CTL* with Graded Path Modalities*

Benjamin Aminof[a,1], Aniello Murano[b,2], Sasha Rubin[b,3]

[a] *Technische Universität Wien, Austria*
[b] *Università degli Studi di Napoli "Federico II", Italy*

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 powerful logic since (as we show) it is equivalent, over trees, to monadic path logic. We establish the complexity of the satisfiability problem of GCTL*, i.e., 2ExpTime-Complete, the complexity of the model checking problem of GCTL*, i.e., PSpace-Complete, and the complexity of the realizability/synthesis problem of GCTL*, i.e., 2ExpTime-Complete. The lower bounds already hold for CTL*, and so we supply the upper bounds. The significance of this work is that 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.

Keywords: Path Quantifiers, Graded Temporal Logic, Satisfiability, Automata Theoretic Approach to Verification

BA: (Remove all comments.)

1. Introduction

Quantitative Verification and Graded Modalities. Temporal logics are the cornerstone of the field of formal verification. In recent years, much attention has been given to extending these by quantitative measures of function and robustness, e.g., [32]. Unfortunately, these extensions often require one to reason about weighted automata for which much is undecidable [21, 2, 3]. One

*A preliminary version of this work appeared in [6].

Email addresses: benj@forsyte.at (Benjamin Aminof), murano@unina.it (Aniello Murano), rubin@unina.it (Sasha Rubin)

[1] Benjamin Aminof was supported by the Austrian National Research Network S11403-N23 (RiSE) of the Austrian Science Fund (FWF) and by the Vienna Science and Technology Fund (WWTF) through grant ICT12-059.

[2] Aniello Murano is partially supported by the FP7 EU project 600958-SHERPA.

[3] Sasha Rubin was supported by a Marie Curie fellow of the Istituto Nazionale di Alta Matematica.

way to extend classical temporal logics at a lower cost is by counting quantifiers, known as graded modalities.

Graded world modalities were introduced in formal verification as a useful extension of the standard existential and universal quantifiers in branching-time modal logics [15, 30, 34, 45]. 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 $G\mu$-calculus [34, 15]. 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 to write) formulas involving even 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 [13, 14]. Thus one can express, in words, "there exist at least n paths" satisfying a formula.[4] 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 limits the usefulness of GCTL and so justifies studying GCTL* in which one can naturally express complex properties of systems. Although the syntax and semantics of GCTL* were defined and justified in [14], only a rudimentary study of it was made. In particular, the complexity of the satisfiability and model checking problem for this logic has never been established. Instead, research has focused on the much simpler GCTL fragment.

Our results. We establish the exact complexity of the satisfiability, model checking, and realizability/synthesis problems for GCTL* to be 2ExpTime-Complete, PSpace-Complete, and 2ExpTime-Complete respectively. Thus, in all cases, the problems for GCTL* are not harder than for CTL*. Along the way, 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, and we show that GCTL* is expressively equivalent, over trees, to monadic path logic.

The importance of our results. We obtain that GCTL* has the following desirable combination of attributes:
a) GCTL* *can naturally express properties of paths as well as count them.* For example, the formula $\mathsf{E}^{\geq 2}\mathsf{G}(\mathit{request} \rightarrow (\mathit{request}\,\mathsf{U}\,\mathit{granted}))$ says: "there are at least two ways to schedule the computation such that every request is eventually granted". This cannot be expressed in CTL* nor in GCTL.

The naive semantics for $\mathsf{E}^{\geq n}\psi$ counts two paths as different if they diverge. While at first glance this may seem natural, on closer examination it is undesirable and less informative. For example, consider a faulty program in which requests are sometimes not granted, and consider the formula $\mathsf{E}^{\geq 2}[\mathsf{F}(\mathit{request} \wedge \neg \mathsf{F}\mathit{granted})]$. This formula is true under the naive counting

[4] By writing "there exist at least n paths satisfying $\mathsf{X}\psi$" one can also express graded world modalities, see Remark 3.

even if the only fault occurs on a common prefix of two different paths. In contrast, in GCTL* (unlike the naive counting) the formula requires at least two *incomparable* sequences of operations each causing this faulty behaviour. In other words, in GCTL* the formula indicates whether the faulty behaviour is the result of multiple underlying problems, and is not confused by multiple paths that are extensions of a single faulty prefix. Furthermore, the naive counting very quickly leads to unnatural interpretations, as convincingly argued in [14].

This ability to easily count paths fits various application domains. For example, in databases there is a close relationship between model-checking CTL* and XML navigation (see [7, 11]). The logic GCTL* allows one to express quantitative requirements such as "the client has at least 5 items in last-month orders". More generally, graded operators are common in description logics, which are prominently used for formal reasoning in AI (e.g., knowledge querying, planning with redundancies).

b) GCTL* *is expressive.* Not only does GCTL* extend CTL* (and thus, in particular, 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) interpreted over trees but with set quantification restricted to branches.

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 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, 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. It is very hard to see how other techniques for deciding questions in logic (e.g. effective quantifier elimination, tableaux, composition) can be used to achieve optimal complexity results for GCTL*. We relate GCTL* to a new model of automata, i.e., *Graded Hesitant Tree Automata* (GHTA). These automata work on finitely-branching trees (not just k-ary trees) and their transition relations can count up to a given number (usual alternating automata only count up to 1).

Related Work. Counting modalities were first introduced by Fine [30] under the name *graded world modalities.* A systematic treatment of the complexity of various graded modal logics followed [45, 46, 20, 26, 37]. Number restrictions naturally occur in description logics [42, 33, 44, 17, 11, 18, 9, 8, 12]. The extension of μ-calculus by graded world modalities was investigated in [34, 15]. Although some of these articles introduce automata that can count, including [5, 39, 10, 25, 40, 38, 1], our GHTA are more complicated since they have to deal with graded path modalities and not just graded world modalities. The extension of CTL* by the ability to say "there exist at least n successors satisfying ψ", called counting-CTL*, was defined in [41], and its connection with Monadic Path Logic was studied using the composition method. It is unclear if that method, although elegant, can yield the complexity bounds we achieve (even for counting-CTL*). As shown in [14], $G\mu$-calculus cannot succinctly

reason about paths, or even grandchildren of a given node (the same is true for counting-CTL).

The first work to deal with graded path modalities introduced the extension of CTL* by these modalities, i.e., GCTL* [13]. However, only the fragment GCTL was studied since, as the authors note, their techniques do not work for GCTL*. Graded path modalities over CTL were also studied in [27, 28, 29]. There, the semantics of a formula of the form $\mathsf{E}^{\geq g}\psi$ is defined by counting if there are at least g different paths that serve as 'evidence' for ψ. The authors introduce two notions for capturing when paths should be counted as different: the first is a syntactic one that considers two paths to be different iff one is not a prefix of the other; and the second (introduced to alleviate some of the deficiencies of the first) modifies the first by requiring the paths to be edge-disjoint when the formula is of the form $\mathsf{E}^{\geq g}\mathsf{G}$ or of the form $\mathsf{E}^{\geq g}\mathsf{U}$. Unfortunately, their definition of a path being an evidence to a path formula ψ makes crucial use of the fact that CTL path formulas are of a very limited form (since nesting of temporal operators is not allowed). Hence, it is unclear if and how one can extend it to formulas in CTL*.

2. The temporal logic GCTL*

Let $\mathbb{N}$ denote the positive integers, and $[d] = \{1, 2, \ldots, d\}$ for $d \in \mathbb{N}$.

Transition Systems. An LTS (Labeled Transition System/Kripke structure) is a tuple $\mathsf{S} = \langle \Sigma, S, E, \lambda \rangle$, where Σ is a set of *labels*, S is a countable set of *states*, $E \subseteq S \times S$ is the *transition relation*, and $\lambda : S \to \Sigma$ is the *labeling function*. Typically, $\Sigma = 2^{\mathrm{AP}}$ where AP is a finite set of *atomic propositions*. The *degree* of a state s is the cardinality of the set $\{t \in S : (s,t) \in E\}$ of its successors. We assume that E is total, i.e., that every state has a successor.

Paths. A path in S is a finite or infinite sequence $\pi_0\pi_1 \cdots \in (S^*) \cup (S^\omega)$ such that $(\pi_i, \pi_{i+1}) \in E$ for all $i < |\pi|$ ($|\pi|$ is the *length* of π). Note that we count positions in a sequence starting with 0. Given a path π in an LTS, and some $0 \leq i < |\pi|$, then write $\pi_{\geq i}$ for the suffix of π starting at position i, namely the path $\pi_i\pi_{i+1} \cdots \in (S^*) \cup (S^\omega)$. The set of (finite and infinite) paths in S is written $\mathsf{pth}(\mathsf{S})$, and the set of (finite and infinite) paths in S that start in a given state $q \in S$ is written $\mathsf{pth}(\mathsf{S}, q)$.

Minimal paths. Let $\preceq$ be the prefix ordering on sequences. If $\pi \preceq \pi'$ we say that π' is an *extension* of π. For a set of paths X, denote by $\mathsf{min}(X)$ the minimal elements of X according to $\preceq$. For instance, if $X = \{s_1s_2s_3, s_1s_2s_3s_4, s_1s_7\}$ then $min(X) = \{s_1s_2s_3, s_1s_7\}$.

Trees. A Σ-*labeled tree* T is a pair $\langle T, V \rangle$ where $T \subseteq \mathbb{N}^*$ is a $\preceq$-downward closed set of strings over $\mathbb{N}$, and $V : T \to \Sigma$ is a labeling. We implicitly view a tree $\mathsf{T} = \langle T, V \rangle$ as the LTS $\langle \Sigma, T, E, V \rangle$ where $(t,s) \in E$ iff s is a son of t. If every node of a tree T has a finite degree then T is *finitely branching*. If every node has at most degree $k \in \mathbb{N}$, then T is *boundedly branching* or *has branching degree* k. We denote by ϵ the empty string (which is also the root of every tree). Given a tree T, for $t \in T$, we write $sons(t) \subset T$ for the set $\{s \in T : t \prec s \wedge \neg\exists z.t \prec z \prec s\}$.

Given an LTS $\mathsf{S} = \langle \Sigma, S, E, \lambda \rangle$, with $S = [d]$ for some $d \in \mathbb{N}$, define the *unwinding* of S to be the tree $\mathsf{T} = \langle T, V \rangle$ where $T = \{\epsilon\} \cup \{\pi \in \mathsf{pth}(\mathsf{S}, x) \mid (1, x) \in E \wedge |\pi| \in \mathbb{N}\}$ represents all paths in S starting in state 1; and V assigns to each element in T the label λ assigns to the last node on it, i.e., $V(\pi) = \lambda(\pi_{|\pi|-1})$ for $\pi \in T \setminus \{\epsilon\}$, and $V(\epsilon) = \lambda(1)$. Such an unwinding is called a *regular tree*.

2.1. Syntax and Semantics of GCTL*

GCTL* extends CTL* by *graded path quantifiers* of the form $\mathsf{E}^{\geq g}$. We follow the definition of GCTL* from [14], but give a slightly simpler syntax. We assume that the reader is familiar with the logics CTL*, LTL, and CTL (see [48, 35]).

The semantics of GCTL* is defined for an LTS S. Intuitively, the GCTL* formula $\mathsf{E}^{\geq g}\psi$, for GCTL* path formula ψ, can be read as *"there exist at least g (minimal ψ-conservative) paths"*. Minimality was defined above, so we now informally describe what it means for a path to be ψ-conservative (the formal definition appears in the semantics, below). An infinite path of S is ψ-conservative if it satisfies ψ, and a finite path of S is ψ-conservative if all its (finite and infinite) extensions in S satisfy ψ. 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 [14], we use the weak-version of semantics of temporal operators for finite paths (defined in [22]). Intuitively, temporal operators are interpreted pessimistically (with respect to possible extensions of the path), e.g., $(\mathsf{S}, \pi) \models \mathsf{X}\psi$ iff $|\pi| \geq 2$ and $(\mathsf{S}, \pi_{\geq 1}) \models \psi$.

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

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E}^{\geq g}\psi$$

and

$$\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}\psi \mid \psi\mathsf{U}\psi \mid \psi\mathsf{R}\psi.$$

In the first part, p varies over AP and g varies over $\mathbb{N}$ (and thus, technically, there are infinitely many rules in this grammar). As usual, X, U and R are called *temporal operators* and $\mathsf{E}^{\geq g}$ (for $g \in \mathbb{N}$) are called *path modalities* (also called *path quantifiers*). We write $\mathsf{F}\varphi$ instead of $\mathbf{true}\mathsf{U}\varphi$, and $\mathsf{G}\varphi$ instead of $\mathbf{false}\mathsf{R}\varphi$. The class of GCTL* *formulas* is the set of state formulas generated by the above grammar.

The *degree* of the quantifier $\mathsf{E}^{\geq g}$ is the number g. The *degree* $\mathsf{deg}(\varphi)$, of a state formula φ, is the maximum of the degrees of the quantifiers appearing in φ. The *length* $|\varphi|$, of a formula φ, is defined inductively on the structure of φ as usual, and using $|\mathsf{E}^{\geq g}\psi|$ equal to $g + 1 + |\psi|$ (i.e., g is coded in unary).

We now describe two syntactic fragments. The class of *Graded* CTL *formulas* (GCTL) is obtained by requiring each temporal operator to be immediately preceded by a path quantifier. The class of *linear temporal logic* (LTL) formulas consists of the path formulas in which no path quantifier appears.

Semantics of GCTL*. Given an LTS S and a state $s \in \mathsf{S}$, the definition of $(\mathsf{S}, s) \models \varphi$ is by induction on the structure of φ, exactly as for CTL*, with the only change concerning the new path quantifier $\mathsf{E}^{\geq g}$.

Fix an LTS S. If φ is a GCTL* state formula and $s \in S$, then define $(\mathsf{S}, s) \models \varphi$ inductively:

- $(\mathsf{S}, s) \models p$, for $p \in \mathrm{AP}$, iff $p \in \lambda(s)$.
- $(\mathsf{S}, s) \models \neg\varphi$ iff $(\mathsf{S}, s) \not\models \varphi$.
- $(\mathsf{S}, s) \models \mathsf{E}^{\geq g}\psi$, for ψ a GCTL* path formula, iff the cardinality of the set $\mathsf{min}(Con(\mathsf{S}, s, \psi))$ is at least g, where $Con(\mathsf{S}, s, \psi)$ is defined by

  $$\{\pi \in \mathsf{pth}(\mathsf{S}, s) \mid \forall \pi' \in \mathsf{pth}(\mathsf{S}, s) : \pi \preceq \pi' \text{ implies } (\mathsf{S}, \pi') \models \psi\}$$

  and $\mathsf{min}(X)$ is the set of minimal elements of X according to the prefix ordering $\preceq$ on paths. The paths in $Con(\mathsf{S}, s, \psi)$ are called *ψ-conservative (in S starting at s)*, and paths in $\mathsf{min}(Con(\mathsf{S}, s, \psi))$ are called *minimal ψ-conservative*.

If ψ is a GCTL* path formula and $\pi = \pi_0\pi_1 \cdots \in \mathsf{pth}(\mathsf{S})$ is a finite or infinite path in S, then define $(\mathsf{S}, \pi) \models \psi$ inductively:

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

If ψ is an LTL formula, we may write $\pi \models \psi$ instead of $(\mathsf{S}, \pi) \models \psi$. This is justifiable since the truth of ψ depends only on the path π independently of the rest of S. Two state formulas ϕ, ϕ' are *equivalent* if for all S and $s \in S$, we have $(\mathsf{S}, s) \models \phi$ iff $(\mathsf{S}, s) \models \phi'$. Two path formulas ψ, ψ' are *equivalent* if for all S and $\pi \in \mathsf{pth}(\mathsf{S})$, we have that $(\mathsf{S}, \pi) \models \psi$ if and only if $(\mathsf{S}, \pi) \models \psi'$. An LTS S with a designated state $q \in S$ is a *model* of a GCTL* formula φ, sometimes denoted $\mathsf{S} \models \varphi$, if $(\mathsf{S}, q) \models \varphi$. For a labeled tree T, the designated node is by default the root, and thus, $\mathsf{T} \models \varphi$ means that $(\mathsf{T}, \epsilon) \models \varphi$ (recall that ϵ refers to the root of T). A GCTL* formula φ is *satisfiable* iff it has a model.

Example 1. *We unpack the meaning of the* GCTL* *formula* $\mathsf{E}^{\geq g}\mathsf{F}request$. *Let ψ be the path formula* $\mathsf{F}request$ *and let π be a path. Then, π is ψ-conservative*

iff π satisfies ψ, i.e., at some point on π the atom request holds, and it is not ψ-conservative iff at no point on it does request hold. Thus, π is minimal ψ-conservative iff π is finite and request occurs at the last point of π, and only there. Note that no infinite path is minimal ψ-conservative. Thus, $\mathsf{E}^{\geq g}\mathsf{F}request$ says that there are at least g many distinct finite paths in which request occurs at the end of each of these paths, and only there.

Example 2. *We unpack the meaning of the* GCTL* *formula $\mathsf{E}^{\geq g}\mathsf{GF}request$. Let ψ be the path formula $\mathsf{GF}request$ and let π be a path. Then, π is ψ-conservative iff π is infinite and satisfies ψ (recall that no finite path satisfies a formula of the form $\mathsf{G}\varphi$). Thus, π is minimal ψ-conservative iff it is infinite and it satisfies ψ, and $\mathsf{E}^{\geq g}\mathsf{GF}request$ says that there are at least g many different infinite paths in which request occurs infinitely often.*

Example 3. *We unpack the meaning of the* GCTL* *formula from the introduction: $\mathsf{E}^{\geq 2}\mathsf{F}(request \wedge \neg \mathsf{F}granted)$. Let ψ denote the path formula $\mathsf{F}(request \wedge \neg \mathsf{F}granted)$. First, a finite or infinite path π satisfies ψ if at some point t the atom request holds, and at no later point on π does the atom granted hold. A finite π is ψ-conservative if and only if it satisfies ψ and the atom granted does not hold in any node of the subtree rooted at the end of π; and an infinite path is ψ-conservative if and only if it satisfies ψ. Thus, $\mathsf{E}^{\geq 2}\psi$ holds if and only if there exist two possibly finite paths, say π^1 and π^2, neither one a prefix of the other, both satisfying ψ (i.e., π^i has a request that is never granted on π^i), and such that if π^i is finite then that path has a request that is not granted in any possible extension of π^i.*

Remark 1. The additional operators present in the syntax of GCTL* in [14], namely $\wedge, \tilde{\mathsf{X}}, \tilde{\mathsf{R}}, \tilde{\mathsf{U}}$ and $\mathsf{A}^{<g}$ are dual to the operators $\vee, \mathsf{X}, \mathsf{R}, \mathsf{U}$ and $\mathsf{E}^{\geq g}$. There are more dual operators in GCTL* than in CTL*, e.g., X is not a dual of itself. The reason is that one also has to consider finite paths. Thus, the tilded operators are defined like the un-tilded operators on infinite paths, but with 'optimistic semantics' (as opposed to the pessimistic semantics of the un-tilded operators) on finite paths; e.g., $(\mathsf{S}, \pi) \models \tilde{\mathsf{X}}\psi$:if $|\pi| = 1$ or $(\mathsf{S}, \pi_{\geq 1}) \models \psi$.

Remark 2. The GCTL* formula $\mathsf{E}^{\geq 1}\psi$ means that there exists a ψ-conservative path, i.e., either there exists an infinite path satisfying ψ, or there exists a finite path π satisfying ψ such that every (finite and infinite) extension of π satisfies ψ. Thus, if S is total then $\mathsf{S} \models \mathsf{E}^{\geq 1}\psi$ if and only if S has an infinite path satisfying ψ. Hence, for total LTSs, the classic logic CTL* coincides with the fragment of GCTL* in which the degree g of all quantifiers $\mathsf{E}^{\geq g}$ is 1.

Remark 3. GCTL* can easily express graded world modalities (i.e., it can easily count state formulas over successors). Indeed, for a state formula φ, the GCTL* formula $\mathsf{E}^{\geq g}\mathsf{X}\varphi$ expresses that there exist at least g immediate successors of the current node satisfying φ. To see that this is the correct semantics of this formula observe that: i) a path of length 1 does not satisfy the path formula $\mathsf{X}\varphi$, and thus is not $\mathsf{X}\varphi$-conservative; ii) if $(\mathsf{S}, \pi) \models \mathsf{X}\varphi$, then

φ holds on the second state of π (recall that φ is a state formula), and thus every extension π', of the prefix of π of length 2, satisfies $\mathsf{X}\varphi$, and thus π is minimal $\mathsf{X}\varphi$-conservative iff $|\pi| = 2$. Recall that $(\mathsf{S}, s) \models \mathsf{E}^{\geq g}\mathsf{X}\varphi$ iff there are at least g minimal $\mathsf{X}\varphi$-conservative paths, which by the facts above must all be of length 2. I.e, $(\mathsf{S}, s) \models \mathsf{E}^{\geq g}\mathsf{X}\varphi$ iff there are at least g immediate successors of s satisfying φ. The extension of CTL* by the state formulas $\mathsf{E}^{\geq g}\mathsf{X}\varphi$, sometimes written $\mathsf{D}^g\varphi$, is called counting-CTL* [41].

2.2. Important Properties of GCTL*

In this section we state and prove some important or useful properties of the logic just defined.

2.2.1. Treating path formulas as LTL *formulas*

We show how to treat GCTL* path formulas as LTL formulas over new atoms. We do this just as is done for CTL* [35]. Roughly, one can think of a GCTL* path formula ψ over atoms AP as an LTL formula Ψ over atoms which themselves are GCTL* state formulas.

Here are the details. A formula φ is a *state sub-formula* of ψ if i) φ is a state formula, and ii) φ is a sub-formula of ψ. A formula φ is a *maximal state sub-formula* of ψ if φ is a state sub-formula of ψ, and φ is not a proper sub-formula of any other state sub-formula of ψ. Let $max(\psi)$ be the set $\{\varphi \mid \varphi \text{ is a maximal state sub-formula of } \psi\}$, and let $\overline{max(\psi)}$ be the set of all maximal state sub-formulas of ψ and their negations, i.e., $\bigcup_{\varphi \in max(\psi)}\{\varphi, \neg\varphi\}$.

Every GCTL* path formula ψ can be viewed as the formula Ψ whose atoms are elements of $max(\psi)$. Note that Ψ is an LTL formula. For example, for $\psi = ((\mathsf{X}p)\ \mathsf{U}\ (\mathsf{E}^{\geq 2}\mathsf{X}q)) \vee p$, the state sub-formulas are $\{p, q, \mathsf{E}^{\geq 2}\mathsf{X}q\}$, and $max(\psi) = \{p, \mathsf{E}^{\geq 2}\mathsf{X}q\}$, and thus Ψ is the LTL formula $(\mathsf{X}\underline{p}\ \mathsf{U}\ \underline{\mathsf{E}^{\geq 2}\mathsf{X}q}) \vee \underline{p}$ over the atoms $\{\underline{p}, \underline{\mathsf{E}^{\geq 2}\mathsf{X}q}\}$ (here we underline sub-formulas that are treated as atoms).

Given an LTS $\mathsf{S} = \langle 2^{\mathrm{AP}}, S, E, \lambda\rangle$ and a GCTL* path formula ψ, we define the *relabeling* of the LTS S by the values of the formulas in $max(\psi)$ as $\mathsf{S}_\psi = \langle max(\psi), S, E, L\rangle$ where $L(s)$ is the union of $\lambda(s)$ and the set of $\varphi \in max(\psi)$ such that $(\mathsf{S}, s) \models \varphi$.

Lemma 1. *For every* GCTL* *path formula* ψ *over* AP *there is an* LTL *formula* Ψ *over* $max(\psi)$ *such that for all* S *and all paths* π *in* S*:* $(\mathsf{S}, \pi) \models \psi$ *iff* $(\mathsf{S}_\psi, \pi) \models \Psi$*.*

The proof of this Lemma is in Appendix A.

2.2.2. Invariance under bisimulation and unwinding

It is not hard to see that GCTL* is not invariant under bisimulation (cf. [14]), and that it is invariant under unwinding (cf. [14]).

Lemma 2. *[14]*

1. GCTL* *is not invariant under bisimulation.*

2. GCTL* *is invariant under unwinding.*

PROOF. The proof for the bisimulation is straightforward: Consider the formula $\mathsf{E}^{\geq 2}\mathsf{X}p$. 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.

The proof for the unwinding is standard (cf. [14]): to treat $\mathsf{E}^{\geq g}$ 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 $\mathsf{E}^{\geq g}$ involve reasoning about $\prec$. See Appendix B for details. □

2.2.3. Expressive Power

The next theorem shows that GCTL* is a powerful logic. Indeed, it is equivalent, over trees, to Monadic Path Logic (MPL) which is MSO with quantification restricted to branches. Note that MPL is only defined over trees, while GCTL* (like CTL*) is defined over arbitrary LTS. This is the reason we compare their expressiveness over trees.

We briefly summarise the syntax and semantics of MPL [41]. For a tree T write $\mathsf{branches}(\mathsf{T})$, 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 $=$, $\prec$, ϵ and L_p for atoms $p \in \mathrm{AP}$. The semantics are defined for labeled trees $\mathsf{T} = \langle T, V\rangle$ where $V : T \to 2^{\mathrm{AP}}$. The interpretation of variables x are over elements of T, of variables X are over branches of T, of $=$ is the usual equality of variables, of $\prec$ is as the ancestor relation of T, of $x \in X$ 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 $p \in V(t)$. An MPL formula without free variables is called a *sentence*.

Theorem 1. GCTL* *is equivalent, over trees, to Monadic Path Logic (MPL). That is, for every* GCTL* *formula* φ *there is an MPL sentence* $\widehat{\varphi}$ *such that for all trees* T, $\mathsf{T} \models \varphi$ *iff* $\mathsf{T} \models \widehat{\varphi}$*; and vice versa.*

Remark 4. Before supplying the proof, observe that since GCTL* is invariant under unwinding, and since MPL has the finitely-branching tree model property[5], Theorem 1 implies that GCTL* also has the finitely-branching tree model property, i.e., if a GCTL* formula φ is satisfiable then it is satisfiable in a finitely-branching tree. Note that the finitely-branching tree model property of GCTL* does not follow solely from the fact that GCTL* is closed under unwinding because we allow countable LTSs, and thus possibly infinitely-branching trees as models. Also note that we have not (yet) deduced that there is a bound on the number of children of every node. This will be done, with more work, in Theorem 5.

[5]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 [49].

PROOF. Recall from Remark 3 that the logic counting-CTL* is defined by adding to CTL* the state formulas $\mathsf{D}^n\phi$ (where ϕ is a state formula and $n \in \mathbb{N}$) which are interpreted as saying that at least n children of the current node satisfy ϕ. By Remark 3, GCTL* is at least as expressive (over LTS, and thus over trees) as counting-CTL*. But the main result in [41] 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 ϕ there exists an MPL formula $\widehat{\phi}(x)$ such that for all trees T, and all $t \in \mathsf{T}$: $(\mathsf{T}, t) \models \phi$ if and only if $\mathsf{T} \models \widehat{\phi}(t)$.

In this proof we freely switch between viewing T as a tree and as an LTS. We start with some notation and three facts. We begin with some notation. For $\pi \in \mathsf{pth}(\mathsf{T})$ and a node a on π write $\pi_{[a,\infty)} \in \mathsf{pth}(\mathsf{T}, a)$ for the tail of π starting at a. Also, for $a, b \in T$ with $a \preceq b$, write $\pi_{[a,b]} \in \mathsf{pth}(\mathsf{T}, a)$ for the subpath of π starting at a and ending at b.

Fact 1. For every LTL formula Ψ 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 paths $\pi \in \mathsf{branches}(\mathsf{T})$ 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 Ψ 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 $a \preceq b$: $T \models \Psi''(a, b)$ if and only if $(T, \pi_{[a,b]}) \models \Psi$.

Fact 3. For every LTL formula Ψ 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 π starting at a is minimal Ψ-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 $a \preceq b$: $\mathsf{T} \models mincon_\Psi(a, b)$ if and only if the path between a and b is minimal Ψ-conservative in (T, a).

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 Ψ. If Ψ is an atom, say $p \in$ 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 Ψ is a disjunction. If $\Psi = \Psi_1 \mathsf{U} \Psi_2$ then $\Psi''(x, y)$ is defined as $\exists z.x \preceq z \preceq y \wedge [\Psi_2''(x, z) \wedge \forall v.x \preceq v \prec z \rightarrow \Psi_1'(x, v)]$. The cases X and R are similar to U. This completes the proof of Fact 2.

Proof of Fact 3: We show how to define the ingredients of the required formulas. We will use Ψ' from Fact 1, and Ψ'' from Fact 2.

- Let $end(X, z)$ denote the formula $z \in X \wedge \forall y \in X.y \preceq z$ stating that z is the last node of the branch X.
- Let $finite(X)$ denote the formula $\exists z.end(X, z)$ stating that branch X is finite.
- We use the shorthand $end(X)$ for the unique value $end(X, z)$ if it exists.

- Let $fin\_mincon_\psi(X, x)$ denote the formula

$$\begin{aligned} finite(X) \wedge\ & (\forall y.end(X) \preceq y \rightarrow \Psi''(x, y)) \\ & \wedge (\forall Y.end(X) \in Y \rightarrow \Psi'(x, Y)) \\ & \wedge (\forall z.x \preceq z \prec end(X) \rightarrow \neg\Psi''(x, z)) \end{aligned}$$

  stating that the path X starting at x is finite and minimal Ψ-conservative.

- Let $inf\_mincon_\psi(X, x)$ denote the formula

$$\neg finite(X) \wedge \Psi'(x, X) \ \wedge\ (\forall z.x \preceq z \in X \rightarrow \neg\Psi''(x, z))$$

  stating that the path X starting at x is infinite and minimal Ψ-conservative.

- Finally, let $mincon_\Psi(x, X)$ denote the formula

$$fin\_mincon_\psi(X, x) \vee inf\_mincon_\psi(X, x)$$

  stating that the path X starting at x is minimal Ψ-conservative (irrespective of X being finite or infinite).

Similarly, the formula $mincon_\Psi(x, w)$ is defined as in $fin\_mincon_\psi(X, x)$ but replacing $end(X)$ by w.

This completes the proof of Fact 3.

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

- If ϕ is an atom, say p, then $\widehat{\phi}(x)$ is defined as the unary predicate $L_p(x)$.

- If ϕ is of the form $\neg\phi'$, then $\widehat{\phi}$ is defined as $\neg\widehat{\phi}$; and similarly for $\vee$ and $\wedge$.

- If ϕ is of the form $\mathsf{E}^{\geq g}\psi$, then let Ψ be the LTL formula corresponding to ψ 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_θ for $\theta \in max(\psi)$.

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

$$\begin{aligned} \bigvee_{h\in[0,g]} \exists X_1, \cdots, X_h. \bigwedge_{1\leq i<i\leq h} X_i \neq X_j \wedge \exists x_1, \cdots, x_{g-h}. \\ \Big[\bigwedge_{i\in[1,h]} mincon_\Psi(x, X_i)[L_\theta(z)/\hat{\theta}(z)] \\ \wedge \bigwedge_{i\in[1,g-h]} mincon_\Psi(x, x_i)[L_\theta(z)/\hat{\theta}(z)\Big] \end{aligned}$$

  where $mincon_\Psi(x, X_i)[L_\theta(z)/\hat{\theta}(z)]$ is the MPL formula $mincon_\Psi(x, X_i)$ in which every occurrence of a subformula 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. □

By the proof of Theorem 1 we also have the following:

Corollary 1. GCTL* *is equivalent, over trees, to counting-*CTL**. That is, for every* GCTL* *formula* φ *there is a counting-*CTL**formula* $\widehat{\varphi}$ *such that for all trees* T*,* $\mathsf{T} \models \varphi$ *iff* $\mathsf{T} \models \widehat{\varphi}$*; and vice versa.*

3. Graded Hesitant Tree Automata

In this section we define a new kind of automaton called Graded Hesitant Tree Automata (GHTA). In the next section we will show how to compile GCTL* formulas into GHTA.

We make use of the classical non-deterministic finite word automata (NFW) and non-deterministic Büchi word automata (NBW) (see [48]), alternating parity tree automata (APTA) (see [23]), and alternating hesitant tree automata (AHTA) (see [35]). We write $\langle \Sigma, Q, q_0, \delta, G\rangle$ for NBWs and $\langle \Sigma, Q, q_0, \delta, F\rangle$ for NFWs where Σ is the input alphabet, Q 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. For a set X, let $\mathsf{B}^+(X)$ be the set of positive Boolean formulas over X, including the constants **true** and **false**. A set $Y \subseteq X$ satisfies a formula $\theta \in \mathsf{B}^+(X)$, written $Y \models \theta$, if assigning **true** to elements in Y and **false** to elements in $X \setminus Y$ makes θ true. Graded hesitant tree automata (GHTA) generalise AHTA[6]: 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. We formally define AHTA and GHTA below.

3.1. Definition of AHTA

An *Alternating Hesitant Tree Automaton* (AHTA) is a tuple

$$\mathsf{A} = \langle \Sigma, D, Q, q_0, \delta, \langle G, B\rangle, \langle \mathsf{part}, \mathsf{type}, \preceq\rangle\rangle$$

where Σ is a non-empty finite set of *input letters*; $D \subset \mathbb{N}$ 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 $\langle G, B\rangle \in 2^Q \times 2^Q$ is the *acceptance condition*[7] (we sometimes

[6]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.

[7]The combination of a Büchi and a co-Büchi condition that hesitant automata use 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 (using the parity condition, our automata strictly generalise the ones in [34, 13]). However, we do not need the full power of the parity condition, and in order to achieve optimal complexity for model checking of GCTL* we need to be able to decide membership of our automata in a space efficient way, which cannot be done with the parity acceptance condition.

call the states in G *good states* and the states in B *bad states*); $\delta : Q \times \Sigma \to \mathsf{B}^+(D \times Q)$ is the *alternating transition function*; $\mathsf{part} \subset 2^Q$ is a partition of Q, $\mathsf{type} : \mathsf{part} \to \{trans, exist, univ\}$ is a function assigning the label *transient*, *existential* or *universal* to each element of the partition, and $\preceq \subset 2^Q \times 2^Q$ is a partial order on part. Moreover, the transition function δ is required to satisfy the following *hesitancy condition*: for every $\mathbf{Q} \in \mathsf{part}$, every $q \in \mathbf{Q}$, and every $\sigma \in \Sigma$: (i) for every $\mathbf{Q}' \in \mathsf{part}$ and $q' \in \mathbf{Q}'$, if q' occurs in $\delta(q, \sigma)$ then $\mathbf{Q}' \preceq \mathbf{Q}$; (ii) if $\mathsf{type}(\mathbf{Q}) = trans$ then no state of $\mathbf{Q}$ occurs in the formula $\delta(q, \sigma)$; (iii) if $\mathsf{type}(\mathbf{Q}) = exist$ (resp., $\mathsf{type}(\mathbf{Q}) = univ$) then there is at most one element of $\mathbf{Q}$ in each disjunct of the DNF (resp., conjunct of CNF) of $\delta(q, \sigma)$. Intuitively, the hesitancy condition guarantees that paths in the run of the automaton eventually get trapped in a single existential or universal element of part.

An *input tree (for AHTA)* is a Σ-labeled tree $\mathsf{T} = \langle T, V \rangle$ 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 $\mathsf{T} = \langle T, V \rangle$ is a $(T \times Q)$-labeled tree $\langle T_r, r \rangle$, such that *(a)* $r(\varepsilon) = (\varepsilon, q_0)$ and *(b)* for all $y \in T_r$, with $r(y) = (x, q)$, there exists a *minimal* set $\mathsf{S} \subseteq D \times Q$, such that $\mathsf{S} \models \delta(q, V(x))$, and for every $(d, q') \in \mathsf{S}$, it is the case that $x \cdot d$ is a son of x, and there exists a son y' of y, such that $r(y') = (x \cdot d, q')$.

Note that if $\delta(q, V(x)) = \mathbf{true}$ 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 $loc(v)$, and call q the *state associated with* v, denoted $state(v)$.

We now discuss the acceptance condition. Fix a run tree $\langle T_r, r \rangle$ and an infinite path π in it. Say that the path *visits* a state q at time i if $state(\pi_i) = q$. The hesitancy condition (i) guarantees that the path π eventually gets trapped and visits only states in some element of the partition, i.e., there exists $\mathbf{Q} \in \mathsf{part}$ such that from a certain time i on, $state(\pi_j) \in \mathbf{Q}$ for all $j \geq i$. The hesitancy condition (ii) ensures that this set is either existential or universal, i.e., $\mathsf{type}(\mathbf{Q}) \in \{exist, univ\}$. Thus, we say that the path π *gets trapped in an existential set* if $\mathsf{type}(\mathbf{Q}) = exist$, 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 *accepting* iff it visits some state of G infinitely often, and a path that gets trapped in a universal set is *accepting* iff it visits every state of B finitely often. A run $\langle T_r, r \rangle$ of an AHTA is *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 $\langle T, V \rangle$. The *language* of A, denoted $\mathcal{L}(\mathsf{A})$, is the set of Σ-labeled D-trees accepted by A. We say that A is nonempty iff $\mathcal{L}(\mathsf{A}) \neq \emptyset$.

The *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 *depth* of the AHTA is the size of the longest $\preceq$-chain over part. The *size* $||\delta||$ of the transition

function is the sum of the lengths of the formulas it contains. The *size* $||\mathsf{A}||$ of the AHTA is $|D|+|Q|+||\delta||$. The partition, partial order and type function are not counted in the size of the automaton. The following is implicit in [35]:

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

3.2. Definition of GHTA

We now define *Graded Hesitant Tree Automata* (GHTA). These 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 $\Box(q_1,...,q_k)$. 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 of itself in state q_i to son s_i. Note that the states $q_1,...,q_k$ are not necessarily all different. Dually, $\Box(q_1,...,q_k)$ says that for every k different sons $s_1,...,s_k$ of the current node, the automaton picks one of these sons s_i to which it sends a copy of itself in state q_i.

A GHTA A is a tuple $\langle \Sigma, Q, q_0, \delta, \langle G, B\rangle, \langle \mathsf{part}, \mathsf{type}, \preceq\rangle\rangle$ where all elements but δ are defined as for AHTA, and $\delta : Q \times \Sigma \to \mathsf{B}^+(\Diamond_Q \cup \Box_Q)$ is a transition function that maps a state and an input letter to a positive Boolean combination of elements in $\Diamond_Q = \{\Diamond(q_1,...,q_k) \,|\, (q_1,...,q_k) \in Q^k, k \in \mathbb{N}\}$ and $\Box_Q = \{\Box(q_1,...,q_k) \,|\, (q_1,...,q_k) \in Q^k, k \in \mathbb{N}\}$.

We show how to define the run of a GHTA A on a Σ-labeled finitely-branching tree $\mathsf{T} = \langle T, V\rangle$ by (locally) unfolding every $\Diamond_Q$ and $\Box_Q$ in $\delta(q, V(t))$ into a formula in $\mathsf{B}^+([d] \times Q)$ 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 $expand_d : \mathsf{B}^+(\Diamond_Q \cup \Box_Q) \to \mathsf{B}^+([d] \times Q)$ that maps formula ϕ to the formula formed from ϕ by replacing every occurrence of a sub-formula of the form $\Diamond(q_1,...,q_k)$ by the formula

$$\bigvee_{(s_1,...,s_k)\in S(k,d)} \bigwedge_{i\leq k} (s_i, q_i),$$

and every occurrence of a sub-formula of the form $\Box(q_1,...,q_k)$ by the formula

$$\bigwedge_{(s_1,...,s_k)\in S(k,d)} \bigvee_{i\leq k} (s_i, q_i).$$

Observe that if $k > d$ then $\Diamond(q_1,...,q_k)$ becomes the constant formula **false**, and $\Box(q_1,...,q_k)$ becomes the constant formula **true**. The *run of a GHTA* A is defined as for an alternating tree automaton, except that one uses $expand_n(\delta(q, V(x)))$ instead of $\delta(q, V(x))$ for nodes x of T of degree n. Finally, the *hesitancy condition* defined above for AHTA is required to apply to the expanded transition

function, i.e., for every $\mathbf{Q} \in \mathsf{part}$, every $q \in \mathbf{Q}$, every $n \in \mathbb{N}$, and every $\sigma \in \Sigma$: (i) for every $\mathbf{Q}' \in \mathsf{part}$ and $q' \in \mathbf{Q}'$, if q' occurs in $expand_n(\delta(q,\sigma))$ then $\mathbf{Q}' \preceq \mathbf{Q}$; (ii) if $\mathsf{type}(\mathbf{Q}) = trans$ then no state of $\mathbf{Q}$ occurs in the formula $expand_n(\delta(q,\sigma))$; (iii) if $\mathsf{type}(\mathbf{Q}) = exist$ (resp., $\mathsf{type}(\mathbf{Q}) = univ$) then there is at most one element of $\mathbf{Q}$ in each disjunct of the DNF (resp., conjunct of CNF) of $expand_n(\delta(q,\sigma))$. Acceptance is as for AHTA.

Lemma 3. *The emptiness problem for a GHTA* A*, over trees of branching degree at most* d*, is decidable in time* $2^{O(d\cdot|Q|^3)}$*, where* Q *is the state set of* A*. Furthermore, if* A *is not empty then one can obtain (in the same time complexity) a regular tree accepted by* A*.*

Proof. Given a GHTA A with state set Q, convert it into an AHTA A' accepting the same language, with the same states, by using the function $expand_d$ defined above to transform its transition relation into a non-graded one. This is possible since we assumed a bound d on the branching degree of the input trees, and thus the transformation $expand_d$ can be used in advance. This construction takes time that is $2^{O(|Q|\log d)}$. Recall that AHTA are a special case of alternating parity tree automata (APTA) with 3 priorities. Now apply the fact that the emptiness problem (as well as obtaining a regular tree witnessing non-emptiness) for APTA with p priorities over d-ary trees can be solved in time $2^{O(d\cdot|Q|^p)}$ [23]. □

Lemma 4. *The membership problem for GHTA* A *for regular trees is decidable in* $O(\partial \log^2(|\delta||Q|^{|S|}))$ *space, i.e., linear in the depth of* A*, quadratic in the size of* S*, polylogarithmic in the size of the transition function of* A *and the number of states of* A*.*

Proof. Let A have state set Q, transition function δ, and depth ∂. Recall that the translation used in Lemma 3 that transforms a GHTA A into an equivalent AHTA A' (on trees of degree at most d) works by replacing $\delta(q,\sigma)$ by $expand_d(\delta(q,\sigma))$. Note that expanding the $\Diamond_Q$s and $\Box_Q$s only blows up the size of the transition relation by a multiplicative factor of $|Q|^d$, and leaves the state space unchanged. Conclude that A' is the same as A except that its transition function δ' is such that $||\delta'|| \leq ||\delta|| \cdot |Q|^d$. Then $||\mathsf{A}'|| = d + |Q| + |\delta||Q|^d$.

By Theorem 2, and letting d be the maximum degree in S, so $d \leq |S|$, the membership problem of the AHTA A' on the unwinding of the LTS S can be solved in space $O(\partial \log^2(|S| \cdot ||\mathsf{A}'||))$ which is linear in ∂, quadratic in $|\mathsf{S}|$, and polylogarithmic in $|\delta|$ and $|Q|$. □

4. 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 [35]. 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 Remark 4, 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:

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

This section, devoted to the proof of this Theorem, is structured as follows. In Section 4.1 we give an important characterization of the semantics of the graded path modality $\mathsf{E}^{\geq g}\psi$ that allows us to achieve an optimal construction. We also give an intuition of the construction as well as some supporting lemmas that build NBWs and NFWs for LTL formulas. In Section 4.2 we formally give the construction of GHTA A_ϑ for a GCTL* formula ϑ. In Section 4.3 we analyse the size of the resulting automata. In Section 4.4 we give the proof that the construction is correct.

4.1. Characterization of the graded path modality and intuition of the construction

An important observation that allows us to achieve an optimal construction is the following. Suppose that the formula $\mathsf{E}^{\geq g}\psi$ holds at some node w of a tree. Then, by definition, there are at least g different paths $\rho'^1,...,\rho'^g \in \mathsf{min}(Con(\mathsf{S}, w, \psi))$. Look at any g infinite extensions $\rho^1,...,\rho^g$ of these paths in the tree, and note that by the definition of ψ-conservativeness all these extensions must satisfy ψ. Also observe that for every $i \neq j$, the fact that ρ'^i, ρ'^j are different and minimal implies that the longest common prefix ρ'^{ij} of ρ^i and ρ^j is not ψ-conservative. As it turns out, the other direction is also true, i.e., if there are g infinite paths $\rho^1,...,\rho^g$ satisfying ψ, such that for every $i \neq j$ the common prefix ρ'^{ij} is not ψ-conservative, then there are g prefixes $\rho'^1,...,\rho'^g$ of $\rho^1,...,\rho^g$ respectively, such that $\rho'^1,...,\rho'^g \in \mathsf{min}(Con(\mathsf{S}, w, \psi))$. Note that this allows us to reason about the cardinality of the set $\mathsf{min}(Con(\mathsf{S}, w, \psi))$, by considering only the infinite paths $\rho^1,...,\rho^g$ and their common prefixes, without actually looking at the minimal ψ-conservative paths $\rho'^1,...,\rho'^g$. In reality, we do not even have to directly consider the common prefixes ρ'^{ij}. Indeed, since the property of being ψ-conservative is upward closed (with respect to the prefix ordering $\preceq$ of paths), showing that ρ'^{ij} is not ψ-conservative can be done by finding any extension of ρ'^{ij} that is not ψ-conservative. The following proposition formally captures this.

Proposition 1. *Given a* GCTL* *path formula* ψ *and a* 2^{AP}*-labeled tree* $\mathsf{T} = (T, V)$*, then* $\mathsf{T} \models \mathsf{E}^{\geq g}\psi$ *iff there are* g *distinct nodes* $y_1,...,y_g \in \mathsf{T}$ *(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 ψ-conservative; (iii) there is an infinite path ρ^i in T, starting at the root and going through y_i, such that $\rho^i \models \psi$.

PROOF. Assume first that $\mathsf{T} \models \mathsf{E}^{\geq g}\psi$, and let $\rho'^1, \ldots, \rho'^g$ be g different paths in the set $\mathsf{min}(Con(\mathsf{S}, \epsilon, \psi))$. For $1 \leq i \leq g$, let y_i be an arbitrarily chosen point on the path ρ'^i satisfying, for every $j \neq i$, that y_i is not on the path ρ'^j. Observe that such a point exists since, by minimality, $\rho'^i \not\preceq \rho'^j$ 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 ρ'^i. Indeed, the path from the root to the father of y_i is a proper prefix of ρ'^i, and is thus not in $Con(\mathsf{S}, \epsilon, \psi)$. By the definition of ψ-conservativeness, we have that every path ρ^i in T such that $\rho'^i \preceq \rho^i$ satisfies ψ. Recall that we assume that trees are total, i.e., that they contain no leaves, and thus property (iii) holds by simply taking ρ^i to be any infinite extension of ρ'^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 ρ'^i be the shortest prefix of ρ^i such that ρ'^i is ψ-conservative, and note that $\rho'^i \in \mathsf{min}(Con(\mathsf{S}, \epsilon, \psi))$. The path ρ'^i is well defined since ρ^i is infinite and satisfies ψ and thus, by definition, it is ψ-conservative. In order to prove that $\mathsf{T} \models \mathsf{E}^{\geq g}\psi$, it remains to show that for every $i \neq j$ we have that $\rho'^i \neq \rho'^j$. To see that, observe that for every $1 \leq i \leq g$, property (ii) together with the fact that the property of being ψ-conservative is upward closed (with respect to the prefix ordering $\preceq$ of paths), imply that the path from the root to the father of y_i is a proper prefix of ρ'^i and thus, ρ'^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 $\rho'^i \neq \rho'^j$, which completes the proof. □

We are in a position to describe our construction of a GHTA accepting all finitely-branching tree-models of a given GCTL* formula. We begin with an intuition and some supporting lemmas. The full construction is given in Section 4.2.

Naturally, the main difficulty lies in handling the graded modalities. The basic intuition behind the way our construction handles formulas of the form $\varphi = \mathsf{E}^{\geq g}\psi$ is the following. Given an input tree, the automaton A_φ for this formula has to find at least g minimal ψ-conservative paths. At its core, A_φ runs g pairs of copies of itself in parallel. The reason 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 overcounting the number of minimal ψ-conservative paths). The task of each of the g pairs is to detect some minimal ψ-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 if certain paths satisfy ψ, the automaton A_φ will access certain classic NBWs. The following Theorem, whose proof is reported in Appendix C, states the

existence of these. The first part is classic, see [47, 19], and the second is a simple adaptation.

Theorem 4. 1. *Given an* LTL *formula* ζ*, there is an NBW* $\mathbb{A}_\zeta$ *(resp. NFW* $\mathbb{B}_\zeta$*), of size* $2^{O(|\zeta|)}$*, accepting exactly all infinite (resp. finite) words that satisfy* ζ*.*

2. *Given an* LTL *formula* ζ*, there is an NBW* $\mathbb{A}^\zeta$ *(of size* $2^{O(|\psi|)}$*) such that* $\mathbb{A}^\zeta$ *accepts a word* w *iff* $w \models \zeta$*, or* $u \models \zeta$ *for a prefix* u *of* w*.*
Moreover, $\mathbb{A}^\zeta$ *has an accepting sink* $\top$*, such that if* $r_0, r_1, \ldots$ *is an accepting run of* $\mathbb{A}^\zeta$ *on* w*, and* $i \geq 0$ *satisfies* $r_i \neq \top$*, then a (finite or infinite) prefix* u *of* w*, of length* $|u| > i$*, satisfies* ζ*, and vice-versa (i.e., if a prefix* u *of* w *satisfies* ζ*, then there is an accepting run on* w *with* $r_i \neq \top$ *for all* $i < |u|$*).*

We can now finish the intuitive description of the construction of the automaton A_φ associated with a formula $\varphi = \mathsf{E}^{\geq g}\psi$. Let Ψ be the LTL formula resulting from applying Lemma 1 to ψ. In essence, A_φ 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 $\mathbb{A}^{\neg\Psi}$ to verify that the path π, from the root to the father of y_i, is not ψ-conservative (by guessing some finite or infinite extension $\pi \preccurlyeq \pi'$ of it such that $\pi' \models \neg\Psi$), and one copy of $\mathbb{A}_\Psi$ to guess an infinite path π'' from the root through y_i such that $\pi'' \models \Psi$ (and is thus ψ-conservative).

4.2. The construction of GHTA A_ϑ *for a* GCTL* *formula* ϑ*.*

We induct on the structure of ϑ. Given a state sub-formula ϕ of ϑ (possibly including ϑ), for every formula $\theta \in \overline{max(\phi)}$, let $\mathsf{A}_\theta = \langle \Sigma, Q^\theta, q_0^\theta, \delta^\theta, \langle G^\theta, B^\theta \rangle, \langle \mathsf{part}^\theta, type^\theta, \preceq^\theta \rangle \rangle$ be a GHTA accepting the finitely-branching tree-models of θ.

We build the GHTA A_ϕ accepting all finitely-branching tree-models of ϕ 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.[8] Formally:

1. If $\phi = p \in AP$, then $\mathsf{A}_\phi = \langle \Sigma, \{q\}, q, \delta, \langle \emptyset, \emptyset \rangle, \langle \mathsf{part}, \mathsf{type}, \preceq \rangle \rangle$ where $\delta(q, \sigma) =$ **true** if $p \in \sigma$ and **false** otherwise.
2. If $\phi = \varphi_0 \vee \varphi_1$ then A_ϕ is obtained by nondeterministically invoking either A_{φ_0} or A_{φ_1}.
Thus, $\mathsf{A}_\phi = \langle \Sigma, \bigcup_{i=0,1} Q^{\varphi_i} \cup \{q_0\}, q_0, \delta, \langle \bigcup_{i=0,1} G^{\varphi_i}, \bigcup_{i=0,1} B^{\varphi_i} \rangle, \beta \rangle$, where $\beta = \langle \mathsf{part}, \mathsf{type}, \preceq \rangle$, and for every $i \in \{0, 1\}$, every $\sigma \in \Sigma$, and every

[8] For example, when building an automaton for $\phi = \varphi_0 \vee \varphi_1$, in the degenerate case that $\varphi_0 = \varphi_1$ then A_{φ_1} is taken to be a copy of A_{φ_0} with its states renamed to be disjoint from those of A_{φ_0}. Also, the new state q_0 may be renamed to avoid a collision with any of the other states.

$q \in Q^{\varphi_i}$ we have that: $\delta(q,\sigma) = \delta^{\varphi_i}(q,\sigma)$, and $\delta(q_0,\sigma) = \delta^{\varphi_0}(q_0^{\varphi_0},\sigma) \vee \delta^{\varphi_1}(q_0^{\varphi_1},\sigma)$.

3. If $\phi = \neg\varphi$, then A_ϕ is obtained by dualizing the automaton A_φ. Formally, the *dual of a GHTA* A is the GHTA obtained by dualizing the transition function of A (i.e., switch $\vee$ and $\wedge$, switch $\top$ and $\bot$, and switch $\Box$ and $\Diamond$), replacing the acceptance condition $\langle G, B\rangle$ with $\langle B, G\rangle$ (and toggling types).

Finally we deal with the case that $\phi = \mathsf{E}^{\geq g}\psi$.

In this case $\mathsf{A}_\phi = \langle \Sigma, Q, q_0, \delta, \langle G, B\rangle, \langle \mathsf{part}, \mathsf{type}, \preceq\rangle\rangle$ and its structure is detailed below. Observe that ψ is a path formula and, by Lemma 1, reasoning about ψ can be reduced to reasoning about the LTL formula Ψ whose atoms are elements of $max(\psi)$. Let $\Sigma' = 2^{max(\psi)}$. By Theorem 4 Part 1, there is an NBW $\mathbb{A}_\Psi = \langle \Sigma', Q^+, q_0^+, \delta^+, G^+\rangle$ accepting all infinite words in Σ'^ω satisfying Ψ. By Theorem 4 Part 2, there is an NBW $\mathbb{A}^{\neg\Psi} = \langle \Sigma', Q^\neg, q_0^\neg, \delta^\neg, G^\neg\rangle$ accepting all infinite words in Σ'^ω that either satisfy $\neg\Psi$ or have a prefix that does. Note that the states of these automata are denoted Q^+ and $Q^\neg$.

The set of states. $Q = Q_1 \cup Q_2$, where $Q_1 = (Q^+ \cup \{\bot\})^g \times (Q^\neg \cup \{\bot\})^g \setminus \{\bot\}^{2g}$, and $Q_2 = \bigcup_{\theta \in \overline{max(\psi)}} Q^\theta$. The Q_1 states are used to run g copies of $\mathbb{A}^{\neg\Psi}$ and g copies of $\mathbb{A}_\Psi$ in parallel. Every state in Q_1 is a vector of $2g$ coordinates where coordinates 1,...,g (called Ψ *coordinates*) contain states of $\mathbb{A}_\Psi$, and coordinates $g+1$,...,$2g$ (called $\neg\Psi$ *coordinates*) contain states of $\mathbb{A}^{\neg\Psi}$. In addition, each coordinate may contain the special symbol $\bot$ indicating that it is *disabled*, as opposed to *active*. We disallow the vector $\{\bot\}^{2g}$ with all coordinates disabled. States in Q_2 are all those from the automata A_θ for every maximal state subformula of ψ, or its negation. These are used to run A_θ whenever A_ϕ guesses that θ holds at a node. Also, for every $1 \leq i \leq g$, we denote by $Q^i_{single} = \{(q_1,...,q_{2g}) \in Q_1 \mid q_i \neq \bot$, and for all $j \leq g$, if $j \neq i$ then $q_j = \bot\}$ 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 $1 \leq i \leq g$ we have that $q_i = q_0^+$ and for every $g+1 \leq i \leq 2g$ we have that $q_i = q_0^\neg$.

The acceptance condition. $B = \cup_{\theta \in \overline{max(\psi)}} B^\theta$ and $G = G' \cup G'' \cup (\cup_{\theta \in \overline{max(\psi)}} G^\theta)$, where $G' = \{(q_1,...,q_{2g}) \in Q^i_{single} \mid q_i \in G^+\}$ is the set of all states in Q_1 in which the only active Ψ coordinate contains a good state, and $G'' = \{(q_1,\ldots,q_{2g}) \in Q_1 \mid \forall i.1 \leq i \leq g \rightarrow q_i = \bot$, and $\exists j. g+1 \leq j \leq 2g \wedge q_j \in G^\neg\}$ is the set of all states in Q_1 in which all the Ψ coordinates are inactive, and some $\neg\Psi$ coordinate contains a good state.

The transition function. δ is defined, for every $\sigma \in \Sigma$, as follows:

- For every $q \in Q_2$, let $\theta \in \overline{max(\psi)}$ be such that $q \in Q^\theta$, and define $\delta(q,\sigma) = \delta^\theta(q,\sigma)$. I.e., for states in Q_2, follow the rules of their respective automata.
- For every $q \in Q_1$, we define $\delta(q,\sigma) := \bigvee_{\sigma' \in \Sigma'}(J \wedge K \wedge L)$ where $J = \bigvee_{X \in Legal(q,\sigma')} \Diamond(X)$, $K = \bigwedge_{\theta \in \sigma'} \delta^\theta(q_0^\theta,\sigma)$, $L = \bigwedge_{\theta \notin \sigma'} \delta^{\neg\theta}(q_0^{\neg\theta},\sigma)$, where $Legal(q,\sigma')$ is the set of all *legal distributions* of (q,σ'), and is defined later.

Informally, the disjunction $\bigvee_{\sigma' \in \Sigma'}$ 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 $\mathbb{A}^{\neg\Psi}$ and $\mathbb{A}_\Psi$ simulated by the states appearing in $Legal(q, \sigma')$ proceed as if the input node was labeled by the letter σ'. The conjunction $(\wedge_{\theta \in \sigma'} \delta^\theta(q_0^\theta, \sigma)) \wedge (\wedge_{\theta \notin \sigma'} \delta^{\neg\theta}(q_0^{\neg\theta}, \sigma))$ ensures that a guess is correct by launching a copy of A_θ for every subformula $\theta \in \sigma'$ that was guessed to hold, and a copy of $\mathsf{A}_{\neg\theta}$ for every subformula θ guessed not to hold.

We define *legal distribution*. 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 $1 \leq i \leq g$ coordinate i (which simulates a copy of $\mathbb{A}_\Psi$) does not get separated from the coordinate $i+g$ (which simulates its partner copy of $\mathbb{A}^{\neg\Psi}$) for as long as i is not the only active Ψ coordinate. 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: $\mathbb{A}_\Psi$ if $j \leq g$, or $\mathbb{A}^{\neg\Psi}$ if $j > g$.

More formally, given a letter $\sigma' \in \Sigma'$, and a state $q = (q_1,...q_{2g}) \in Q_1$ in which the active coordinates are $\{i_1,...,i_k\}$, we say that a sequence $X = q^1,...,q^m$ (for some $m \geq 1$) 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., $\{i_1,...,i_k\} = \cup\{i \in \{1,...,2g\} \mid \exists 1 \leq l \leq m \text{ s.t. } q_i^l \neq \bot\}$. *(ii)* if a coordinate i_j is active in some $q' \in X$ then it is not active in any other $q'' \in X$; *(iii)* if $1 \leq i_j < i_l \leq g$ are two active Ψ coordinates in some $q' \in X$, then $q'_{i_j+g}, q'_{i_l+g} \in Q^\neg \setminus \{\top\}$, i.e., the coordinates $i_j + g, i_l + g$ are also active in q' and do not contain the accepting sink of $\mathbb{A}^{\neg\Psi}$; *(iv)* if i_j is active in some $q' \in X$ then $(q_{i_j}, \sigma', q'_{i_j}) \in \delta^+$ if $i_j \leq g$, and $(q_{i_j}, \sigma', q'_{i_j}) \in \delta^\neg$ if $i_j > g$. I.e., active Ψ coordinates evolve according to the transitions of $\mathbb{A}_\Psi$, and active $\neg\Psi$ coordinates according to the those of $\mathbb{A}^{\neg\Psi}$.

Remark 5. We make two observations. First, the $2g$ copies of $\mathbb{A}^{\neg\Psi}$ and $\mathbb{A}_\Psi$ 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 $\mathbb{A}^{\neg\Psi}$ and $\mathbb{A}_\Psi$ that are active in the current vector may be merged. Unfortunately, this cannot be done since $\mathbb{A}^{\neg\Psi}$ and $\mathbb{A}_\Psi$ 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.

4.2.1. Definition of the Hesitancy Structure $\langle \mathsf{part}, \mathsf{type}, \preceq \rangle$ of A_ϑ

We remind the reader that the hesitancy structure is used to decide membership in a space-efficient way, which is needed for our result that model-checking of GCTL* is in PSPACE.

We now define the hesitancy structure.

1. If $\phi = p \in AP$, then $\mathsf{part} = \{\{q\}\}$, $\mathsf{type}(\{q\}) = trans$, and $\preceq$ is the empty relation.

2. If $\phi = \varphi_0 \vee \varphi_1$ then $\mathsf{part} = \{\{q_0\}\} \cup \mathsf{part}^{\phi_1} \cup \mathsf{part}^{\phi_2}$; $\mathsf{type}(\{q_0\}) = \mathit{trans}$, and for $i \in \{1,2\}$, and every $\mathbf{Q} \in \mathsf{part}^{\phi_i}$, we have $\mathit{type}(\mathbf{Q}) = \mathit{type}^{\phi_i}(\mathbf{Q})$; and $\preceq$ is the union of the relations $\preceq^{\phi_1}, \preceq^{\phi_2}$ as well as the inequalities $\mathbf{Q} \preceq \{q_0\}$ for every $\mathbf{Q} \in \mathsf{part}$. In words, we maintain the partitioning (with the associated types and order) of the states of Q^{φ_0} and Q^{φ_1}, and add a transient set $\{q_0\}$ that it larger then all other sets.
3. If $\phi = \neg\varphi$, then A_ϕ is obtained by dualizing the automaton A_φ, and thus part and $\preceq$ are as in A_φ, and the types are reversed, i.e., every existential set in A_φ becomes universal in A_ϕ, and vice versa.
4. if $\phi = \mathsf{E}^{\geq g}\psi$ then the hesitancy structure is as follows. Given a set of coordinates $I \subseteq \{1,...,2g\}$, let $\mathbf{Q}_I \subseteq Q_1$ be the set of all vectors whose active coordinates are exactly I. We set $\mathsf{type}(\mathbf{Q}_I) = \mathit{exist}$, and set the partitioning part of Q be the union of $\cup_{I \subseteq \{1,..,2g\}}\{\mathbf{Q}_I\}$ and $\cup_{\theta \in \overline{max(\psi)}}(\mathsf{part}^\theta)$. For every $I \subseteq \{1,...,2g\}$, we have $\mathbf{Q}_J \prec \mathbf{Q}_I$ for every $J \subset I$, and $\mathbf{Q} \preceq \mathbf{Q}_I$ for every $\mathbf{Q} \in \cup_{\theta \in \overline{max(\psi)}}(\mathsf{part}^\theta)$. Observe that if a transition $\delta(q,\sigma)$ from some state $q \in \mathbf{Q}_I$, on some letter σ, refers to another state $q' \in \mathbf{Q}_I$ then q was not split (since q' has the same active coordinates as q), i.e., the $\Diamond$ in which q' occurs is of the form $\Diamond(q')$. Hence, by the definition of $\delta(q,\sigma)$, there is no other $q'' \in \mathbf{Q}_I$ that is conjuncted with q' in $\mathit{expand}_d(\delta(q,\sigma))$ for any d, and thus $\delta(q,\sigma)$ respects the hesitancy constraint.

4.3. Depth and Size Analysis

We now analyse the depth, number of states, and size of the transition function of the constructed GHTA A_ϑ.

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

Proof. The depth is clearly $O(|\vartheta|)$.

We analyse the number of states, by cases. As usual, the cases $\vartheta = p$, $\vartheta = \varphi_0 \vee \varphi_1$, and $\vartheta = \neg\varphi$ follow easily from the definitions and the induction hypothesis. For the case $\vartheta = \mathsf{E}^{\geq g}\psi$, the states of the automaton are the union of Q_1 and Q_2.

The set Q_2 uses states from each automaton A_θ for every $\theta \in \overline{max(\phi)}$, and thus $|Q_2|$ is $\Sigma_{\theta \in \overline{max(\psi)}}|A_\theta|$. But by induction $|\mathsf{A}_\theta|$ is $2^{O(|\theta|\cdot \mathsf{deg}(\theta))}$, and so $|Q_2|$ is at most $O(|\psi|) \cdot 2^{O(|\psi|\cdot \mathsf{deg}(\psi))} = 2^{O(|\psi|\cdot \mathsf{deg}(\psi))}$.

The set Q_1 uses a vector of g copies of $\mathbb{A}_\Psi$ and g copies of $\mathbb{A}^{\neg\Psi}$. Thus $|Q_1|$ is $|\mathbb{A}_\Psi|^g \cdot |\mathbb{A}^{\neg\Psi}|^g$. Since $|\mathbb{A}_\Psi|$ and $|\mathbb{A}^{\neg\Psi}|$ are $2^{O(|\psi|)}$, we get that $|Q_1|$ is $2^{O(|\psi|\cdot g)} = 2^{O(|\vartheta|\cdot \mathsf{deg}(\vartheta))}$

Thus, the number of states of A_ϑ is $|Q_1| + |Q_2|$ which is $2^{O(|\vartheta|\cdot \mathsf{deg}(\vartheta))}$.

We treat the size of the transition function similarly. For the case $\phi = \mathsf{E}^{\geq g}\psi$ 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 $\theta \in \overline{max(\psi)}$ be such that $q \in Q^\theta$. Then the length of formula $\delta(q,\sigma)$ is, by induction, at most $2^{O(|\theta|\cdot \mathsf{deg}(\theta))}$. Thus the length of the transitions leaving Q_2 is at most $|Q_2| \cdot 2^{O(|\psi|\cdot \mathsf{deg}(\psi))} = 2^{O(|\vartheta|\cdot \mathsf{deg}(\vartheta))}$.

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

$$\bigvee_{\sigma' \in 2^{max(\psi)}} \Big[(\vee_{X \in Legal(q,\sigma')} \Diamond(X)) \bigwedge (\wedge_{\theta \in \sigma'} \delta^{\theta}(q_0^{\theta}, \sigma)) \bigwedge (\wedge_{\theta \notin \sigma'} \delta^{\neg\theta}(q_0^{\neg\theta}, \sigma)) \Big]$$

has length at most

$$2^{O(|\psi|)} \cdot \left[\Big(\sum_{X \in Legal(q,\sigma')} |\Diamond(X)| \Big) + (\Sigma_{\theta \in \overline{max(\psi)}} 2^{O(|\vartheta| \cdot \mathsf{deg}(\vartheta))}) \right].$$

Now $Legal(q, \sigma')$ 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 $(2^{O(|\psi|)})^{2g} \cdot 2g \cdot (2g)^{2g}$ which is at most $2^{O(|\psi| \cdot \mathsf{deg}(|\psi|))}$.

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 $|\Diamond(X)|$ is at most $|Q_1|^{2g} + k2g \leq |Q_1|^{2g} + (2g)^2 = 2^{O(|\vartheta| \cdot \mathsf{deg}(|\vartheta|)} + O(\mathsf{deg}(|\vartheta|)^2) = 2^{O(|\vartheta| \cdot \mathsf{deg}(|\vartheta|)}$.

Thus the length is at most

$$2^{O(|\psi|)} \cdot \Big[2^{O(|\psi| \cdot \mathsf{deg}(|\psi|))} \cdot 2^{O(|\vartheta| \cdot \mathsf{deg}(|\vartheta|)} + O(|\psi|) \cdot 2^{O(|\vartheta| \cdot \mathsf{deg}(\vartheta))}) \Big]$$

which is $2^{O(|\vartheta| \cdot \mathsf{deg}(\vartheta))}$. □

4.4. Proof of Correctness

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

Lemma 5. *Let A_ϕ be the automaton constructed for $\phi = \mathsf{E}^{\geq g}\psi$, and let $\langle T_r, r \rangle$ be a run of A_ϕ on a tree $\mathsf{T} = \langle T, V \rangle$. For every $1 \leq i \leq 2g$, the set of nodes of T_r in which i is active forms an infinite path π^i from the root. Furthermore, if $\langle T_r, r \rangle$ is an accepting run, then for $1 \leq i \leq g$, there is an $i_k > 1$ such that $state(\pi^i_{i_k - 1})_{i+g} \in Q^{\neg} \setminus \{\top\}$, and for every $l \geq i_k$ the only active Ψ coordinate in $state(\pi^i_l)$ is i.*

Proof. 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 γ of the nodes in I. The induction hypothesis is that there is exactly one node of depth γ in I, and that for $\gamma \geq 1$ it is a son of a node in I. For the base case $\gamma = 0$, the root ε of T_r satisfies $state(\varepsilon) = q_0$, and note that all coordinates, and in particular the i'th coordinate, are active in the initial state q_0.

For $\gamma > 1$, 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 γ in I. Assume by way of contradiction that there are two such nodes $y \neq y' \in I$ of depth γ. 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 x' of 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 $expand_d(\delta(q^x, V(s^x))$ 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^x, that reads the input node s^x, launches (at least) two copies *in parallel*: one in state q^y to the son s^y, and one in state $q^{y'}$ to the son $s^{y'}$. Observe that all transitions from states in Q_1, and thus in particular from q^x, are of the form $\vee_{\sigma' \in \Sigma'}((\vee_{X \in Legal(q,\sigma')} \Diamond(X)) \wedge \Omega_{\sigma'})$, where $\Omega_{\sigma'}$ is a boolean formula that involves only states in Q_2. But this is a contradiction since this implies that for some $\Diamond(X)$ in this transition we have that $q^y, q^{y'} \in X$, which is impossible by property (ii) of a legal distribution.

We now prove that if T_r is accepting, then for $1 \leq i \leq g$ an index i_k as stated by the lemma exists. Since π^i is an infinite path in T_r, and all states in Q_1 belong to existential sets (i.e. sets with $\mathsf{type} = exist$), the fact that T_r is an accepting run implies that for infinitely many $l's$ we have that $state(\pi^i_l) \in G$. Note that all states in G have only one active Ψ coordinate, and that once a coordinate becomes disabled it is never enabled again. Thus, there is a minimal index i_k such that for every $l \geq i_k$ the only active Ψ coordinate in $state(\pi^i_l)$ is i. The fact that $state(\pi^i_{i_k-1})_{i+g} \in Q^\neg \setminus \{\top\}$ follows immediately from the minimality of i_k and property (iii) in the definition of a legal distribution. $\square$

Proof (of correctness of construction for Theorem 3). The proof is by induction on the structure of ϑ and shows that, at each stage of the construction, for every sub-formula ϕ of ϑ, the automaton A_ϕ satisfies the statement of the theorem. The depth, number of states, and size of the transition function are already computed in Proposition 2.

We begin by showing that if $\langle T_r, r\rangle$ is an accepting run of A_ϕ on a tree $\mathsf{T} = \langle T, V\rangle$, then $\mathsf{T} \models \varphi$. The cases $\phi = p$, $\phi = \varphi_0 \vee \varphi_1$, and $\phi = \neg\varphi$ follow easily from the definitions and the induction hypothesis. Consider now the case $\phi = \mathsf{E}^{\geq g}\psi$, and for every $1 \leq i \leq g$, let π^i, and i_k be as given by Lemma 5; furthermore, let $r(\pi^i_{i_k}) = (y_i, q^i)$, and take $y_1, \ldots, y_g$ to be the breakpoints in the statement of Proposition 1. We claim that the conditions of Proposition 1 are satisfied, and thus, it's conclusion also holds, i.e., that $\mathsf{T} \models \varphi$ as required.

First, consider condition (i) of Proposition 1: given $i \neq j$, we have to show that y_i is not a descendant of y_j. Observe that since i is the only active Ψ coordinate in $\pi^i_{i_k}$, and j is the only active Ψ coordinate in $\pi^j_{j_k}$, then $\pi^i_{i_k} \neq \pi^j_{j_k}$. Let $x \in \mathsf{T}_r$ be a node of maximal depth γ, in which both coordinates i and j are active (note that x is well defined since both coordinates are active at the root), and let $r(x) = (s^x, q^x)$. By Lemma 5, x must be a common ancestor of both $\pi^i_{i_k}$

and $\pi^j_{j_k}$, and $\pi^i_{\gamma+1}, \pi^j_{\gamma+1}$ are thus sons of x. Let $r(\pi^i_{\gamma+1}) = (s', q'), r(\pi^j_{\gamma+1}) = (s'', q'')$, and note that by the maximality of γ, in q' coordinate i is active and j is not, and vice versa for q''. Hence, $q' \neq q''$. In other words, the copy of A_ϕ in state q^x, that reads the input node s^x, launches (at least) two *different* copies in parallel: one in state q' to s', and the other in state q'' to s''. Recall that the transition $\delta(q^x, V(s^x))$ is of the form $\vee_{\sigma' \in \Sigma'}((\vee_{X \in Legal(q,\sigma')} \Diamond(X)) \wedge \Omega_{\sigma'})$, where $\Omega_{\sigma'}$ is a boolean formula that involves only states in Q_2 (and thus not q' and q''). By the definition of a run, $\langle T_r, r\rangle$ 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 $s' \neq s''$. Recall that $i_k \geq \gamma + 1$, and $j_k \geq \gamma + 1$, and thus y_i 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 $1 \leq i \leq g$, let $m = i + g$, and take the path π^m guaranteed by Lemma 5. Consider the path $\rho^m = loc(\pi^m_0) \cdot loc(\pi^m_1) \cdots$ in T, of the nodes associated with π^m. For every $l \geq 0$, let $\sigma'_l \in \Sigma'$ be the set of maximal state subformulas of ϕ that hold in ρ^m_l. Applying the induction hypothesis to all $\theta \in \overline{max(\phi)}$, we can conclude that the only way T_r can be accepting is if for every $0 \leq l$ it resolves the outermost disjunction in $\delta(state(\pi^m_l), V(\rho^m_l))$ by taking the disjunct

$$(\vee_{X \in Legal(state(\pi^m_l), \sigma'_l)} \Diamond(X)) \wedge (\wedge_{\theta \in \sigma'_l} \delta^\theta(q^\theta_0, \sigma)) \wedge (\wedge_{\theta \notin \sigma'_l} \delta^{\neg\theta}(q^{\neg\theta}_0, \sigma)).$$

It is thus not hard to see that since T_r is an accepting run of A_ϕ on T, then

$$r' = state(\pi^m_0)_m, state(\pi^m_1)_m, \ldots$$

is an accepting run of $\mathbb{A}^{\neg\Psi}$ on the word $w = \sigma'_0 \cdot \sigma'_1 \cdots \in \Sigma'^\omega$. Note that Lemma 5 implies that $\pi^i_{i_k-1} = \pi^m_{i_k-1}$, and it also states that $state(\pi^i_{i_k-1})_{i+g} \neq \top$. Hence, by Theorem 4 Part 2, some (finite or infinite) prefix $u = \sigma'_0 \cdot \sigma'_1 \cdots$ of w of length at least i_k satisfies $\neg\Psi$. By Lemma 1, it follows that the prefix ϱ of ρ^m, of the same length as u, satisfies $\neg\psi$. Observe that the length of ϱ implies that the path $\rho^m_0 \cdots \rho^m_{i_k-1}$, from the root of T to the father of y_i, is a prefix (possibly not a proper prefix) of ϱ, and is thus not ψ-conservative, as required by condition (ii).

Addressing condition (iii) of Proposition 1 follows in the footsteps of the reasoning used for condition (ii). Given $1 \leq i \leq g$ and the path π^i, the associated path $\rho^i = loc(\pi^i_0) \cdot (\pi^i_1) \cdots$ in T induces the infinite word $w'' = \sigma''_0 \cdot \sigma''_1 \cdots$ who's letters are the sets of maximal state subformulas of ψ that hold along the path ρ^i. By the induction hypothesis, and since T_r is accepting, the run $state(\pi^i_0)_i, state(\pi^i_1)_i, \ldots$ is an accepting run of $\mathbb{A}_\Psi$ on the word w'', and thus by Lemma 1, the path ρ^i satisfies ψ. Since y_i lies on ρ^i, condition (iii) of Proposition 1 is met, and we can conclude that $\mathsf{T} \models \varphi$.

For the other direction, let $\mathsf{T} = \langle T, V\rangle$ be such that $\mathsf{T} \models \varphi$. We have to show that A_ϕ has an accepting run $\langle T_r, r\rangle$ on T. As before, the cases $\phi = p$, $\phi = \varphi_0 \vee \varphi_1$, and $\phi = \neg\varphi$ follow easily from the definitions and the induction

hypothesis. For the case $\phi = \mathsf{E}^{\geq g}\psi$, by Proposition 1, 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 $\rho^i_0, \ldots, \rho^i_{i_k-1}$ is not ψ-conservative, and thus, it can be extended to an infinite path ρ^{i+g} such that some (finite or infinite) prefix of ρ^{i+g} of length $i_n \geq i_k$ satisfies $\neg\psi$. For every node $x \in T$, let $\sigma'(x) \subseteq max(\phi)$ be the set of all maximal state subformulas of ψ that hold in x. By Lemma 1, for every $1 \leq i \leq g$, the infinite word $w^i = \sigma'(\rho^i_0) \cdot \sigma'(\rho^i_1) \cdots$ satisfies Ψ, and the (finite or infinite) word $w^{i+g} = \sigma'(\rho^{i+g}_0) \cdots \sigma'(\rho^{i+g}_{i_n-1})$ satisfies $\neg\Psi$. By Theorem 4 Part 1 there is an accepting run r^i of $\mathbb{A}_\Psi$ on w^i; and by Theorem 4 Part 2 there is an accepting run r^{i+g} of $\mathbb{A}^{\neg\Psi}$ on w^{i+g} for which $r^{i+g}_j \neq \top$ for all $j < i_n$.

We build an accepting run $\langle T_r, r\rangle$ of A_ϕ on T, by induction on the depth γ of the node $x \in T$. At the root ε of T, the automaton is in the initial state q_0, and note that $q_0 \in Q_1$. For the induction step, a copy of A_ϕ in some state $q \in Q_1$, that is at a node $x \in T$ of depth $\gamma \geq 0$ whose labelling $V(x) = \sigma$, proceeds as follows. Recall that $\delta(q, \sigma)$ is $\bigvee_{\sigma' \in \Sigma'}(J \wedge K \wedge L)$ where

$$J = \bigvee_{X \in Legal(q,\sigma')} \Diamond(X) \qquad K = \bigwedge_{\theta \in \sigma'} \delta^\theta(q^\theta_0, \sigma) \qquad L = \bigwedge_{\theta \notin \sigma'} \delta^{\neg\theta}(q^{\neg\theta}_0, \sigma)$$

First, the automaton resolves $\vee_{\sigma' \in \Sigma'}$ by choosing $\sigma' = \sigma'(x)$. By the induction hypothesis (of the theorem), we know that for every $\theta \in \sigma'$ (alternatively $\theta \notin \sigma'$) the automaton A_θ (alternatively $\mathsf{A}_{\neg\theta}$) has an accepting run on the subtree of T rooted at x; thus, by following these accepting runs, A_ϕ can satisfy the conjunction $(\wedge_{\theta \in \sigma'} \delta^\theta(q^\theta_0, \sigma)) \wedge (\wedge_{\theta \notin \sigma'} \delta^{\neg\theta}(q^{\neg\theta}_0, \sigma))$. It remains to show how the automaton handles $\vee_{X \in Legal(q,\sigma')} \Diamond(X)$. For every node $t \in T$, let $live(t) = \{1 \leq i \leq g \mid t \in \rho^i\}$ be the set of all i's for which the path ρ^i goes through t. Let $s_1, \ldots, s_m$ be the sons of x for which these $live()$ sets are not empty. Let $Y = q^1, \ldots, q^m$ be a sequence of states where for every $1 \leq h \leq m$, and every $1 \leq i \leq 2g$, we have that $q^h_i = r^i_{k+1}$ if $i \in live(s_h)$, and $q^h_i = \bot$ otherwise. In words, the i'th coordinate of q^h follows the run r^i if ρ^i goes through the h'th son of x, and is disabled if ρ^i does not go through this son. We claim that Y is a legal distribution of (q, σ'). Indeed, it is not hard to see that $q^1, \ldots, q^m$ are all different, and that properties (i), (ii) and (iv) of a legal distribution are satisfied. As for property (iii), recall that by Proposition 1, if $j, l \in \{1, \ldots, g\}$ and $j \neq l$, then y_j is not a descendant of y_l (and vice-versa). Thus, if both j and l are active in q^h (i.e., both ρ^j and ρ^l go through s_h), it must be that y_j and y_l are both descendants of s_h, and thus $k+1 < j_k$ and $k+1 < l_k$. Recall that for every $1 \leq i \leq g$, the paths ρ^i and ρ^{i+g} coincide at least up to (and including) the father of y_i. Hence, coordinates $j+g$, and $l+g$ must also be active in q^h. Also, recall that for every $1 \leq i \leq g$ we have that $r^{i+g}_m \neq \top$ for all $m < i_n$, and that $i_n \geq i_k$. Thus, in particular, $r^{j+g}_{k+1} = q^h_{j+g} \neq \top$ and $r^{l+g}_{k+1} = q^h_{l+g} \neq \top$, and property (iii) holds, and Y is a legal distribution of (q, σ'). Hence, the automaton can handle $\vee_{X \in Legal(q,\sigma')} \Diamond(X)$ by taking $\Diamond(Y)$, and resolving $\Diamond(Y)$ by sending, for every $1 \leq h \leq m$, a copy in state q^h to the son s_h of x.

We now argue that the run $\langle T_r, r\rangle$ described above is accepting. Let π be

a path in the run tree, and consider the case that for some j we have that $state(\pi_j) \in Q_2$. Take j to be minimal with this property, and observe that it must be that $state(\pi_j) \in Q^\theta$ for some $\theta \in \overline{max(\phi)}$. By our construction of $\langle T_r, r\rangle$, the subtree rooted at π_j is an accepting run of A_θ on the subtree of T rooted at $loc(\pi_j)$, and thus π is an accepting path. Consider now paths for which all states associated with the nodes of the path are in Q_1. By Lemma 5, there are exactly $2g$ such paths $\pi^1, \ldots, \pi^{2g}$, and it is easy to see that by our construction of the run, for every $1 \leq i \leq 2g$, we have that $state(\pi^i_1)_i, state(\pi^i_2)_i, \ldots$ is exactly the run r^i, and that for every $j \geq i_n$ the only active Ψ coordinate in $state(\pi^i_j)$ is i. Hence, by the definition of the acceptance condition of A_ϕ, the path π^i is accepting. This completes the proof of the correctness of the construction. □

5. Complexity of Decision Problems for GCTL*

In this section we establish the complexity of satisfiability, model-checking and realizability/synthesis of GCTL*. We begin by proving that GCTL* has the bounded-degree (in fact, exponential-degree) tree-model property. Recall that Remark 4 only established the finitely-branching tree-model property.

Theorem 5. *A satisfiable* GCTL* *formula ϑ has a tree model of branching degree at most $2^{O(|\vartheta|\cdot \mathsf{deg}(\vartheta))}$.*

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

To prove this claim, we use the membership game $G_{\mathsf{T},\mathsf{A}_\vartheta}$ of the input tree T and the automaton A_ϑ. The game is played by two players, *automaton* and *pathfinder*. Player automaton moves by resolving disjunctions in the transition relation of A_ϑ, and is trying to show that T is accepted by A_ϑ. Player pathfinder moves by resolving conjunctions, and is trying to show that T is not accepted by A_ϑ. The game uses auxiliary tree structured arenas to resolve each transition of the automaton. This is a simple case of a *hierarchical parity game* [4]. As usual, player automaton has a winning strategy if and only if $\mathsf{T} \models \mathsf{A}_\vartheta$. 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_ϑ, 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.

Here are the details. Recall that the automaton A_ϑ is built by recursion on state subformulas (and their negations) of ϑ. In a stage where a subformula ϕ is considered, an automaton is built which consists of some new states as well as the states of automata built from subformulas of ϕ and their negations. Let ϕ_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_p finds its way to the automaton for $(\neg p) \vee q$, because the latter uses states from the dual automaton $\mathsf{A}_{\neg p}$).

Definition of $G_{\mathsf{T},\mathsf{A}_\vartheta}$ (for tree $\mathsf{T} = \langle T, V\rangle$ and formula ϑ). The arena consists of the *main nodes* $Q \times T$, two *sink nodes* $\top, \bot$, 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 $\delta(q, V(t))$. The auxiliary arena $aux(q,t)$ is a finite tree, and when a play π exits this arena it results in a node $exit_\pi(q,t)$ which is either a main node from $(Q \times sons(t)) \cup (Q \times \{t\})$ or a sink node. A play π that visits (q,t), proceeds, via some auxiliary nodes and main nodes of the form $Q \times \{t\}$, to a node $next_\pi(q,t) \in (Q \times sons(t)) \cup \{\bot, \top\}$.

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

- If $\phi_q = p$ for $p \in \mathrm{AP}$, then the game $aux(q,t)$ immediately results in sink node $\top$ if $p \in V(t)$ and in sink node $\bot$ otherwise.
- If $\phi_q = \varphi_0 \vee \varphi_1$, 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_φ.
- If $\phi_q = \mathsf{E}^{\geq g}\psi$, 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^θ, t), or ii) she picks $\theta \notin \sigma'$ and exits at main node $(q_0^{\neg\theta}, t)$, or iii) she transfers play to automaton, in which case automaton picks a legal distribution, say $X = (q_1, \cdots, q_m) \in Legal(q, \sigma')$, 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_ϕ that the transition relation for this case is defined as

$$\bigvee_{\sigma' \in 2^{max(\psi)}} \Big[(\vee_{X \in Legal(q,\sigma')} \Diamond(X)) \bigwedge (\wedge_{\theta \in \sigma'} \delta^\theta(q_0^\theta, \sigma)) \bigwedge (\wedge_{\theta \notin \sigma'} \delta^{\neg\theta}(q_0^{\neg\theta}, \sigma)) \Big].$$

The hesitant acceptance condition of A_ϑ can be easily translated into a parity condition with priorities $\{0,1,2\}$ (also, let sink node $\top$ 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 $G_{\mathsf{T},\mathsf{A}_\vartheta}$. We now continue with the proof. Since ϑ is satisfiable, it is satisfiable by some finitely-branching tree T. Thus, fix a memoryless winning strategy *str* for player automaton in the game $G_{\mathsf{T},\mathsf{A}_\vartheta}$.

Lemma (†)**.** For every main node (q,t) of $G_{\mathsf{T},\mathsf{A}_\vartheta}$, there exists a set $Y(q,t) \subseteq Q \times sons(t)$ such that i) $|Y(q,t)| \leq |Q|$, and ii) every play π consistent with *str*

that exits the arena $aux(q,t)$ with $exit_\pi(q,t) \in Q \times sons(t)$ actually satisfies that $exit_\pi(q,t) \in Y(q,t)$.

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

For the proof, suppose every proper subformula of ϕ satisfies the inductive hypothesis. There are four cases:

- $\phi = p$ for some $p \in \mathrm{AP}$. Define $Y^\phi(q,t) := \emptyset$, and note that the exit node of $aux(q,t)$ is a sink node.
- $\phi = \varphi_1 \vee \varphi_2$. Define $Y^\phi(q,t) := \emptyset$, and note that the exit node of $aux(q,t)$ is a main node of the form $Q \times \{t\}$.
- $\phi = \neg\varphi$. Define $Y^\phi(q,t) := \emptyset$, and note that the exit node of $aux(q,t)$ is a main node of the form $Q \times \{t\}$.
- $\phi = \mathsf{E}^{\geq g}\psi$. The only way to exit $aux(q,t)$ in a main node of the form $Q \times sons(t)$ is via option iii) in the definition of $aux(q,t)$ above; i.e., automaton picks σ', and then pathfinder transfers play to automaton, who then, according to str, picks a legal distribution, say $X = (q_1, \cdots, q_m)$, and corresponding sons of t, say $(s_1, \cdots, s_m)$, and then pathfinder picks an exit of the form (q_i, s_i). Define $Y^\phi(q,t) := \{(q_1,s_1), \cdots, (q_m,s_m)\}$. Since $m \leq |Q^\phi|$ (the components of the legal distribution X are distinct elements of Q^ϕ), we have that $|Y^\phi(q,t)| \leq |Q^\phi|$.

This completes the proof of the Lemma.

We finish the proof of the theorem. Every play consistent with str only visits, besides the main nodes $Q \times \{root\}$ (here $root$ is the root vertex of 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 $G_{\mathsf{T}', \mathsf{A}_\vartheta}$ is a subgame of $G_{\mathsf{T}, \mathsf{A}_\vartheta}$, and player automaton's strategy str is well defined on this subgame, and is winning. Thus $\mathsf{T}' \models \vartheta$. □

Theorem 6. *The satisfiability problem for* GCTL* *over LTSs is 2*ExpTime-Complete. *The model checking problem for* GCTL* *for finite LTSs is* PSpace-Complete.

Proof. The lower bounds already hold for CTL*.

The time upper-bounds for satisfiability follow from Theorems 3, 5 and Lemma 3.

For the space upper-bound for model checking proceed as follows. Given a GCTL* formula ϑ, use Theorem 3 and construct the GHTA A_ϑ, which has $2^{O(|\vartheta|\cdot \mathsf{deg}(\vartheta))}$ states, and transition function of size $2^{O(|\vartheta|\cdot \mathsf{deg}(\vartheta))}$, and which has depth $O(|\vartheta|)$. Then by Lemma 4, the membership problem for A_ϑ and LTS S can be solved in space linear in $|\vartheta|$, quadratic in $|S|$, and polylogarithmic in $2^{O(|\vartheta|\cdot \mathsf{deg}(\vartheta))}$. To finish note that $\mathsf{deg}(\vartheta) \leq |\vartheta|$. □

The automaton constructed in Theorem 3 can serve not only as the basis for solving the satisfiability and model-checking problems of GCTL* as we do in Theorem 6 above, but also for solving other problems for GCTL* for which the automata-theoretic approach was successfully applied to CTL*. Essentially, this usually requires no new ideas, and amounts to using the approach taken for CTL* and modifying it by plugging in an automaton that can handle GCTL* based on the GHTA from Theorem 3. We briefly demonstrate this for (perfect information) realizability and synthesis of GCTL*, showing that the realizability problem for GCTL* is 2ExpTime-Complete, and that one can synthesise a strategy (if one exists) in 2ExpTime. In these problems, sometimes called Church's synthesis, we are given disjoint sets of input variables I and output variables O, and a specification formula φ over atoms $I \cup O$. The realizability problem is to decide if there is a program $f : (2^I)^* \to 2^O$ such that the computation tree of f satisfies φ. The synthesis problem is to return a finite representation of such a program. For more details see [43, 36].

Our automata-theoretic technique allows us to solve these problems for GCTL* specification formulas in a similar way that the automata-theoretic approach solves this problem for simpler CTL* specifications [36]. That is, given a GCTL* formula φ, build a GHTA B that accepts exactly the computation trees satisfying φ (i.e., the 2^I-branching $2^{I\cup O}$-labeled trees satisfying φ) by taking the product of a GHTA that accepts the finitely-branching $2^{I\cup O}$-labeled trees satisfying φ (obtained by Theorem 3) and a GHTA that accepts all 2^I-branching $2^{I\cup O}$-labeled trees. The latter automaton can be easily built as follows: at every node of the tree, check using a $\Diamond$-operator that there are at least $|2^I|$ children each labeled by a different element of 2^I, and check using a $\Box$-operator that there are at most $|2^I|$ children (alternatively, using a conjunction of $\Box$-operators, check that there are no two children labeled with the same element of 2^I). Finally, use Lemma 3 to test if the GHTA B is non-empty and, if so, obtain a regular computation tree witnessing its non-emptiness. Since the size of the automaton B is exponential in the size of φ we get that GCTL*-realizability is 2ExpTime-Complete (for the lower bound, recall that the problem is already 2ExpTime-Hard for CTL* formulas [36]) and GCTL*-synthesis can be done in 2ExpTime.

6. Discussion

We have shown that GCTL* is an expressive logic: it has the tree-model property and is equivalent, over trees, to MPL; and it can express fairness and

counting over paths. Moreover, we have shown that the satisfiability, model-checking and realizability/synthesis problems for GCTL* have the same complexity as that of CTL*.

Our technique suggests a flexible new way to deal with graded path modalities. For instance, our technique immediately recovers the main results about GCTL from [13], i.e., the complexity of satisfiability is ExpTime-Complete and the complexity of model checking is in PTime. Indeed, consider the construction in Theorem 3 of A_ϑ when ϑ it taken from the fragment GCTL of GCTL*, and in particular where it comes to a subformula ϕ of the form $\phi = \mathsf{E}^{\geq g}\psi$. Since ψ is either of the form $p\mathsf{U}q$ or $\mathsf{X}p$, the number of new states added at this stage is a constant. Thus, the number of states of A_ϑ is linear in the size of ϑ. Also, our results immediately apply to counting-CTL*(by Remark 3) and show that the complexity of satisfiability is 2ExpTime-Complete and the complexity of model checking is PSpace-Complete.[9]

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 [16]. As pointed out in [34], this reflects the way in which many decision procedures for these logics work: they explicitly generate n individuals for $\exists^{\geq n}$. Second, although the complexity of the binary case is sometimes the same as that of the unary case, the constructions are significantly more complicated, and are thus much harder to implement [15, 14]. At any rate, as the binary case is useful in some circumstances we plan to investigate this in the future.

Comparison with (some) other 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 another logic that would yield the optimal 2ExpTime upper bound. We discuss two such candidate translations. 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 ExpTime-Complete) would yield the optimal complexity since a) already the usual translation from CTL* to μ-calculus does not yield optimal complexity [24], and b) the translation given in [14] from GCTL to $G\mu$-calculus does not yield optimal complexity. Moreover, the usual translation from CTL* to μ-calculus uses automata, and thus automata for GCTL* (from which we get our results directly) have to be developed anyway.

Future work. First, recall that logics extended with graded world modali-

[9] It was conjectured in [14] that a) GCTL* has the same expressive power as counting-CTL*, and b) GCTL* is exponentially more succinct than counting-CTL*. The proof of Theorem 1 confirms a), while b) is still open.

ties have been further enriched with backward-modalities and with nominals [15]. A similar direction can be taken for graded path modalities, and GCTL* in particular. Second, recall that the graded μ-calculus was used to solve questions (such as satisfiability) for the description logic $\mu\mathcal{ALCQ}$ [15]. Similarly, our techniques for GCTL* might be useful for solving questions in $\mathcal{ALCQ}$ combined with temporal logic, such as for the graded extension of CTL$^*_{\mathcal{ALC}}$ [31]. Third, graded strategy modalities that generalize the graded path modalities in this work are studied in [37], and in particular the complexity of model checking a graded extension of ATL is established as being PTIME-complete. We believe our techniques can be extended to deal with a graded extension of ATL*.

References

[1] Graded modalities in strategy logic. *Information and Computation*, IN PRESS:–, 2018.

[2] S. Almagor, U. Boker, and O. Kupferman. What's decidable about weighted automata? In *ATVA*, pages 482–491, 2011.

[3] B. Aminof, O. Kupferman, and R. Lampert. Rigorous approximated determinization of weighted automata. In *Symposium on Logic in Computer Science*, pages 345–354, 2011.

[4] B. Aminof, O. Kupferman, and A. Murano. Improved model checking of hierarchical systems. *Information and Computation*, 210:68–86, 2012.

[5] Benjamin Aminof, Vadim Malvone, Aniello Murano, and Sasha Rubin. Graded strategy logic: Reasoning about uniqueness of nash equilibria. In Catholijn M. Jonker, Stacy Marsella, John Thangarajah, and Karl Tuyls, editors, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 698–706. ACM, 2016.

[6] Benjamin Aminof, Aniello Murano, and Sasha Rubin. On CTL* with graded path modalities. In Martin Davis, Ansgar Fehnker, Annabelle McIver, and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings*, volume 9450 of *Lecture Notes in Computer Science*, pages 281–296. Springer, 2015.

[7] M. Arenas, P. Barceló, and L. Libkin. Combining Temporal Logics for Querying XML Documents. In *ICDT*, LNCS 4353, pages 359–373, 2007.

[8] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal conjunctive queries in expressive description logics with transitive roles. In Bernhard Pfahringer and Jochen Renz, editors, *AI 2015: Advances in Artificial Intelligence - 28th Australasian Joint Conference, Canberra, ACT, Australia, November 30 - December 4, 2015, Proceedings*, volume 9457 of *Lecture Notes in Computer Science*, pages 21–33. Springer, 2015.

[9] Pablo Barceló, Leonid Libkin, and Juan L. Reutter. Querying regular graph patterns. *J. ACM*, 61(1):8:1–8:54, 2014.

[10] Everardo Bárcenas, Edgard Benítez-Guerrero, and Jesús Lavalle. On the model checking of the graded μ-calculus on trees. In Grigori Sidorov and Sofía N. Galicia-Haro, editors, *Advances in Artificial Intelligence and Soft Computing - 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Cuernavaca, Morelos, Mexico, October 25-31, 2015, Proceedings, Part I*, volume 9413 of *Lecture Notes in Computer Science*, pages 178–189. Springer, 2015.

[11] Everardo Bárcenas and Jesús Lavalle. Global numerical constraints on trees. *Logical Methods in Computer Science*, 10(2), 2014.

[12] Everardo Bárcenas, Guillermo Molero, Gabriela Sánchez, Edgard Benítez-Guerrero, and Carmen Mezura-Godoy. Reasoning on expressive description logics with arithmetic constraints. In *2016 International Conference on Electronics, Communications and Computers, CONIELECOMP 2016, Cholula, Mexico, February 24-26, 2016*, pages 180–185. IEEE, 2016.

[13] A. Bianco, F. Mogavero, and A. Murano. Graded computation tree logic. In *Symposium on Logic in Computer Science*, pages 342–351. IEEE, 2009.

[14] A. Bianco, F. Mogavero, and A. Murano. Graded computation tree logic. *ACM Transactions on Computational Logic*, 13(3), 2012.

[15] P.A. Bonatti, C. Lutz, A. Murano, and M.Y. Vardi. The complexity of enriched mu-calculi. *Logical Methods in Computer Science*, 4(3):, 2008.

[16] D. Calvanese, G. De Giacomo, and M. Lenzerini. Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In *International Joint Conference on Artificial Intelligence*, pages 84–89, 1999.

[17] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. Node selection query languages for trees. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

[18] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics via alternating tree-automata. *Inf. Comput.*, 237:12–55, 2014.

[19] Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 854–860. IJCAI/AAAI, 2013.

[20] M. de Rijke. A note on graded modal logic. *Studia Logica*, 64(2):271–283, 2000.

[21] M. Droste, W. Kuich, and H. Vogler. *Handbook of Weighted Automata*. Springer, 2009.

[22] C. Eisner, D.Fisman, J. Havlicek, Y. Lustig, A. McIsaac, and D. V. Campenhout. Reasoning with temporal logic on truncated paths. In *Computer Aided Verification*, LNCS 2725, pages 27–39, 2003.

[23] E.A. Emerson and C.S. Jutla. The complexity of tree automata and logics of programs. *SIAM Journal of Computing*, 29(1):132–158, 1999.

[24] E.A. Emerson and A.P. Sistla. Deciding branching time logic. In *Symposium on Theory of Computing*, pages 14–24, 1984.

[25] Cristina Feier and Thomas Eiter. Reasoning with forest logic programs using fully enriched automata. In Francesco Calimeri, Giovambattista Ianni, and Miroslaw Truszczynski, editors, *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *Lecture Notes in Computer Science*, pages 346–353. Springer, 2015.

[26] A. Ferrante, A. Murano, and M. Parente. Enriched mu-calculi module checking. *Logical Methods in Computer Science*, 4(3):, 2008.

[27] A. Ferrante, M. Napoli, and M. Parente. Model Checking for Graded CTL. *Fundamenta Informaticae*, 96(3):323–339, 2009.

[28] Alessandro Ferrante, Margherita Napoli, and Mimmo Parente. CTL model-checking with graded quantifiers. In *Automated Technology for Verification and Analysis, 6th International Symposium, ATVA 2008, Seoul, Korea, October 20-23, 2008. Proceedings*, pages 18–32, 2008.

[29] Alessandro Ferrante, Margherita Napoli, and Mimmo Parente. Graded-CTL: Satisfiability and symbolic model checking. In *Formal Methods and Software Engineering, 11th International Conference on Formal Engineering Methods, ICFEM 2009, Rio de Janeiro, Brazil, December 9-12, 2009. Proceedings*, pages 306–325, 2009.

[30] K. Fine. In So Many Possible Worlds. *Notre Dame Journal of Formal Logic*, 13:516–520, 1972.

[31] V. Gutiérrez-Basulto, J.C. Jung, and C. Lutz. Complexity of branching temporal description logics. In *European Conference on Artificial Intelligence*, pages 390–395, 2012.

[32] T.A. Henzinger. Quantitative reactive modeling and verification. *Comput. Sci.*, 28(4):331–344, November 2013.

[33] Ian Horrocks and Ulrike Sattler. Decidability of SHIQ with complex role inclusion axioms. *Artif. Intell.*, 160(1-2):79–104, 2004.

[34] O. Kupferman, U. Sattler, and M.Y. Vardi. The Complexity of the Graded μ-Calculus. In *Conference on Automated Deduction*, LNCS 2392, pages 423–437. Springer, 2002.

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

[36] Orna Kupferman and Moshe Y. Vardi. Church's problem revisited. *Bulletin of Symbolic Logic*, 5(2):245–263, 1999.

[37] Vadim Malvone, Fabio Mogavero, Aniello Murano, and Loredana Sorrentino. On the counting of strategies. In Fabio Grandi, Martin Lange, and Alessio Lomuscio, editors, *22nd International Symposium on Temporal Representation and Reasoning, TIME 2015, Kassel, Germany, September 23-25, 2015*, pages 170–179. IEEE Computer Society, 2015.

[38] Vadim Malvone, Fabio Mogavero, Aniello Murano, and Loredana Sorrentino. Reasoning about graded strategy quantifiers. *Inf. Comput.*, 259(Part):390–411, 2018.

[39] Vadim Malvone, Aniello Murano, and Loredana Sorrentino. Games with additional winning strategies. In Davide Ancona, Marco Maratea, and Viviana Mascardi, editors, *Proceedings of the 30th Italian Conference on Computational Logic, Genova, Italy, July 1-3, 2015.*, volume 1459 of *CEUR Workshop Proceedings*, pages 175–180. CEUR-WS.org, 2015.

[40] Vadim Malvone, Aniello Murano, and Loredana Sorrentino. Additional winning strategies in reachability games. *Fundam. Inform.*, 159(1-2):175–195, 2018.

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

[42] Magdalena Ortiz and Mantas Simkus. Reasoning and query answering in description logics. In Thomas Eiter and Thomas Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, volume 7487 of *Lecture Notes in Computer Science*, pages 1–53. Springer, 2012.

[43] Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*, pages 179–190. ACM Press, 1989.

[44] Lutz Schröder and Dirk Pattinson. How many toes do I have? parthood and number restrictions in description logics. In Gerhard Brewka and Jérôme Lang, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 307–317. AAAI Press, 2008.

[45] S. Tobies. PSPACE Reasoning for Graded Modal Logics. *Journal of Logic and Computation*, 11(1):85–106, 2001.

[46] W. van der Hoek and JJ.Ch. Meyer. Graded modalities in epistemic logic. In *Symposium on Logical Foundations of Computer Science*, pages 503–514, 1992.

[47] M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Banff Higher Order Workshop*, pages 238–266, 1995.

[48] M. Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.

[49] I. Walukiewicz. Monadic second-order logic on tree-like structures. *Theor. Comput. Sci.*, 275(1-2):311–346, 2002.

Appendix A. Proof of Lemma 1.

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

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

So suppose now that ψ 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 $\widehat{\psi} = \neg\widehat{\gamma}$, and so
$$\begin{aligned}(\mathsf{S}, \pi) \models \psi &\text{ iff } (\mathsf{S}, \pi) \not\models \gamma \\ &\text{iff } (\mathsf{S}_\gamma, \pi) \not\models \widehat{\gamma} \\ &\text{iff } (\mathsf{S}_{\neg\gamma}, \pi) \not\models \widehat{\gamma} \\ &\text{iff } (\mathsf{S}_{\neg\gamma}, \pi) \models \neg\widehat{\gamma} \\ &\text{iff } (\mathsf{S}_{\neg\gamma}, \pi) \models \widehat{\psi}.\end{aligned}$$
2. Suppose $\psi = \mathsf{X}\gamma$. Then, $max(\psi) = max(\gamma)$, and so $\widehat{\psi} = \mathsf{X}\widehat{\gamma}$, and so
$$\begin{aligned}(\mathsf{S}, \pi) \models \psi &\text{ iff } (\mathsf{S}, \pi_{\geq 1}) \models \gamma \text{ and } |\pi| > 1 \\ &\text{iff } (\mathsf{S}_\gamma, \pi_{\geq 1}) \models \widehat{\gamma} \text{ and } |\pi| > 1 \\ &\text{iff } (\mathsf{S}_{\mathsf{X}\gamma}, \pi) \models \widehat{\gamma} \text{ and } |\pi| > 1 \\ &\text{iff } (\mathsf{S}_{\mathsf{X}\gamma}, \pi) \models \mathsf{X}\widehat{\gamma} \\ &\text{iff } (\mathsf{S}_{\mathsf{X}\gamma}, \pi) \models \widehat{\psi}.\end{aligned}$$
3. Suppose $\psi = \gamma\mathsf{U}\zeta$. Then, $max(\psi) = max(\gamma) \cup max(\zeta)$, and so $\widehat{\psi} = \widehat{\gamma}\mathsf{U}\widehat{\zeta}$. Then: $(\mathsf{S}, \pi) \models \psi$, if and only if, $\exists i.0 \leq i < |\pi|.(\mathsf{S}, \pi_{\geq i}) \models \zeta$ and $\forall j.0 \leq j < i, (\mathsf{S}, \pi_{\geq j}) \models \gamma$, if and only if, $\exists i.0 \leq i < |\pi|.(\mathsf{S}_\zeta, \pi_{\geq i}) \models \widehat{\zeta}$ and $\forall j.0 \leq j < i, (\mathsf{S}_\gamma, \pi_{\geq j}) \models \widehat{\gamma}$, if and only if, $\exists i.0 \leq i < |\pi|.(\mathsf{S}_\psi, \pi_{\geq i}) \models \widehat{\zeta}$ and $\forall j.0 \leq j < i, (\mathsf{S}_\psi, \pi_{\geq j}) \models \widehat{\gamma}$, which is equivalent to $(\mathsf{S}_\psi, \pi) \models \widehat{\gamma}\mathsf{U}\widehat{\zeta}$, which is equivalent to $(\mathsf{S}_\psi, \pi) \models \widehat{\psi}$.

The other cases, $\vee$ and R, are similar to the U case. □

Appendix B. Proof of Part 2 of Lemma 2.

PROOF. Notation. We use '$\cdot$' for concatenation of paths in S^t, and we use adjacency or '.' for concatenation of paths in S.

The statement we want to prove is this: for all GCTL* formulas ϕ (these are state formulas), and all LTSs S, and all $t \in S$, $(\mathsf{S}, t) \models \phi$ if and only if $(\mathsf{S}^t, t) \models \phi$.

We first define a natural bijection f between $\mathsf{pth}(\mathsf{S}, t)$ and $\mathsf{pth}(\mathsf{S}^t, t)$. Define $f(t) := t$, and $f(\pi_0 \dots \pi_n r) := f(\pi_0 \dots \pi_n) \cdot \pi_0 \dots \pi_n r$, and $f(\pi_0\pi_1 \dots) := \pi_0 \cdot \pi_0\pi_1 \cdot \pi_0\pi_1\pi_2 \cdot \dots$.

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

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

- if α is a state formula then $(\mathsf{S}, t) \models \alpha$ iff $(\mathsf{S}^t, t) \models \alpha$,
- if α is a path formula then $(\mathsf{S}, \pi) \models \alpha$ iff $(\mathsf{S}^t, f(\pi)) \models \alpha$.

Suppose the inductive hypothesis holds for all proper subformulas of α. Fix S, t, π. There are two main cases.

Suppose α is a state formula. There are three cases.

1. Suppose α is of the form p for $p \in \mathrm{AP}$. In this case we must prove that $(\mathsf{S}, t) \models p$ iff $(\mathsf{S}^t, t) \models p$, which is immediate from the definition of λ'.
2. The case that $\alpha = \neg\varphi_1$ or $\alpha = \varphi_1 \vee \varphi_2$ for state formulas φ_i is immediate from the semantics of these Boolean operations and the inductive hypothesis.
3. Suppose α is of the form $\mathsf{E}^{\geq g}\psi$ for path formula ψ.
   For the first direction, suppose $(\mathsf{S}, t) \models \alpha$, i.e., there are at least g many minimal ψ-conservative paths in $\mathsf{pth}(\mathsf{S}, t)$. In other words, there exists distinct $\pi_1, \cdots, \pi_g \in \mathsf{pth}(\mathsf{S}, t)$ such that for every i, we have
   (a) Every extension $\pi' \in \mathsf{pth}(\mathsf{S}, t)$ of π_i satisfies $(\mathsf{S}, \pi') \models \psi$.
   (b) Every prefix τ of π_i has an extension $\rho \in \mathsf{pth}(\mathsf{S}, t)$ that satisfies $(\mathsf{S}, \rho) \not\models \psi$.

   Thus: $f(\pi_1), \cdots, f(\pi_g) \in \mathsf{pth}(\mathsf{S}^t, t)$ are distinct, and for every i we have:
   (a) Every extension $\pi' \in \mathsf{pth}(\mathsf{S}^t, t)$ of $f(\pi_i)$ satisfies $(\mathsf{S}^t, \pi') \models \psi$. To see this note that $\pi' = f(\pi)$ for some $\pi \in \mathsf{pth}(\mathsf{S}, t)$, and so $f(\pi_i) \preceq f(\pi)$, and so $\pi_i \preceq \pi$, and so by 3a. $(\mathsf{S}, \pi) \models \psi$, and so by induction $(\mathsf{S}^t, f(\pi)) \models \psi$.
   (b) Every prefix τ' of $f(\pi_i)$ has an extension $\rho' \in \mathsf{pth}(\mathsf{S}^t, t)$ that satisfies $(\mathsf{S}^t, \rho') \not\models \psi$ (use similar reasoning to the previous case).

   Thus $(\mathsf{S}^t, t) \models \mathsf{E}^{\geq g}\psi$, and this completes the first direction. The other direction, i.e., that $(\mathsf{S}^t, t) \models \mathsf{E}^{\geq g}\psi$ implies $(\mathsf{S}, t) \models \mathsf{E}^{\geq g}\psi$ is done by simply reversing the argument for the first direction.

Suppose α is a path formula, say ψ. Let Ψ be the LTL formula from Lemma 1. Then

$$\begin{aligned}(S, \pi) \models \psi &\text{ iff } (S_\psi, \pi) \models \Psi \\ &\text{iff } ((S_\psi)^t, f(\pi)) \models \Psi \\ &\text{iff } ((S^t)_\psi, f(\pi)) \models \Psi \\ &\text{iff } (S^t, f(\pi)) \models \psi.\end{aligned}$$

The first and fourth equivalences follow from Lemma 1, the second and third equivalences follows from inductive hypothesis applied to the maximal state sub-formulas of ψ and the fact that $\pi \in S_\psi$ and $f(\pi) \in (S_\psi)^t$ and $f(\pi) \in (S^t)_\psi$ induce the same infinite sequence of labels (and thus the paths agree on the LTL formula Ψ). $\square$

Appendix C. Proof of Theorem 4.

PROOF. The existence of $\mathbb{A}_\zeta$ is shown in [47, Corollary 23]. The existence of $\mathbb{B}_\zeta$ is proved by a simple adaptation of the construction in [47, Theorem 22] (given below, or see [19] for the translation of Linear Dynamic Logic on finite traces to alternating finite automata). This yields an alternating finite word automaton $\mathbb{B}'_\zeta$ (of linear size) accepting all finite paths that satisfy ζ. This automaton is then converted $\mathbb{B}'_\zeta$ to an equivalent NFW $\mathbb{B}_\zeta$ (using [47, Proposition 16]), of size $2^{O(\zeta)}$.

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

1. if $\varphi = eow$, then $\delta(\varphi, a) = \mathbf{false}$;
2. if $\varphi = p$ for $p \in AP$, then $\delta(\varphi, a) = \mathbf{true}$ if $p \in a$, and $\delta(\varphi, a) = \mathbf{false}$ otherwise;
3. if $\varphi = \neg p$ for $p \in AP$, then $\delta(\varphi, a) = \mathbf{false}$ if $p \in a$, and $\delta(\varphi, a) = \mathbf{true}$ otherwise;
4. If $\varphi = \varphi_1 \dagger \varphi_2$, for $\dagger \in \{\vee, \wedge\}$, then $\delta(\varphi, a) = \delta(\varphi_1, a) \dagger \delta(\varphi_2, a)$;
5. If $\varphi = \neg(\varphi_1 \dagger \varphi_2)$, for $\dagger \in \{\vee, \wedge\}$, then $\delta(\varphi, a) = \delta(\neg\varphi_1, a) \ddagger \delta(\neg\varphi_2, a)$, where $\ddagger$ is the dual of $\dagger$, i.e., $\ddagger = \vee$ if $\dagger = \wedge$, and $\ddagger = \wedge$ if $\dagger = \vee$;
6. if $\varphi = \mathsf{X}\theta$, then $\delta(\varphi, a) = \theta$;
7. if $\varphi = \neg\mathsf{X}\theta$, then $\delta(\varphi, a) = eow \vee \neg\theta$;
8. if $\varphi = \varphi_1 \mathsf{U} \varphi_2$, then $\delta(\varphi, a) = \delta(\varphi_2, a) \vee (\delta(\varphi_1, a) \wedge \delta(\mathsf{X}(\varphi_1 \mathsf{U} \varphi_2), a))$;
9. if $\varphi = \neg(\varphi_1 \mathsf{U} \varphi_2)$, then $\delta(\varphi, a) = \delta(\neg\varphi_2, a) \wedge (\delta(\neg\varphi_1, a) \vee \delta(\neg\mathsf{X}(\varphi_1 \mathsf{U} \varphi_2), a))$.
10. if $\varphi = \varphi_1 \mathsf{R} \varphi_2$, then $\delta(\varphi, a) = (\delta(\varphi_1, a) \wedge \delta(\varphi_2, a)) \vee (\delta(\varphi_2, a) \wedge \delta(\mathsf{X}(\varphi_1 \mathsf{R} \varphi_2), a))$;
11. if $\varphi = \neg(\varphi_1 \mathsf{R} \varphi_2)$, then $\delta(\varphi, a) = \delta(\neg\varphi_2, a) \vee (\delta(\neg\varphi_1, a) \wedge \delta(\neg\mathsf{X}(\varphi_1 \mathsf{R} \varphi_2), a))$.

By defining the transition relation rules for the cases of $\mathsf{X}, \mathsf{U}, \mathsf{R}$ and their negations using one-step unfolding, the adaptation of the construction in [47, Theorem 22] to the finite words semantics addressed here is confined to the definition of the set of accepting states, and the transitions from *eow* and $\neg\mathsf{X}$. In order to accept $\neg\mathsf{X}\theta$, the automaton can either guess that the input word has ended and go to the accepting state *eow*, or (by going to the state $\neg\theta$) guess that the input has not ended and that the remaining suffix of the input word does not satisfy θ. Having $\delta(eow, a) = \mathbf{false}$ ensures that if the automaton guessed that the input has ended, then any further input would result in a rejecting run. This completes the proof of the first part.

For the second part, consider the NBW and the NFW from Part 1, i.e., the NBW $\mathbb{A}_\zeta = \langle \Sigma, Q, q_0, \delta, G \rangle$ and the NFW $\mathbb{B}_\zeta = \langle \Sigma, Q', q'_0, \delta', F \rangle$. Assume w.l.o.g. that Q, Q' are disjoint (and do not contain $\top$, $\widetilde{q_0}$) and construct from them a single NBW $\mathbb{A}^\zeta = \langle \Sigma, Q \cup Q' \cup \{\top, \widetilde{q_0}\}, \widetilde{q_0}, \delta'', G \cup \{\top\} \rangle$, where δ'' is the

union of δ and δ' as well as the transitions $(\widetilde{q_0}, \sigma, q)$ for every σ and q such that $(q_0, \sigma, q) \in \delta$ or $(q'_0, \sigma, q) \in \delta'$; $(\top, \sigma, \top)$ 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 $\mathbb{A}_\zeta$ and $\mathbb{B}_\zeta$, adding a new accepting sink state $\top$, and matching any transition that goes to a final state of $\mathbb{B}_\zeta$ with a transition that goes to the accepting sink $\top$. It is not hard to see that this construction yields the desired automaton. $\square$