^-Ψ in parallel. Every state in Q_1 is a vector of \(2g\) coordinates where coordinates 1,...,g contain states of A^Ψ, and coordinates \(g+1,...,2g\) contain states of A^-Ψ. In addition, each coordinate may contain the special symbol ⊥ indicating that it is disabled, as opposed to active. Note that we disallow the vector \(\{⊥\}^2g\) that has all coordinates disabled. States in Q_2 are all those from the automata A^-Ψ for every maximal state subformula of ψ, its negation, and used to run copies of these automata whenever A^-Ψ guesses that θ holds at a given node. We call the coordinates \(\{1...g\}\) the Ψ coordinates, and the ones \(\{g+1...2g\}\) the ¬Ψ coordinates. Also, for every \(1 ≤ i ≤ g\), we denote by Q^single = \{(q_1,...,q_2g) ∈ Q_1 | q_i ≠ ⊥, and for all \(j ≤ g\), \(j ≠ i\) then \(q_j = ⊥\)\} the set of all states in Q_1 in which the only active Ψ coordinate is i.

The initial state: q_0 = (q_1,...,q_2g) where for every \(1 ≤ i ≤ g\) we have that \(q_i = q_i^+\) and for every \(g+1 ≤ i ≤ 2g\) we have that \(q_i = q_i^+\).

The acceptance condition: B = \(\bigcup_{q ∈ \text{max}(ψ)} B^q\) and G = \(G' ∪ \{q ∈ \text{max}(ψ) \mid q ∈ G^+\}\), where G' = \{(q_1,...,q_2g) ∈ Q^single | q_i ∈ G^+\} is the set of all states in Q_1 in which the only active Ψ coordinate contains a good state.

The transition function: δ is defined for every \(σ ∈ Σ\), as follows:

- For every \(q ∈ Q_2\), let \(θ ∈ \text{max}(ψ)\) be such that \(q ∈ Q^θ\), and define \(δ(q, σ) = δ^θ(q, σ)\). Thus, for states in Q_2, we simply follow the rules of their respective automata.
- For every \(q ∈ Q_1\), we define \(δ(q, σ) := \bigvee_{σ' ∈ Σ} (J ∧ K ∧ L)\) where

\[
J = \bigvee_{X ∈ \text{Legal}(q, σ')} \diamond (X)
\]

\[
K = \bigwedge_{θ ∈ Σ'} δ^θ(q_0, σ)
\]

\[
L = \bigwedge_{θ ∈ Σ'} δ^-θ(q_0^θ, σ)
\]

where Legal(q, σ') is the set of all legal distributions of (q, σ'), and is defined later.

The meaning of this transition is as follows. The disjunction \(\bigvee_{σ' ∈ Σ'}\) corresponds to all possible guesses of the set of maximal subformulas of ψ that currently hold. Once a guess σ' is made, the copies of A^Ψ and A^-Ψ simulated by the states appearing in Legal(q, σ') proceed as if the input node was labeled by the letter σ'. The conjunction \(\bigwedge_{θ ∈ Σ'} δ^θ(q_0, σ)\) ensures that a guess is correct by launching a copy of A^Ψ for every subformula \(θ ∈ σ'\) that was guessed to hold, and a copy of A^-Ψ for every subformula \(θ\) guessed not to hold.

We now define what a legal distribution is. Intuitively, a legal distribution of (q, σ') is a sequence \(q^1,...,q^m\) of different states from Q_1 that “distribute” among them, without duplication, the coordinates active in q, while making sure that for every \(1 ≤ i ≤ g\) coordinate i (which simulates a copy of A^-Ψ) does not get separated from the coordinate i + g (which simulates its partner copy of A^Ψ) for as long as i is not the only active Ψ coordinate in the state. As expected, every active coordinate j, in any of the states q^1,...,q^m, follows from q_j by using the transitions available in the automaton it simulates: A^-Ψ if \(j ≤ g\), or A^Ψ if \(j > g\).

More formally, given a letter σ' ∈ Σ', and a state \(q = (q_1,...,q_2g) ∈ Q_1\) in which the active coordinates are \(i_1,...,i_k\), we say that a sequence \(X = q^1,...,q^m\) (for some \(m ≥ 1\)) of distinct states in Q_1 is a legal distribution of (q, σ') if the following conditions hold: (i) the coordinates active in the states q^1,...,q^m are exactly i_1,...,i_k, i.e., \{i_1,...,i_k\} = \(\bigcup_{i ∈ \{1,...,2g\} \mid \exists 1 ≤ l ≤ m \text{ s.t. } q^l_i ≠ ⊥\}\), (ii) if a coordinate i_j is active in some q^l ∈ X then it is not active in any other q'^l ∈ X; (iii) if \(1 ≤ i_j < i_l ≤ g\) are two active Ψ coordinates in some q^l ∈ X, then \(q_{i_j}^l + q_{i_j}^l + q_{i_l}^l + q_{i_l}^l ∈ Q^- \setminus \{⊥\}\), i.e., the coordinates \(i_j + g, i_l + g\) are also active in q^l and do not contain the accepting sink of A^-Ψ, (iv) if i_j is active in some q^l ∈ X then \(q_{i_l}^l + q_{i_l}^l + q_{i_j}^l + q_{i_j}^l ∈ δ^-\) if \(i_j ≤ g\), and \(q_{i_j}^l + q_{i_l}^l + q_{i_j}^l + q_{i_l}^l ∈ δ^-\) if \(i_j > g\). I.e., active Ψ coordinates evolve according to the transitions of A^Ψ, and active ¬Ψ coordinates according to the transitions of A^-Ψ.
The hesitant structure: Given a set of coordinates $I \subseteq \{1, \ldots, 2g\}$, let $Q_I \subseteq Q_1$ be the set of all vectors whose active coordinates are exactly $I$. We set $\text{type}(Q_I) = \text{exist}$, and set the partitioning part of $Q$ be the union of $\bigcup_{I \subseteq \{1, \ldots, 2g\}} \{Q_I\}$ and $\bigcup_{I \in \text{max}(\psi)} \text{part}$. For every $I \subseteq \{1, \ldots, 2g\}$, we have $Q_J \prec Q_I$ for every $J \subset I$, and $Q \preceq Q_I$ for every $Q \in \bigcup_{I \in \text{max}(\psi)} \text{part}$. Observe that if a transition $\delta(q, \sigma)$ from state $q \in Q_1$, on some letter $\sigma$, refers to another state $q' \in Q_I$ then $q$ was not split (since $q'$ has the same active coordinates as $q$), i.e., the $\bigvee$ in which $q'$ occurs is of the form $\bigvee(q')$. Hence, by the definition of $\delta(q, \sigma)$, there is no other $q'' \in Q_1$ that is conjuncted with $q'$ in $\text{expand}_d(\delta(q, \sigma))$ for any $d$, and thus $\delta(q, \sigma)$ respects the hesitancy constraint.

This completes the formal definition of the construction of the automaton $A_\psi$. The interested reader can find a formal proof of correctness (i.e., that the GHTA $A_\psi$ accepts exactly the finitely-branching tree models of $\psi$) in Appendix $E$. Together with the following straightforward analysis, Theorem $5$ is proved:

**Proposition 2:** The automaton $A_\psi$ is a GHTA, its depth is $O(|\psi|)$, it has $2^{O(|\psi| \cdot \text{deg}(\psi))}$ many states, and the size of its transition function is $2^{O(|\psi| \cdot \text{deg}(\psi))}$.

**Remark 4:** We end the section with two observations. First, the $2g$ copies of $A_\neg \psi$ and $A_\psi$ can not simply be launched from the root of the tree using a conjunction in the transition relation. The reason is that if this is done then there is no way to enforce property (i) of Proposition $1$. Second, a cursory look may suggest that different copies of $A_\neg \psi$ and $A_\psi$ that are active in the current vector may be merged. Unfortunately, this cannot be done since $A_\neg \psi$ and $A_\psi$ are nondeterministic, and thus, different copies of these automata must be able to make independent guesses in the present in order to accept different paths in the future.

V. GCTL* HAS THE BOUNDEDLY-BRANCHING TREE MODEL PROPERTY

The first fruit of the previous section is that we are now able to prove the following theorem:

**Theorem 5:** If a GCTL* formula $\psi$ is satisfiable then it has a tree model of branching degree at most $2^{O(|\psi| \cdot \text{deg}(\psi))}$.

**Proof:** Suppose $\psi$ is satisfiable. By Corollary $1$, $\psi$ has a finitely-branching tree model. Observe that, by Theorem $3$, we have that $|Q| = 2^{O(|\psi| \cdot \text{deg}(\psi))}$, where $Q$ is the state set of the automaton $A_\psi$ defined in that proof. Hence, it is enough to prove that every tree model of $\psi$ has a subtree of branching degree $|Q|^2$ that is also a model of $\psi$.

To prove this claim, we use the membership game $G_{T, A_\psi}$ of the input tree $T$ and the automaton $A_\psi$. The game is played by two players, *automaton and pathfinder*. Player automaton moves by resolving disjunctions in the transition relation of $A_\psi$, and is trying to show that $T$ is accepted by $A_\psi$. Player pathfinder moves by resolving conjunctions, and is trying to show that $T$ is not accepted by $A_\psi$. The game uses auxiliary tree structured arenas to resolve each transition of the automaton. This is a simple case of a *hierarchical parity game* [33]. As usual, player automaton has a winning strategy if and only if $T \models A_\psi$. By memoryless determinacy of parity games on infinite arenas, player automaton has a winning strategy if and only if he has a memoryless winning strategy. For a fixed memoryless strategy $str$, one can prove, by looking at the transition function of $A_\psi$, that every play consistent with $str$, and every node $t$ of the input tree $T$, only visits at most $|Q|^2$ sons of $t$, thus inducing a subtree which is the required boundedly-branching tree model.

Recall that the automaton $A_\psi$ is built by recursion on state subformulas (and their negations) of $\psi$. In a stage where a subformula $\phi$ is considered, an automaton is built which consists of some new states as well as the states of automata built from subformulas of $\phi$ and their negations. Let $\phi_q$ denote the stage at which state $q$ enters the construction for the first time. Note that every state of $Q$ enters the construction at some time, but some created states are not part of $Q$ (for example, no state of the automaton $A_\psi$ finds its way to the automaton for $(\neg p) \lor q$, because the latter uses states from the dual automaton $A_{\neg p}$).

**Definition of $G_{T, A_\psi}$ (for tree $T = (T, V)$ and formula $\psi$):** The arena consists of the main nodes $Q \times T$, two sink nodes $T, \bot$, as well as auxiliary nodes which are used to play the auxiliary games $aux(q, t)$ for $(q, t) \in Q \times T$. Play proceeds from a main node $(q, t)$ to the auxiliary arena $aux(q, t)$ (formally defined below) played on the parse tree of the formula defined by $\delta(q, \psi(T))$. The auxiliary arena $aux(q, t)$ is a finite tree, and when a play $\pi$ exits this arena it results in a node $\text{exit}_\pi(q, t)$ which is either a main node from $(Q \times \text{sons}(t)) \cup (Q \times \{t\})$ or a sink node. A play $\pi$ that visits $(q, t)$, proceeds, via some auxiliary nodes and main nodes of the form $Q \times \{t\}$, to a node $\text{next}_\pi(q, t) \in (Q \times \text{sons}(t)) \cup \{ot, T\}$.

The definition of the game $aux(q, t)$ depends on the form of $\phi_q$ and the definition of the transition $\delta(q, \psi(T))$.

- If $\phi_q = p$ for $p \in \text{AP}$, then the game $aux(q, t)$ immediately results in sink node $T$ if $p \in \psi(T)$ and in sink node $\bot$ otherwise.
- If $\phi_q = \varphi_0 \lor \varphi_1$, then in the game $aux(q, t)$ automaton chooses to exit either to main node $(q_0^{\varphi_0}, t)$ or to main node $(q_0^{\varphi_1}, t)$.
- If $\phi_q = \neg \varphi$, then the game $aux(q, t)$ immediately results in main node $(q', t)$ where $q'$ is the initial state of the dual automaton for $A_{\neg \varphi}$.
- If $\phi_q = E^g \psi$, then the game $aux(q, t)$ proceeds as follows: first player automaton picks $\sigma' \in \Sigma'$, and then pathfinder has three choices. Either she i) picks $\theta \in \sigma'$ and exits at main node $(q_0^{\sigma'}, t)$, or ii) she picks $\theta \notin \sigma'$ and exits at main node $(q_0^{\neg \sigma'}, t)$, or iii) she transfers play to automaton, in which case automaton picks a legal distribution, say $X = (q_1, \cdots, q_m) \in \text{Legal}(q', \sigma')$, and automaton also picks $m$-many different sons of $t$, say $(s_1, \cdots, s_m)$, and then pathfinder picks some $i \leq m$, and exits at main node $(q_i, s_i)$. To understand this game, recall from the construction of $A_\psi$ that the transition relation for
this case is defined as
\[
\underset{\sigma \in 2^\text{max}(\nu)}{\bigvee} \left[ (\forall X \in \text{Legal}(q,\sigma)) \Diamond (X) \right] \land \\
(\land \theta \in \sigma^* \delta^0(q_0, \sigma)) \land \left[ \land \theta \in \sigma^* \delta^1(q_0, \sigma) \right].
\]

The hesitant acceptance condition of $A_{\psi}$ can be easily translated into a parity condition with priorities $\{0,1,2\}$ (also, let sink node $T$ have priority 2, and sink node $\bot$ have priority 1). We say that player automaton wins a play if the largest priority occurring infinitely often is even.

This completes the description of the membership game $G_{T,A_{\psi}}$.

We now continue with the proof. Since $\vartheta$ is satisfiable, it is satisfiable by some finitely-branching tree $T$. Thus, fix a memoryless winning strategy $\text{str}$ for player automaton in the game $G_{T,A_{\psi}}$.

Lemma (\textcircled{1}). For every main node $(q,t)$ of $G_{T,A_{\psi}}$, there exists a set $Y(q,t) \subseteq Q \times \text{sons}(t)$ such that
- $|Y(q,t)| \leq |Q|$, and
- every play $\pi$ consistent with $\text{str}$ that exits the arena $\text{aux}(q,t)$ with $\text{exit}_{\pi}(q,t) \in Q \times \text{sons}(t)$ actually satisfies that $\text{exit}_{\pi}(q,t) \in Y(q,t)$.

The proof of this Lemma is by induction on $\phi$, and appears in Appendix \textcircled{C}.

We finish the proof of the theorem. Every play consistent with $\text{str}$ only visits, besides the main nodes $Q \times \{\text{root}\}$ (here root is the root vertex of the tree $T$), main nodes from $\cup_{t \in T} X(t)$ where $X(t) := \cup_{q \in Q} Y(q,t)$ (for $t \in T$). Note that for all $t \in T$, $|X(t)| \leq |Q|^2$. Define the subtree $T'$ of $T$ where the domain $T'$ consists of root and the elements in the set $\{t \in T : \exists q \in Q. t,q \in X(t)\}$. Note that every node in $T'$ has degree at most $|Q|^2$. The membership game $G_{T,A_{\psi}}$ is a subgame of $G_{T,A_{\psi}}$, and player automaton’s strategy $\text{str}$ is well defined on this subgame, and is winning. Thus $T' \models \vartheta$.

VI. COMPLEXITY OF SATISFIABILITY AND MODEL-CHECKING OF $GCTL^*$

We can now establish the main results of this paper:

Theorem 6: The satisfiability problem for $GCTL^*$ over LTS is $2\text{EXP-TIME-COMPLETE}$.

Proof: Theorems 3 \textcircled{5} and Lemma 4 \textcircled{3} yield that satisfiability of $GCTL^*$ is solvable in time double exponential in $O(|\vartheta| \cdot \text{deg}(\vartheta))$. The lower bound already holds for $\text{CTL}^*$ \textcircled{34}.

Theorem 7: Model checking $GCTL^*$ for finite LTSs is $\text{PSPACE-COMPLETE}$.

Proof: The lower bound follows from the corresponding lower-bound for $\text{CTL}^*$ \textcircled{34}. For the upper bound, given a $GCTL^*$ formula $\vartheta$, using Theorem 3 \textcircled{5}, construct the GHTA $A_{\vartheta}$, which has $2^{O(|\vartheta| \cdot \text{deg}(\vartheta))}$ states, and transition function of size $2^{O(|\vartheta| \cdot \text{deg}(\vartheta))}$, and which has depth $O(|\vartheta|)$. Let $d$ be the largest degree in $S$.

Claim: For every GHTA $A = \langle S, \delta, Q_0, \delta, \langle G,B \rangle, \langle \text{part}, \text{type}, \leq \rangle \rangle$ and $d \in \mathbb{N}$, there is an AHTA $A' = \langle S, Q_0, \delta, \langle G,B \rangle, \langle \text{part}, \text{type}, \leq \rangle \rangle$ that accepts the same $d$-ary trees as $A$, and such that $|\delta'| \leq |\delta| \cdot |Q|^d$.

Proof of Claim: As noted in the proof of Lemma 4 \textcircled{3} when only considering trees with a bounded branching degree $d$, one can transform a GHTA $A$ into an equivalent AHTA $A'$ by simply replacing $\delta(q,\sigma)$ by $\text{expand}_d(\delta(q,\sigma))$. Note that expanding the $\Diamond_{QS}$ and $\Box_{QS}$ only blows up the size of the transition relation by a multiplicative factor of $|Q|^d$, and leaves the state space unchanged. This completes the proof of the Claim.

Apply the Claim to get an equivalent AHTA $A'$ of size $|A'| = d + (2^{O(|\vartheta| \cdot \text{deg}(\vartheta))} \cdot |Q|^d + 2^{O(|\vartheta| \cdot \text{deg}(\vartheta))}) = 2^{O(|\vartheta| \cdot \text{deg}(\vartheta)) + d \cdot |\vartheta| \cdot \text{deg}(\vartheta))}$, and of depth $\vartheta = O(|\vartheta|)$. By Theorem 2 \textcircled{2} the membership problem of the AHTA $A'$ on LTS $S$ can be solved in space $O(|\vartheta|^2 \cdot |Q|^d \cdot |S|)$ which is polynomial in $|\vartheta|$ and $|S|$ (using $\text{deg}(\vartheta) \leq |\vartheta|$ and $d \leq |S|$).

VII. DISCUSSION

This work shows that $GCTL^*$ is an extremely expressive logic (it is equivalent to MPL and can express fairness and counting over paths) whose satisfiability and model-checking problems have the same complexity as that of $\text{CTL}^*$.

$GCTL^*$ was defined in \textcircled{19}. However, only the fragment $GCTL$ was studied. As the authors note in the conference version of that paper, their techniques, that worked for $GCTL$, do not work for $GCTL^*$. Moreover, they also suggested a line of attack that does not seem to work; indeed, it was left out of the journal version of their paper \textcircled{20}. Instead, our method is a careful combination of the automata-theoretic approach to branching-time logics \textcircled{5}, a novel characterization of the graded path modality (Proposition \textcircled{1}), and a boundedly-branching tree model property whose proof uses game-theoretic arguments (Theorem 5). Moreover, our technique immediately recovers the main results about $GCTL$ from \textcircled{19}, i.e., satisfiability for $GCTL$ is $\text{EXP-Time-Complete}$ and the model checking problem for $GCTL$ is in $\text{PTime}^\ast$. In other words, our technique suggests a powerful new way to deal with graded path modalities.

When investigating the complexity of a logic with a form of counting quantifiers, one must decide how the numbers in these quantifiers contribute to the length of a formula, i.e., to the input of a decision procedure. In this paper we assume that these numbers are coded in unary, rather than binary. There are a few reasons for this. First, the unary coding naturally appears in description and predicate logics \textcircled{35}. As pointed out in \textcircled{17}, this reflects the way in which many decision procedures for these logics work: they explicitly generate $n$ individuals for $\exists n$. Second, although the complexity of the binary case is sometimes the same as that of the unary case, the constructions

\footnote{Indeed, consider the construction in Theorem 3 \textcircled{5} of $A_{\vartheta}$ when $\vartheta$ it taken from the fragment $GCTL$ of $GCTL^*$, and in particular where it comes to a subformula $\phi$ of the form $\phi = \Diamond_{\exists \psi} \psi$. Since $\psi$ is either of the form $\rho \Box_{\psi}$ or $\rho \Box_{\psi}$, the number of new states added at this stage is a constant.}

is an extremely expressive logic (it is equivalent to MPL and can express fairness and counting over paths) whose satisfiability and model-checking problems have the same complexity as that of $\text{CTL}^*$.
are significantly more complicated, and are thus much harder to implement\cite{14, 20}. At any rate, because the basic idea is useful in some circumstances and corresponds more closely to our intuition of the length of a formula, we plan to investigate this in the future.

Now that the exact complexity of GCTL\(^\ast\) has been determined, we suggest several directions for future work. First, recall that logics extended with graded world modalities have been further enriched with backward-modalities and with nominals\cite{15}. We suggest that a similar direction should be taken for graded path modalities, and GCTL\(^\ast\) in particular. Second, recall that the graded \(\mu\)-calculus was used to solve questions (such as satisfiability) for the description logic \(\mu\text{-}\text{ALCQ}\)\cite{15}. Similarly, our techniques for GCTL\(^\ast\) might be useful for solving questions in \(\text{ALCQ}\) combined with temporal logic, such as for the graded extension of CTL\(^\ast\)\text{-}\text{ALC}\cite{50}. Finally, the GCTL model checking algorithm from\cite{20} has been implemented in the NuSMV tool providing the extra ability of solving questions in one shot while checking a system against a CTL formula. We are thus optimistic that existing CTL\(^\ast\) model-checkers can be fruitfully extended to handle GCTL\(^\ast\).

REFERENCES

\begin{enumerate}
\item M. Y. Vardi and L. Stockmeyer, "Lower bound in full (2EXPSPACE-hardness for CTL\(^\ast\)-SAT)." online, 1985, \url{http://www.cs.rice.edu/~vardi/papers/ctl_star_lower_bound.pdf}.
\end{enumerate}
**A. Proof of Lemma 7**

Let $\Psi$ be the LTL formula over atoms $\max(\psi)$ corresponding to $\psi$, as defined above. However for the duration of this inductive proof, instead of $\Psi$, write $\hat{\psi}$. Proceed by induction on $\psi$ (for all $S, \pi$).

The base case is that $\psi$ is a state formula. Then $\hat{\psi} \in \max(\psi)$. Thus $(S, \pi) \models \psi$ iff $(S, \pi_0) \models \psi$ iff (an atom) is in $L(q)$ iff $(S_\psi, \pi_0) \models \hat{\psi}$ iff $(S, \pi, \psi) \models \hat{\psi}$.

So suppose now that $\psi$ is a path formula that is not a state formula. We have the following cases.

1) Suppose $\psi = \neg \gamma$. Then $\max(\psi) = \max(\gamma)$, and so $\hat{\psi} = \neg \gamma$, and so

$$(S, \pi) \models \psi \iff (S, \pi) \not\models \gamma$$

$$(S, \pi) \models (S_{\neg \gamma}, \pi) \not\models \gamma$$

$$(S, \pi) \models \neg \gamma$$

Note that $f$ is $\preceq$-preserving, i.e., $\pi \preceq \pi'$ iff $f(\pi) \preceq f(\pi')$.

The inductive hypothesis for a GCTL* formula $\alpha$ says that for all LTSs $S, t \in S$, and $\pi \in \text{pth}(S, t)$:

- if $\alpha$ is a state formula then $(S, t) \models \alpha$ iff $(S', t) \models \alpha$.
- if $\alpha$ is a path formula then $(S, \pi) \models \alpha$ iff $(S', f(\pi)) \models \alpha$.

Suppose the inductive hypothesis holds for all proper sub-formulas of $\alpha$. Fix $S, t, \pi$. There are two main cases.

**Suppose $\alpha$ is a state formula.** There are three cases.

1) Suppose $\alpha$ is of the form $p$ for $p \in AP$. In this case we must prove that $(S, t) \models p$ iff $(S', t) \models p$, which is immediate from the definition of $X$.

2) The case that $\alpha = \neg \varphi_1$ or $\alpha = \varphi_1 \land \varphi_2$ for state formulas $\varphi_1$ is immediate from the semantics of these Boolean operations and the inductive hypothesis.

3) Suppose $\alpha$ is of the form $E^g \varphi$ for path formula $\varphi$.

For the first direction, suppose $(S, t) \models \alpha$, i.e., there are at least $g$ many minimal $\varphi$-conservative paths in $\text{pth}(S, t)$.

In other words, there exists distinct $\pi_1, \ldots, \pi_g \in \text{pth}(S, t)$ such that for every $i$, we have:

a) Every extension $\pi''$ in $\text{pth}(S, t)$ of $\pi_i$ satisfies $(S, \pi') \models \varphi$, for all $i$.

b) Every prefix $\tau$ of $\pi_i$ has an extension $\rho \in \text{pth}(S, t)$ that satisfies $(S, \rho) \not\models \varphi$.

Thus: $f(\pi_1), \ldots, f(\pi_g) \in \text{pth}(S', t)$ are distinct, and for every $i$ we have:

a) Every extension $\pi''$ in $\text{pth}(S', t)$ of $f(\pi_i)$ satisfies $(S', \pi') \models \varphi$. To see this note that $\pi' = f(\pi)$ for some $\pi \in \text{pth}(S, t)$, and so $f(\pi') \preceq f(\pi)$, and so $\pi_i \preceq \pi$, and so by $3a$ $(S, \pi) \models \varphi$, and so by induction $(S', f(\pi)) \models \varphi$.

b) Every prefix $\tau'$ of $f(\pi_i)$ has an extension $\rho' \in \text{pth}(S', t)$ that satisfies $(S', \rho') \not\models \varphi$ (use similar reasoning to the previous case).

Thus $(S', t) \models E^g \varphi$, and this completes the first direction. The other direction, i.e., that $(S', t) \models E^g \varphi$ implies $(S, t) \models E^g \varphi$ is done by simply reversing the argument for the first direction.

**Suppose $\alpha$ is a path formula, say $\psi$.** Let $\Psi$ be the LTL formula from Lemma 2. Then

$$(S, \pi) \models \psi \iff (S, \pi) \models \Psi$$

$$(S, \pi) \models \Psi$$

The first and fourth equivalences follow from Lemma 2. The second and third equivalences follow from inductive hypothesis applied to the maximal state sub-formulas of $\psi$ and the fact that $\pi \in S_\psi$ and $f(\pi) \in (S_\psi)\Psi$ and $(S_\Psi)\Psi$ induce the same infinite sequence of labels (and thus the paths agree on the LTL formula $\Psi$).

**C. Proof of missing Facts in proof of Theorem 1**

We prove Facts 2 and 3 (the proof of Fact 1 is similar).
Proof of Fact 2: Construct $\Psi''(x,y)$ by induction on the formula $\Psi$. If $\Psi$ is an atom, say $p \in \text{AP}$, then $\Psi''(x,y)$ is defined as $L_p(x)$. If $\Psi = \neg \Psi_1$ then $\Psi''(x,y)$ is defined as $\neg \Psi'_1(x,y)$. Similarly for the case that $\Psi$ is a disjunction. If $\Psi = \Psi_1 \cup \Psi_2$ then $\Psi''(x,y)$ is defined as $\exists z. x \leq z \leq y \land [\Psi'_1(x,y) \land \forall v. x \leq v \iff z \rightarrow \Psi'_2(x,v)]$. The cases $X$ and $R$ are similar to $U$. This completes the proof of Fact 2.

Proof of Fact 3:

- Let $\text{end}(X,z)$ denote the formula $z \in X \land \forall y \in X,y \leq z$ stating that $z$ is the last node of the branch $X$.
- Let $\text{finite}(X)$ denote the formula $\exists z. \text{end}(X,z)$ stating that branch $X$ is finite.
- We use the shorthand $\text{end}(X,z)$ for the unique value $\text{end}(X,z)$ if it exists.
- Let $\text{finmincon}_\psi(X,x)$ denote the formula
  \[ \text{finite}(X) \land (\forall y. \text{end}(X) \leq y \rightarrow \Psi''(x,y)) \land (\forall Y. \text{end}(X) \in Y \rightarrow \Psi'(x,Y)) \land (\forall z.x \leq z \rightarrow \text{end}(X) \rightarrow \neg \Psi''(x,z)) \]
  stating that the path $X$ starting at $x$ is finite and minimal $\Psi$-conservative.
- Let $\text{infincon}_\psi(X,x)$ denote the formula
  \[ \neg \text{finite}(X) \land \Psi''(x,x) \land (\forall z.x \leq z \in X \rightarrow \neg \Psi''(x,z)) \]
  stating that the path $X$ starting at $x$ is infinite and minimal $\Psi$-conservative.
- Finally, let $\text{mincon}_\psi(X,x)$ denote the formula $\text{finmincon}_\psi(X,x) \lor \text{infincon}_\psi(X,x)$ stating that the path $X$ starting at $x$ is minimal $\Psi$-conservative (irrespective of $X$ being finite or infinite).

Similarly, the formula $\text{mincon}_\psi(x,w)$ is defined as in $\text{finmincon}_\psi(X,x)$ but replacing $\text{end}(X)$ by $w$. This completes the proof of Fact 3.

D. Proof of Theorem 4

The existence of $A_\phi$ is shown in [8] Corollary 23. The existence of $E_\Psi$ is proved by a simple adaptation (given below) of the construction in [8] Theorem 22, yielding an alternating finite word automaton $E_\Psi$ of linear size) accepting all finite paths that satisfy $\Psi$, and then converting $E_\Psi$ to an equivalent NFW $B_\Psi$ (using [8] Proposition 16), of size $2^O(\Psi)$.

The alternating finite automaton $E_\Psi = \langle \Sigma, Q, q_0, \delta, F \rangle$ is constructed as follows: the input alphabet is $\Sigma = 2^{\text{AP}}$ where $\text{AP}$ is the set of atoms used by $\Psi$; the set of states $Q$ is the set of all sub-formulas of $\Psi$ and their negations (as usual $\neg \neg \phi$ is identified with $\phi$), as well as the special state $\text{cow}$ (indicating a guess that we reached the end of the input word); the initial state $q_0$ is $\Psi$; and the set of accepting states $F = \{ \text{cow} \}$. For a state $\phi$, and a set of atoms $a$, the transition function $\delta(\phi,a)$ is given by:

1) if $\phi = \text{cow}$, then $\delta(\phi,a) = \text{false}$;
2) if $\phi = p$ for $p \in \text{AP}$, then $\delta(\phi,a) = \text{true}$ if $p \in a$, and $\delta(\phi,a) = \text{false}$ otherwise;
3) if $\phi = \neg p$ for $p \in \text{AP}$, then $\delta(\phi,a) = \text{false}$ if $p \in a$, and $\delta(\phi,a) = \text{true}$ otherwise;
4) if $\phi = \varphi_1 \lor \varphi_2$, for $\lor \in \{ \land, \lor, \Rightarrow \}$, then $\delta(\varphi_1,a) \lor \delta(\varphi_2,a)$;
5) if $\phi = (\neg \varphi_1 \lor \varphi_2)$, for $\lor \in \{ \land, \lor, \Rightarrow \}$, then $\delta(\neg \varphi_1,a) \lor \delta(\varphi_2,a)$, where $\lor$ is the dual of $\land$, i.e., $\lor = \lor$;
6) if $\phi = X \theta$, then $\delta(\phi,a) = \theta$;
7) if $\phi = \neg X \theta$, then $\delta(\phi,a) = \text{cow} \lor \neg \theta$;
8) if $\phi = \varphi_1 \cup \varphi_2$, then $\delta(\varphi_1,a) \lor \delta(\varphi_2,a)$;
9) if $\phi = \neg \varphi_1 \cup \varphi_2$, then $\delta(\varphi_1,a) \lor \delta(\neg \varphi_2,a) \\
\lor (\delta(\neg \varphi_1,a) \lor \delta(\neg X \varphi_1 \cup \varphi_2,a))$;
10) if $\phi = \varphi_1 \cap \varphi_2$, then $\delta(\varphi_1,a) \land \delta(\varphi_2,a) \\
\lor (\delta(\varphi_1,a) \land \delta(\neg \varphi_2,a))$;
11) if $\phi = \neg \varphi_1 \cap \varphi_2$, then $\delta(\varphi_1,a) \land \delta(\neg \varphi_2,a) \\
\lor (\delta(\neg \varphi_1,a) \land \delta(\neg X \varphi_1 \cap \varphi_2,a))$.

By defining the transition relation rules for the cases of $X, U, R$ and their negations using one-step unfolding, the adaptation of the construction in [8] Theorem 22] to the finite words semantics addressed here is confined to the definition of the set of accepting states, and the transitions from $\text{cow}$ and $\text{cow}$.

E. Proof of correctness of the construction of $A_\phi$

We first give the formal definition of $A_\psi$ in the case that $\phi = \varphi_0 \lor \varphi_1$.

Formally, $A_\psi$ is

$$\langle \Sigma, \bigcup_{i=0}^{i=+1} \bigcup_{q=0}^{q=2^i} Q^q, \bigcup_{i=0}^{i=+1} G^q, B^q, \beta \rangle,$$

where $\beta = (\text{part}, \text{type}, \leq)$, and for every $i \in \{0,1\}$, every $\sigma \in \Sigma$, and every $q \in Q^q$, we have that: $\delta(q,\sigma) = \delta^{q_0}(q,\sigma)$, and $\delta(q_0,\sigma) = \delta^{q_0_0}(q_0,\sigma) \lor \delta^{q_1}(q_0,\sigma) \land \delta^{q_1_0}(q_0,\sigma)$; $\text{part} = \{ \{ q_0 \} \} \cup \text{part}^{q_0} \cup \text{part}^{q_1}$; $\text{type} = \text{trans}$, and for $i \in \{1,2\}$, and every $q \in \text{part}^q$, we have $\text{type}(Q) = \text{type}^q(Q)$; and $\leq$ is the union of the relations $\leq^{q_1} \leq^{q_2}$ as well as the inequalities $\leq^{q_0} \leq^{q_1}$ for every $Q \in \text{part}$. In words, we maintain the partitioning of the states of $Q^{q_0}$ and $Q^{q_1}$ their types and order, and add the disjoint union of the states $q_0$ making it larger then all other sets.

Before proving the correctness of the (entire) construction, we need some notation and a lemma. Let $A_\phi$ be the automaton constructed for a formula of the form $\phi = E^2 \psi$, and let $\langle T_r, \tau \rangle$ be a run of $A_\phi$ on an input tree $T = (T,V)$. Given a node $v \in T_r$, and its label $\tau(v) = (u,q)$, we say that $i$ is active (disabled) in $v$ iff $\text{state}(v) \in Q_1$ and $\text{state}(v) \neq \perp$ (state($v$), $\perp$).

**Lemma 6:** Let $A_\phi$ be the automaton constructed for $\phi = E^2 \psi$, and let $\langle T_r, \tau \rangle$ be a run of $A_\phi$ on a tree $T = (T,V)$. For every $1 \leq i \leq 2^g$, the set of nodes of $T_r$ in which $i$ is
active forms an infinite path $π^i$ from the root. Furthermore, if $(T_r, r)$ is an accepting run, then for $1 ≤ i ≤ g$, there is an $i_k > 1$ such that $state(π_{i_k-1})_{i_k+g} ∈ Q^− \setminus \{T\}$, and for every $l ≥ i_k$ the only active $Ψ$ coordinate in $state(π^i_l)$ is $i$.

Proof of Lemma 4.[1] We first prove that the set of nodes $I$ of $T_r$ in which $i$ is active is an infinite path from the root. The proof is by induction on the depth $κ$ of the nodes in $I$. The induction hypothesis is that there is exactly one node of depth $κ$ in $I$, and that for $κ ≥ 1$ it is a son of a node in $I$. For the base case $κ = 0$, the root $ε$ of $T_r$ satisfies $state(ε) = q_0$, and note that all coordinates, and in particular the $i$'th coordinate, are active in the initial state $q_0$.

For $κ > 1$, assume that the induction hypothesis holds. First, note that, by the definition of $δ$ (and in particular property (i) of a legal distribution), it must be that there is at least one node of depth $κ$ in $I$. Assume by way of contradiction that there are two such nodes $y ≠ y' ∈ I$ of depth $κ$. Observe that the transition function $δ$ is such that once a coordinate is disabled it can never become active again. Hence, the parents $x$ of $y$ and $y'$ both have the $i$’th coordinate active. Thus, by the induction hypothesis, $x = x'$, and $y$ and $y'$ are siblings. Let $r(x) = (s^x, q^x)$, $r(y) = (s^y, q^y)$, $r(y') = (s^{y'}, q^{y'})$, and let $d$ be the number of children of $s^x$. Note that the definition of a run tree implies that the formula $expand_d(δ(q^x, V(s^x)))$ contains a conjunction having both $(d, q^y)$ and $(d', q^y)$, where $d, d'$ are the directions in the input tree assigned to $s^y, s^{y'}$ respectively. In other words, the copy of $A_Φ$ in state $q^y$, that reads the input node $s^x$, launches (at least) two copies in parallel: one in state $q^y$, and the other in state $q''$ to $s''$. Recall that the transition $δ(q^x, V(s^x))$ is of the form $ϕ' ∈ Σ^∗((\forall X ∈ Legal(q, σ)\Diamond(X)) ∧ Ω_σ')$, where $Ω_σ'$ is a boolean formula that involves only states in $Q_2$ (and thus not $q'$ and $q''$). By the definition of a run, $(T_r, r)$ makes use of a single disjunct of any disjunction, and thus in this case, of one $\Diamond(X)$. It follows that both $q'$ and $q''$ appear in $X$, and thus, by the semantics of $\Diamond$, it must be that $q'$ and $q''$ were sent to two different sons of $s^x$, i.e., that $s' ≠ s''$. Recall that $i_k ≥ κ + 1$, and $j_k ≥ κ + 1$, and thus $y_j$ is either equal to $s'$ or is a descendant of it. Similarly, $y_j$ is either equal to $s''$ or is a descendant of it. We conclude that $y_i$ is not a descendant of $y_j$ as needed.

We now address condition (ii) of Proposition 1. Given $1 ≤ i ≤ g$, let $m = i + g$, and take the path $π_m$ guaranteed by Lemma 6. Consider the path $ρ_κ^m = loc(π^1)_κ · loc(π^m)_κ$ in $T$, of the nodes associated with $π_m$. For every $l ≥ 0$, let $σ_l' ∈ Σ^∗$ be the set of maximal state subformulas of $ϕ$ that hold in $ρ_κ^m$. Applying the induction hypothesis to all $θ ∈ max(ϕ)$, we can conclude that the only way $T_r$ can be accepting is if for every $0 ≤ l$ it resolves the outermost disjunction in $δ(state(π^i_l), V(ρ_κ^m)))$ by taking the disjunct

$$(\forall X ∈ Legal(state(π^i_l), σ_l')\Diamond(X)) ∧ (\land θ ∈ σ_l') δ(ϕ_0^d, σ) ∧ (\land θ ∈ σ_l') δ(ϕ_0^q, σ))$$

It is thus hard to see that since $T_r$ is an accepting run of $A_ϕ$ on $T$, then $r' = state(π^m_0)_m, state(π^m_1)_m, \ldots$ is an accepting run of $A_ϕ$ on the word $w = σ_l' · σ_l' · \ldots ∈ Σ^ω$. Note that Lemma 4 implies that $π^i_l = π^1_l = π^m_l$, and it also states that $state(π^i_l)_{i_g + g} ≠ T$. Hence, by Theorem 3 some (finite or infinite) prefix $u = σ_l' · σ_l' · \ldots$ of $w$ of length at least $i_k$ satisfies $¬Ψ$. By Lemma 1 it follows that the prefix $q$ of
\(\rho^m\), of the same length as \(u\), satisfies \(\neg \psi\). Observe that the length of \(\varphi\) implies that the path \(\rho_0^m \cdots \rho_{i-1}^m\), from the root of \(T\) to the father of \(y_i\), is a prefix (possibly not a proper prefix) of \(\varphi\), and is thus not \(\psi\)-conservative, as required by condition (ii).

Addressing condition (ii) of Proposition 5 follows in the footsteps of the reasoning used for condition (ii). Given \(1 \leq i \leq g\) and the path \(\pi^i\), the associated path \(\rho^i = \text{loc}(\pi^i_0) \cdot (\pi^i_1) \cdots \cdot (\pi^i_\ell)\) in \(T\) induces the infinite word \(w'' = \sigma_0^i \cdot \sigma_1^i \cdots \cdot \sigma_\ell^i\); who’s letters are the sets of maximal state subformulas of \(\psi\) that hold along the path \(\rho^i\). By the induction hypothesis, and since \(T_r\) is accepting, the run \(\text{state}(\pi^i_0), \text{state}(\pi^i_1), \ldots\) is an accepting run of \(A_\emptyset\) on the word \(w''\), and thus by Lemma 4 the path \(\rho^i\) satisfies \(\psi\). Since \(y_i\) lies on \(\rho^i\), condition (iii) of Proposition 1 is met, and we can conclude that \(T \models \varphi\).

For the other direction, let \(T = \langle T, V \rangle\) be such that \(T \models \varphi\). We have to show that \(A_\emptyset\) has an accepting run \(\langle T_r, r \rangle\) on \(T\). As before, the cases \(\psi = \varphi\) and \(\psi = \varphi \lor \varphi_1\), and \(\varphi = \neg \varphi\) follow easily from the definitions and the induction hypothesis. For the case \(\psi = \varphi\), there are \(g\) breakpoints \(y_1, \ldots, y_g \in T\), and rooted infinite paths \(\rho^1, \ldots, \rho^g\), such that for every \(1 \leq i \leq g\) we have that \(y_i = \rho^i_{i_k}\) for some \(i_k \geq 1\), and \(\rho^i \models \psi\); furthermore, the prefix \(\rho_0^1, \ldots, \rho_{i_k-1}^i\) is not \(\psi\)-conservative, and thus, it can be extended to an infinite path \(\rho^i_{i_k+1}\) such that \(\rho^i_0, \ldots, \rho^i_{i_k+1}\) satisfies \(\psi\). By the induction hypothesis (of the theorem), we have that \(\rho^i_0, \ldots, \rho^i_{i_k+1}\) satisfies \(\psi\). Since \(y_i\) lies on \(\rho^i\), condition (iii) of Proposition 1 is met, and we can conclude that \(T \models \varphi\).

By the construction of \(\langle T_r, r \rangle\), the subtree rooted at \(\pi_j\) is an accepting run of \(A_\emptyset\) on the subtree of \(T\) rooted at \(\text{loc}\(\pi_j\))\), and \(T\) is an accepting path. Consider now paths for which all states associated with the nodes of the path are in \(Q_1\). By Lemma 6, there are exactly 2\(g\) such paths \(\pi^1, \ldots, \pi^g\), and it is easy to see that by our construction of the run, for every \(1 \leq i \leq 2g\), we have that \(\text{state}(\pi^i_j), \text{state}(\pi^i_{j+1}), \ldots\) is exactly the run \(\rho^i\), and that for every \(j \geq i_n\) the only active \(\Psi\) coordinate in \(\text{state}(\pi^i_j)\) is \(i\). Hence, by the definition of the acceptance condition of \(A_\emptyset\), the path \(\pi^i\) is accepting. Which completes the correctness proof of the construction.

### F. Proof of Proposition 2

The depth is clearly \(O(|\varphi|)\).

We analyse the number of states, by cases. As usual, the cases \(\varphi = p\), \(\varphi = \varphi_0 \lor \varphi_1\), and \(\varphi = \neg \varphi\) follow easily from the definitions and the induction hypothesis. For the case \(\varphi = \varphi\), the states of the automaton are the union of \(Q_1\) and \(Q_2\).

The set \(Q_2\) uses states from each automaton \(A_\emptyset\) for every \(\varphi \in \text{max}(\varphi)\), and thus \(|Q_2| = \sum_{\varphi \in \text{max}(\varphi)} |A_\emptyset|\). But by induction \(|A_\emptyset| = 2^{O(|\varphi| \cdot \deg(\varphi))}\), and so \(|Q_2|\) is at most \(O(|\varphi|) \cdot 2^{O(|\varphi| \cdot \deg(\varphi))} = 2^{O(|\varphi| \cdot \deg(\varphi))}\).

The set \(Q_1\) uses a vector of \(g\) copies of \(A_\emptyset\) and \(g\) copies of \(A_\emptyset\). Thus \(|Q_1| = |A_\emptyset| \cdot |A_\emptyset| \cdots |A_\emptyset| = 2^{O(|\varphi|) \cdot g} = 2^{O(|\varphi| \cdot g)}\).
Thus, the number of states of \( A_\phi \) is \(|Q_1| + |Q_2|\) which is \(2^{O(|\phi| \cdot \deg(\phi))}\).

We treat the size of the transition function similarly. For the case \( \phi = E^\exists g \psi \) we add the lengths of the transitions leaving \( Q_1 \), and the lengths of the transitions leaving \( Q_2 \).

Say \( q \in Q_2 \) and \( \sigma \in \Sigma \), and let \( \theta \in \max(\psi) \) be such that \( q \in Q^\theta \). Then the length of formula \( \delta(q, \sigma) \) is, by induction, at most \( 2^{O(|\theta| \cdot \deg(\theta))} \). Thus the lengths of the transitions leaving \( Q_2 \) is at most \(|Q_2| \cdot 2^{O(|\psi| \cdot \deg(\psi))} = 2^{O(|\phi| \cdot \deg(\phi))}\).

Say \( q \in Q_1 \) and \( \sigma \in \Sigma \). By induction, for \( \theta \in \max(\psi) \), the number of transitions in \( A_\theta \) is \( 2^{O(|\theta| \cdot \deg(\theta))} \). Then the transition \( \delta(q, \sigma) \), defined as,

\[
\bigvee_{\sigma' \in 2^{\max(\psi)}} \left( \bigwedge_{X \in \text{Legal}(q, \sigma')} \left( \bigvee_{Y \in \text{Legal}(q, \sigma')} \bigwedge_{\theta' \in \max(\psi)} \left( \bigwedge_{\theta \in \sigma'} \delta^\theta(y, \sigma) \right) \right) \right)
\]

has length at most

\[
2^{O(|\psi|)} \left( \sum_{X \in \text{Legal}(q, \sigma')} \left| \bigvee (X) \right| + (\Sigma_{\theta \in \max(\psi)} 2^{O(|\theta| \cdot \deg(\theta))}) \right).
\]

Now \( \text{Legal}(q, \sigma') \) is the number of legal distributions of \( q \), which is at most the number of ways each of the \( 2g \) states can evolve times the number of ways to partition the components of \( q \) into \( k \) pieces (for some \( k \leq 2g \)). This is at most \( (2^{O(|\psi|)})^{2g} \cdot 2^{O(|\theta| \cdot \deg(\theta))} \), which is at most \( 2^{O(|\psi| \cdot \deg(\psi))} \).

Also, by writing \( X \) as a \( 2g \)-tuple of states of \( Q_1 \) in which each co-ordinate is also given a number (between 1 and \( k \)) indicating which element of the partition it is in, we get that \( \left| \bigvee (X) \right| \) is at most \( |Q|^2 \cdot k \cdot (2g)^2 = 2^{O(|\theta| \cdot \deg(\theta))} + O(\deg(\theta)^2) = 2^{O(|\theta| \cdot \deg(\theta))} \).

Thus the length is at most

\[
2^{O(|\psi|)} \left[ 2^{O(|\psi| \cdot \deg(\psi))} \cdot 2^{O(|\theta| \cdot \deg(\theta))} + O(|\psi|) \cdot 2^{O(|\theta| \cdot \deg(\theta))} \right]
\]

which is \( 2^{O(|\phi| \cdot \deg(\phi))} \).

G. Proof of Lemma (†) in proof of Theorem

We prove Lemma (†) by induction on \( \phi \): for every state \( q \) such that \( \phi_q = \phi \) for every \( t \in T \), there exists a set \( Y^\phi(q, t) \subseteq Q^\phi \times \text{sons}(t) \) of size at most \( |Q^\phi| \) such that every play \( \pi \) consistent with \( \text{str} \) that exits the auxiliary arena \( aux(q, t) \) in a node of the form \( Q^\phi \times \text{sons}(t) \) actually exits it in a node from \( Y^\phi(q, t) \).

To see why this gives the lemma, take \( (q, t) \in Q \times T \), and consider the induction at stage \( \phi_q \). Then every play \( \pi \) that exits the arena \( aux(q, t) \) with \( \text{exit}_\pi(q, t) \in Q \times \text{sons}(t) \) actually satisfies that \( \text{exit}_\pi(q, t) \in Y^{\phi_q}(q, t) \). But \( Y^{\phi_q}(q, t) \subseteq Q \times \text{sons}(t) \) and \( |Y^{\phi_q}(q, t)| \leq |Q^\phi| \leq |Q| \).

For the proof, suppose every proper subformula of \( \phi \) satisfies the inductive hypothesis. There are four cases:

1. \( \phi = p \) for some \( p \in \text{AP} \). Define \( Y^\phi(q, t) := \emptyset \), and note that the exit node of \( aux(q, t) \) is a sink node.
2. \( \phi = \varphi_1 \lor \varphi_2 \). Define \( Y^\phi(q, t) := \emptyset \), and note that the exit node of \( aux(q, t) \) is a main node of the form \( Q \times \{t\} \).
3. \( \phi = \neg \varphi \). Define \( Y^\phi(q, t) := \emptyset \), and note that the exit node of \( aux(q, t) \) is a main node of the form \( Q \times \{t\} \).
4. \( \phi = E^\exists g \psi \). The only way to exit \( aux(q, t) \) in a main node of the form \( Q \times \text{sons}(t) \) is via option iii) in the definition of \( aux(q, t) \) above; i.e., automaton picks \( \sigma' \), and then pathfinder transfers play to automaton, who then, according to \( \text{str} \), picks a legal distribution, say \( X = (q_1, \ldots, q_m) \) and corresponding sons of \( t \), say \( (s_1, \ldots, s_m) \). If there is no exit of the form \( (q_i, s_i) \), then \( \text{pathfinder} \) picks an exit of the form \( (q, s) \). Define \( Y^\phi(q, t) := \{ (q_1, s_1), \ldots, (q_m, s_m) \} \). Since \( m \leq |Q^\phi| \) (the components of the legal distribution \( X \) are distinct elements of \( Q^\phi \)), we have that \( |Y^\phi(q, t)| \leq |Q^\phi| \).