

Graded Computation Tree Logic

Alessandro Bianco Fabio Mogavero Aniello Murano
Università degli Studi di Napoli "Federico II", 80126 Napoli, Italy.
{alessandrobiano, mogavero, murano}@na.infn.it
<http://people.na.infn.it/~{alessandrobiano, mogavero, murano}>

Abstract—In modal logics, *graded (world) modalities* have been deeply investigated as a useful framework for generalizing standard existential and universal modalities in such a way that they can express statements about a given number of immediately accessible worlds. These modalities have been recently investigated with respect to the μ -calculus, which have provided succinctness, without affecting the satisfiability of the extended logic, i.e., it remains solvable in EXPTIME. A natural question that arises is how logics that allow reasoning about paths could be affected by considering *graded path modalities*. In this paper, we investigate this question in the case of the branching-time temporal logic CTL (GCTL, for short). We prove that, although GCTL is more expressive than CTL, the satisfiability problem for GCTL remains solvable in EXPTIME. This result is obtained by exploiting an automata-theoretic approach. In particular, we introduce the class of *partitioning alternating Büchi tree automata* and show that the emptiness problem for them is EXPTIME-COMPLETE. The satisfiability result turns even more interesting as we show that GCTL is exponentially more succinct than graded μ -calculus.

Keywords-Temporal logics, graded modalities, minimality, conservativeness, satisfiability, automata-theoretic approach.

I. INTRODUCTION

Temporal logics are a special kind of *modal logics* that provide a formal framework for qualitatively describing and reasoning about how the truth values of assertions change over time. First pointed out by Pnueli in 1977 [27], these logics turn out to be particularly suitable for reasoning about correctness of concurrent programs [28].

Depending on the view of the underlying nature of time, two types of temporal logics are mainly considered [20]. In *linear-time temporal logics*, such as LTL [27], time is treated as if each moment in time has a unique possible future. Conversely, in *branching-time temporal logics*, such as CTL [10] and CTL* [12], each moment in time may split into various possible futures and *existential* and *universal quantifiers* are used to express properties along one or all the possible futures. In modal logics, such as \mathcal{ALC} [29] and μ -calculus [17], these kinds of quantifiers have been generalized by means of *graded (worlds) modalities* [14], [31], which allow to express properties such as “there exist at least n accessible worlds satisfying a certain formula” or “all but n accessible worlds satisfy a certain formula”. For example, in a multitasking scheduling specification, we

can express properties such as every time a computation is invoked, immediately next there are at least two spaces available for the allocation of two tasks that take care of the computation, without expressing exactly which spaces they are. This generalization has been proved to be very powerful as it allows to express system specifications in a very succinct way. In some cases, the extension makes the logic much more complex. An example is the guarded fragment of the first order logic, which becomes undecidable when extended with a very weak form of counting quantifiers [15]. In some other cases, one can extend a logic with very strong forms of “counting quantifiers” without increasing the computational complexity of the obtained logic. For example, this is the case for $\mu\mathcal{ALCQ}$ (see [3] for a recent handbook) and graded μ -calculus [18], [6], for which the decidability problem is EXPTIME-COMPLETE.

Despite its high expressive power, the μ -calculus is considered in some sense a low-level logic, making it an “unfriendly” logic for users, whereas simpler logics, such as CTL, can naturally express complex properties of computation trees. Therefore, an interesting and natural question that arises is how the extension of CTL with graded modalities can affect its expressiveness and decidability. There is a technical challenge involved in such an extension, which makes this task non-trivial: in the μ -calculus, and other modal logics studied in the graded context so far, the existential and universal quantifiers range over the set of successors, thus it is easy to count the domain and its elements. In CTL, on the other hand, the underlying objects are both states and paths. Thus, the concept of graded must relapse on both of them. We solve this problem by introducing *graded path modalities* that extend to *minimal* and *conservative* paths the generalization induced to successor worlds by classical graded modalities, i.e., they allow to express properties such as “there are at least n minimal and conservative paths satisfying a formula”, for suitable and well-formed concepts of minimality and conservativeness among paths. We call the introduced logic GCTL, for short. Note that in this framework, a state can have only one direct successor but more than one different path going through it. This must be taken into account while satisfying a given graded path property. To deal with this difficulty, we use a combinatorial tool: the partitioning of a natural number k , that is, we consider all possible decompositions of k into summands

(i.e., $3 = 3 + 0 = 2 + 1 = 1 + 1 + 1$). This is used to distribute k different paths emerging from a state onto all its direct successors. Note that, while CTL linearly translates to μ -calculus, the above complication makes the translation of GCTL to graded μ -calculus not easy at all. Indeed, we show such a translation with an exponential blow-up, by taking into account the above path partitioning. The minimality property allows to decide GCTL formulas on a restricted but significant space domain, i.e., the set of paths of interest, in a very natural way. In more detail, it is enough to consider only the part of a system behavior that is effectively responsible for the satisfiability of a given formula, whenever each of its extensions satisfies the formula as well. So, we only take into account a set of non-comparable paths satisfying the same property using in practice a particular equivalence relation on the set of all paths. Moreover, the minimality allows the graded path modalities to subsume the graded world modalities introduced for the μ -calculus. Indeed, if we drop the minimality, it makes no sense to discuss the existence of a path in a structure, where the existence of a non-minimal path satisfying a formula may induce also the existence of an infinite number of paths satisfying it.

With GCTL it is possible to express properties of a number of (not immediate) successor worlds, in a very succinct way, without explicitly stating properties of the intermediate worlds. As an example, consider the property “in a tree, there exists a path in which everytime p holds at a given node x , n grandchildren of x satisfy q ”. This property can easily be expressed in GCTL (linearly in n). Conversely, a graded μ -calculus formula would require to consider all possible children scenarios (i.e., all partitions of node successors) of p , and therefore it needs a length exponential in n . We also prove that this exponential blow-up is unavoidable. In particular, the idea of counting paths on a tree, behind the previous example, is the core of the proof we use. As another and more practical example of an application of GCTL, consider again the above multitasking scheduling, where we may want to check that every time a non-elementary (i.e., non one-step) computation is required, then there are at least n distinct (i.e., non completely equivalent) non-redundant computational flows that can be executed. This property can be easily expressed in GCTL thanks to the concepts of minimality and conservativeness.

The ability of GCTL to reason about numbers of paths turns out to be suitable in several contexts. For example, it can be useful to query XML documents [2], [21]. These documents, indeed, can be viewed as labeled unranked trees [4] and GCTL allows reasoning about a number of links among tags of descendant nodes, without naming any of the intermediate ones, in a very succinct way. In particular, it is possible to verify how-many minimal data-paths satisfying a given query exist in an XML document. We also note that our framework of graded path quantifiers has some similarity with the concept of *cyclomatic complexity*, as it was defined

by McCabe in a seminal work in software engineering [22]. McCabe studied a way to measure the complexity of a program, identifying it in the number of independent instruction flows. From an intuitive point of view, since graded path quantifiers allow to specify how many minimal computational paths satisfying a given property reside in a program, GCTL subsumes the cyclomatic complexity, where for independent we replace minimal.

The introduced framework of graded path modalities turns out to be very efficient in terms of expressiveness and complexity. Indeed, we prove that GCTL is more expressive than CTL, it retains the tree and the finite model properties, and its satisfiability problem is solvable in EXPTIME, therefore not harder than that for CTL [11]. This, along with the fact that GCTL is exponentially more succinct than graded μ -calculus, makes GCTL even more appealing. The upper bound for the satisfiability complexity result is obtained by exploiting an automata-theoretic approach [32], [19]. To develop a decision procedure for a logic with the tree model property, one first develops an appropriate notion of tree automata and studies their emptiness problem. Then, the satisfiability problem for the logic is reduced to the emptiness problem of the automata. To this aim, we introduce a new automaton model: *partitioning alternating tree automata* (PATA). While a nondeterministic automaton on visiting a node of the input tree sends exactly one copy of itself to each successor of the node, an alternating automaton can send several copies of itself to the same successor. In particular, in *symmetric alternating automata* [16], [33] it is not necessary to specify the direction of the tree on which a copy is sent. In [18], *graded alternating tree automata* (GATA) are introduced as a generalization of symmetric alternating tree automata, in such a way that the automaton can send copies of itself to a given number n of state successors, either in existential or universal way, without specifying which successors these exactly are. PATA further extend GATA in such a way that the automaton can send copies of itself to a given number n of paths. As we show, for each GCTL formula φ , it is always possible to build in linear time a PATA \mathcal{A}_φ along with a Büchi condition (PABT) accepting all the tree models of φ . The major difficulty here is that whenever φ contains graded modalities such as “there exist at least n minimal paths satisfying a path property ψ ”, \mathcal{A}_φ must accept trees in which there are at least n distinct paths satisfying ψ , where some groups of those paths can arbitrarily share the same (proper) prefixes, and we ensure this by constraining the transition relation of the automaton. We show an EXPTIME decision procedure for the emptiness of PABT through an exponential translation into nondeterministic Büchi tree automata (NBT). In more detail, we use a variation of the Miyano and Hayashi technique [23] for tree automata [25], which has been deeply used in the literature for translating alternating Büchi automata (on both words and trees) to nondeterministic ones. Then, the result

follows from the fact that the emptiness problem for NBT is solvable in polynomial time [32].

Related work: Our graded framework is fully based on the concepts of minimality and conservativeness. However, a version of graded CTL that does not use these concepts in its semantics has been recently studied in [13]. There, the authors consider overlapping paths (as we do) as well as disjoint paths. They solve the model checking problem and in particular, by opportunely extending the classical algorithm for CTL [10], they show that, in case of overlapping paths, the model checking problem is PTIME-COMplete (thus not harder than CTL), while for disjoint paths, it is in PSPACE and both NPTIME-HARD and CONPTIME-HARD. We finally remark that, differently from [13], we study the satisfiability problem and our automata-theoretic approach is one of the major contribution of this paper.

The paper is self contained. However, detailed proofs can be found on the accompanying Technical Report [5].

II. PRELIMINARIES

Given a *set* X of *objects* (numbers, words, etc.), we denote by $|X|$ the number of its elements, called the *size* of X , and by 2^X the *powerset* of X . In addition, by X^n we denote the set of all n -tuples of elements from X , by $X^* = \bigcup_{n=0}^{\omega} X^n$ the set of *finite words* on the *alphabet* X , and by X^+ the set $X^* \setminus \{\varepsilon\}$, where as usual, ω is the *numerable infinity* and ε is the *empty word*. By $|x|$ we denote the *length* of a word $x \in X^*$ and by $\{x_i\}_i^n$ the *ordered sequence* $(x_1, \dots, x_n) \in X^+$ of objects varying on the index i . As special sets, we consider \mathbb{N} and $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$ as, respectively, the sets of *natural numbers* and *positive natural numbers*. Furthermore, by $\mathbb{N}_{(n)}$ and $\mathbb{N}_{(n)_+}$ we denote the subsets $\{k \in \mathbb{N} \mid k \leq n\}$ of \mathbb{N} and $\{k \in \mathbb{N}_+ \mid k \leq n\}$ of \mathbb{N}_+ , where $n \in \mathbb{N} \cup \{\omega\}$.

A *structure* \mathcal{S} is an ordered tuple $\langle X, R \rangle$, where (i) $X = \text{dom}(\mathcal{S})$ is a non-empty and countable set of objects, called the *domain* of \mathcal{S} , and (ii) $R \subseteq X \times X$ is a *binary relation* between objects. We denote the size $|\mathcal{S}|$ of \mathcal{S} as the number $|X|$ of objects of its domain. An infinite structure is a structure of infinite size. When the relation R is clear from the context, to refer to a structure we only use its domain. A *tree* is a structure $\langle X, R \rangle$ in which the domain X , in the following also referred to as the set of *nodes*, is a subset of \mathbb{N}^* such that (i) if $x \cdot a \in X$, with $x \in \mathbb{N}^*$ and $a \in \mathbb{N}$, then also $x \in X$ and (ii) $(x, x') \in R$ iff $x' = x \cdot a$, for some $a \in \mathbb{N}$. The empty word ε is the *root* of the tree. A tree is *full* iff $x \cdot a \in X$, with $a \in \mathbb{N}$, implies $x \cdot b \in X$, for all $b \in \mathbb{N}_{(a)}$. A *path* is a tree $\langle X, R \rangle$ in which for all nodes $x \in X$ there is at most one $a \in \mathbb{N}$ such that $x \cdot a \in X$, i.e., the transitive closure of the relation R is a linear (total) order on X . A Σ -labeled structure $\mathcal{S} = \langle \Sigma, X, R, L \rangle$ is a tuple in which (i) Σ is a finite set of *labels*, (ii) $\langle X, R \rangle$ is a structure, and (iii) $L : X \mapsto \Sigma$ is a *labeling function* that colors each object with a label. When both Σ and R are clear from the context, we indicate a labeled structure $\langle \Sigma,$

$X, R, L \rangle$ with the shorter tuple $\langle X, L \rangle$.

Let $\mathcal{S} = \langle X, R \rangle$ and $\mathcal{S}' = \langle X', R' \rangle$ be two structures. We have that \mathcal{S}' is a *substructure* of \mathcal{S} , in symbols $\mathcal{S}' \preceq \mathcal{S}$, iff (i) $X' \subseteq X$ and (ii) $R' = R \cap (X' \times X')$ hold. Also, \mathcal{S} and \mathcal{S}' are *comparable* iff (i) $\mathcal{S} \preceq \mathcal{S}'$ or (ii) $\mathcal{S}' \preceq \mathcal{S}$ holds, otherwise they are *incomparable*. For a set of structures \mathfrak{S} , the set of *minimal substructures* (antichain) $\text{mins}(\mathfrak{S})$ is the set containing all and only the structures $\mathcal{S} \in \mathfrak{S}$ such that for all $\mathcal{S}' \in \mathfrak{S}$, it holds that (i) $\mathcal{S} \preceq \mathcal{S}'$, or (ii) \mathcal{S}' is incomparable with \mathcal{S} . Note that all structures in $\text{mins}(\mathfrak{S})$ are pairwise incomparable. A structure \mathcal{S} is *minimal* w.r.t. a set \mathfrak{S} (or simply minimal, when the context clarifies the set \mathfrak{S}) iff $\mathcal{S} \in \text{mins}(\mathfrak{S})$. A set of structures \mathfrak{S} is *minimal* iff $\mathfrak{S} = \text{mins}(\mathfrak{S})$.

A *Kripke structure* $\mathcal{K} = \langle \text{AP}, W, R, L \rangle$ is a 2^{AP} -labeled structure, where AP is a set of *atomic propositions*, $W = \text{dom}(\mathcal{K})$ is a set of *worlds* (the domain of the structure), R is a relation on W , and $L : W \mapsto 2^{\text{AP}}$ is the labeling function that maps each world to a set of atomic propositions true in that world. For a world $w \in W$, we define the *unwinding* of the structure \mathcal{K} starting from w as the full and possibly infinite 2^{AP} -labeled (Kripke) tree $U_w^{\mathcal{K}} = \langle \text{AP}, W', R', L' \rangle$ such that there is a function $\mathbf{uf} : W' \mapsto W$, called *unwinding function*, satisfying the following properties: (i) $\mathbf{uf}(\varepsilon) = w$ and, for all $w', v' \in W'$ and $u \in W$, it holds that (ii) $L'(w') = L(\mathbf{uf}(w'))$, (iii) if $(w', v') \in R'$, then $(\mathbf{uf}(w'), \mathbf{uf}(v')) \in R$, and, (iv) if $(\mathbf{uf}(v'), u) \in R$, then there is one and only one $u' \in W'$ such that $\mathbf{uf}(u') = u$ and $(v', u') \in R'$. Note that the unwinding function, and so the unwinding structure, is unique up to *isomorphisms*. Given a Kripke structure \mathcal{K} and $w \in W = \text{dom}(\mathcal{K})$, we define $\text{pth}(\mathcal{K}, w)$ as the set of paths of \mathcal{K} starting from w . Formally, a path π is in $\text{pth}(\mathcal{K}, w)$ iff $\pi \preceq U_w^{\mathcal{K}}$. In addition, we set $\text{pth}(\mathcal{K}) = \bigcup_{w \in W} \text{pth}(\mathcal{K}, w)$. With $\pi(\cdot)$ we denote the function $\pi : \mathbb{N}_{(|\pi|-1)} \mapsto W$ that maps each number $k \in \mathbb{N}_{(|\pi|-1)}$ to the world $\pi(k) = \mathbf{uf}(w')$ of \mathcal{K} , which corresponds to the $(k+1)$ -st position on the path π , where \mathbf{uf} is the unwinding function relative to $U_w^{\mathcal{K}}$, $w' \in \text{dom}(\pi)$, and $|w'| = k$. Note that $\pi(0) = \mathbf{uf}(\varepsilon) = w$.

Finally, let $n \in \mathbb{N}_+$, we define the following two sets: the set $\text{P}(n)$ of all *solutions* $\{p_i\}_i^n$ to the *linear Diophantine equation* $1*p_1 + 2*p_2 + \dots + n*p_n = n$ and the set $\text{CP}(n)$ of the *cumulative solutions* $\{q_i\}_i^n$ obtained by summing increasing sets of elements from $\{p_i\}_i^n$. Formally, $\text{P}(n) = \{\{p_i\}_i^n \in \mathbb{N}^n \mid \sum_{i=1}^n i * p_i = n\}$ and $\text{CP}(n) = \{\{q_i\}_i^n \in \mathbb{N}^n \mid \exists \{p_i\}_i^n \in \text{P}(n) \forall i \in \mathbb{N}_{(n)_+} : q_i = \sum_{j=i}^n p_j\}$. Note that $|\text{CP}(n)| = |\text{P}(n)|$ and, since for each solution $\{p_i\}_i^n$ of the above Diophantine equation there is exactly one partition of n , we have that $|\text{CP}(n)| = p(n)$, where $p(n)$ is the number of partitions of n . By [1], we have that, for a constant α , $p(n) = \Theta(\frac{1}{n} 2^{\alpha\sqrt{n}})$. So it follows that $|\text{CP}(n)| = \Theta(\frac{1}{n} 2^{\alpha\sqrt{n}})$.

III. THE GRADED CTL TEMPORAL LOGIC

In this section, we introduce an extension of the classical branching-time temporal logics CTL with graded path quantifiers. We show that this extension allows to gain expressiveness without paying any extra cost on deciding its satisfiability. For technical convenience, we introduce this logic through the state and path framework of CTL*.

The *graded computation tree logic* (GCTL*) extends CTL* by using two special path quantifiers, the universal $A^{<g}$ and the existential $E^{\geq g}$, where g denotes the corresponding *degree*. As in CTL*, these quantifiers can prefix a linear-time formula composed of an arbitrary combination and nesting of the temporal operators X (“effective next”), \bar{X} (“hypothetical next”), U (“until”), and R (“release”). The quantifiers $A^{<g}$ and $E^{\geq g}$ can be respectively read as “all but g minimal paths” and “there exist at least g minimal paths”. The formal syntax of GCTL* follows.

Definition III-A. (Syntax) GCTL* state (φ) and path (ψ) formulas are built inductively from AP using the following context-free grammar, where $p \in \text{AP}$ and $g \in \mathbb{N}$:

- 1) $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid A^{<g}\psi \mid E^{\geq g}\psi$,
- 2) $\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid X\psi \mid \bar{X}\psi \mid \psi U \psi \mid \psi R \psi$.

The class of GCTL* formulas is the set of state formulas generated by the above grammar. In addition, the simpler class of GCTL formulas is obtained by forcing each temporal operator, occurring in a formula, to be coupled with a path quantifier, as in the classical definition of CTL.

For a state formula φ , we define the *degree* $\text{deg}(\varphi)$ of φ as the maximum natural number g occurring among the degrees of all its path quantifiers. We assume that all such degrees are coded in unary. Accordingly, the *length* of a formula φ , denoted by $|\varphi|$, is defined inductively on the structure of φ in a classical way, by considering $|A^{<g}\psi|$ and $|E^{\geq g}\psi|$ to be equal to $g + 1 + |\psi|$. Clearly, $\text{deg}(\varphi) = \mathcal{O}(|\varphi|)$.

We now define the semantics of GCTL* w.r.t. a Kripke structure \mathcal{K} . For a world $w \in \text{dom}(\mathcal{K})$, we write $\mathcal{K}, w \models \varphi$ to indicate that a state formula φ holds at w , and, for a path $\pi \in \text{pth}(\mathcal{K})$, we write $\mathcal{K}, \pi, k \models \psi$ to indicate that a path formula ψ holds on π at position $k \in \mathbb{N}_{(|\pi|-1)}$. Note that, the relation $\mathcal{K}, \pi, k \models \psi$ does not hold for any point $k \in \mathbb{N}$, with $k \geq |\pi|$. For a better readability, in the semantics definition of GCTL* we use the special set $\mathfrak{P}_A(\mathcal{K}, w, \psi)$ and its dual $\mathfrak{P}_E(\mathcal{K}, w, \psi)$, with the following meaning: $\mathfrak{P}_A(\mathcal{K}, w, \psi)$ contains every path π starting in w such that all its extensions π' (including π) satisfy the path formula ψ . The semantics of GCTL* state formulas of the form $A^{<g}\psi$ and $E^{\geq g}\psi$ follows. The semantics of the remaining GCTL* state formulas and that of GCTL* path formulas is defined as usual in CTL* and reported in Appendix A.

Definition III-B. (Semantics of $A^{<g}$ and $E^{\geq g}$) Given a Kripke structure $\mathcal{K} = \langle \text{AP}, W, R, L \rangle$, a world $w \in W$, a path formula ψ , and a natural number g , it holds that:

- 1) $\mathcal{K}, w \models A^{<g}\psi$ iff $|\text{mins}(\text{pth}(\mathcal{K}, w) \setminus \mathfrak{P}_E(\mathcal{K}, w, \psi))| < g$;
- 2) $\mathcal{K}, w \models E^{\geq g}\psi$ iff $|\text{mins}(\mathfrak{P}_A(\mathcal{K}, w, \psi))| \geq g$;

where $\mathfrak{P}_A(\mathcal{K}, w, \psi) = \{\pi \in \text{pth}(\mathcal{K}, w) \mid \forall \pi' \in \text{pth}(\mathcal{K}, w) : \pi \preceq \pi' \text{ implies } \mathcal{K}, \pi', 0 \models \psi\}$ and $\mathfrak{P}_E(\mathcal{K}, w, \psi) = \{\pi \in \text{pth}(\mathcal{K}, w) \mid \exists \pi' \in \text{pth}(\mathcal{K}, w) : \pi \preceq \pi' \text{ and } \mathcal{K}, \pi', 0 \models \psi\}$.

Note that GCTL* (resp., GCTL) formulas with degrees 1 are CTL* (resp., CTL) formulas. Moreover, the above definition of $\mathfrak{P}_A(\mathcal{K}, w, \psi)$ and $\mathfrak{P}_E(\mathcal{K}, w, \psi)$ formally states that they are dual of each other, i.e., $\mathfrak{P}_A(\mathcal{K}, w, \psi) = \text{pth}(\mathcal{K}, w) \setminus \mathfrak{P}_E(\mathcal{K}, w, \neg\psi)$. Let \mathcal{K} be a Kripke structure and φ be a GCTL* formula. Then, \mathcal{K} is a *model* for φ , denoted by $\mathcal{K} \models \varphi$, iff there is $w \in \text{dom}(\mathcal{K})$ such that $\mathcal{K}, w \models \varphi$. In this case, we also say that \mathcal{K} is a model for φ on w . A GCTL* formula φ is said *satisfiable* iff there exists a model for it. Moreover φ is *invariant* on the two Kripke structures \mathcal{K} and \mathcal{K}' iff either $\mathcal{K} \models \varphi$ and $\mathcal{K}' \models \varphi$ or $\mathcal{K} \not\models \varphi$ and $\mathcal{K}' \not\models \varphi$.

We now give the formal definition of *conservativeness* and then, by means of two examples, we clarify the need of the concepts of minimality and conservativeness. A path π of \mathcal{K} is conservative w.r.t. a path formula ψ iff, for all paths π' extending π , i.e., with $\pi \preceq \pi'$, it holds that $\mathcal{K}, \pi', 0 \models \psi$. Note that this concept of conservativeness is automatically embedded in the definition of the set $\mathfrak{P}_A(\mathcal{K}, w, \psi)$, since we consider only paths $\pi \in \mathfrak{P}_A(\mathcal{K}, w, \psi)$ that, if extended, continue to satisfy the formula ψ . Now, for the minimality, consider a finite tree \mathcal{T} having just three nodes all labeled by p , the root and two of its successors. Also, consider the formula $\varphi = E^{\geq 2}Fp$. Because of the minimality, the two paths of length two that satisfy Fp collapse into the path containing just the root, hence $\mathcal{T} \not\models \varphi$. For the conservativeness, consider another tree \mathcal{T}' equal to \mathcal{T} , but with one of the two root successors not labeled with p . Also, consider the formula $\varphi' = E^{\geq 2}Gp$. At this point, by the conservativeness, we have that $\mathcal{T}' \not\models \varphi'$ even if there are two paths satisfying the formula Gp , since the path containing just the root is not conservative. Indeed, this path can be extended in a path that does not satisfy Gp .

For all state formulas φ_1 and φ_2 (resp., path formulas ψ_1 and ψ_2), we say that φ_1 is *equivalent* to φ_2 , formally $\varphi_1 \equiv \varphi_2$, (resp., ψ_1 is *equivalent* to ψ_2 , formally $\psi_1 \equiv \psi_2$) iff for all Kripke structures \mathcal{K} and worlds $w \in \text{dom}(\mathcal{K})$ it holds that $\mathcal{K}, w \models \varphi_1$ iff $\mathcal{K}, w \models \varphi_2$ (resp., $\text{mins}(\mathfrak{P}_A(\mathcal{K}, w, \psi_1)) = \text{mins}(\mathfrak{P}_A(\mathcal{K}, w, \psi_2))$).

In the rest of the paper, we only consider formulas in *existential normal form* or in *positive normal form*, i.e., formulas in which only existential quantifiers occur or negation is applied only to atomic propositions, respectively. In fact, it is to this aim that we have considered in the syntax of GCTL* both the connectives \wedge and \vee , the quantifiers $A^{<g}$ and $E^{\geq g}$, and the dual operators \bar{X} and R. Indeed, all formulas can be converted into existential or positive normal form by using De Morgan’s laws and the following equivalences, which directly follow from the semantics of

the logic. Let ψ , ψ_1 , and ψ_2 be path formulas and $g \in \mathbb{N}$, it holds that $\neg A^{<g}\psi \equiv E^{\geq g}\neg\psi$, $\neg X\psi \equiv \tilde{X}\neg\psi$, and $\neg(\psi_1 U \psi_2) \equiv \neg\psi_1 R \neg\psi_2$. To abbreviate formulas, we also use the boolean values \mathbf{t} (“true”) and \mathbf{f} (“false”) and the path quantifiers $E\psi \equiv E^{\geq 1}\psi$ (“there is a minimal path”) and $E^{>g}\psi \equiv E^{\geq g+1}\psi$ (“there are more than g minimal paths”).

The following lemma shows interesting equivalences among GCTL formulas that are useful to show important properties on the introduced logic. In particular, we show fixed point equivalences that extend to “graded” formulas the well known analogous ones for “ungraded” formulas.

Lemma III-C. *Let φ_1 and φ_2 be state formulas, $g > 1$, and $\text{ex}(\psi, g) = \bigvee_{\{h_i\}_i^g \in \mathcal{CP}(g)} \bigwedge_{i=1}^g E^{\geq h_i} X E^{\geq i}\psi$. Then, the following equivalences hold:*

- $E\varphi_1 U \varphi_2 \equiv \varphi_2 \vee \varphi_1 \wedge \text{ex}(\varphi_1 U \varphi_2, 1)$ and $E^{\geq g}\varphi_1 U \varphi_2 \equiv \neg\varphi_2 \wedge \varphi_1 \wedge \text{ex}(\varphi_1 U \varphi_2, g)$
- $E\varphi_1 R \varphi_2 \equiv \varphi_2 \wedge (\varphi_1 \vee E\tilde{X}\mathbf{f} \vee \text{ex}(\varphi_1 R \varphi_2, 1))$ and $E^{\geq g}\varphi_1 R \varphi_2 \equiv \varphi_2 \wedge \neg\varphi_1 \wedge EXE\neg(\varphi_1 R \varphi_2) \wedge \text{ex}(\varphi_1 R \varphi_2, g)$

The function $\text{ex}(\psi, g)$ used in the above lemma allows to partition g paths through h_1 successor worlds, for a given sequence $\{h_i\}_i^g \in \mathcal{CP}(g)$. Indeed, h_i is the number of successor worlds from which at least i paths satisfying ψ start. Therefore, h_1 is a sufficient bound on the number of successor worlds we have to consider to ensure the satisfiability of the formula. By a simple calculation, it also follows that $|\text{ex}(\psi, g)| = g * (|\psi| + \frac{g+11}{2}) * |\mathcal{CP}(g)| - 1 = \Theta((|\psi| + \frac{g}{2}) * 2^{\alpha\sqrt{g}})$, for a constant α . Note that, for $g = 1$, Lemma III-C gives the two classical fixed point expansions for CTL: $E(\varphi_1 U \varphi_2) \equiv \varphi_2 \vee \varphi_1 \wedge EX E(\varphi_1 U \varphi_2)$ and $E(\varphi_1 R \varphi_2) \equiv \varphi_2 \wedge (\varphi_1 \vee E\tilde{X}\mathbf{f} \vee EX E(\varphi_1 R \varphi_2))$.

In the next theorem, we show an exponential reduction of GCTL to the graded μ -calculus¹. Some adding details of the proof are reported in [5].

Theorem III-D. *Given a GCTL formula φ there exists an equivalent graded μ -calculus formula φ' whose size is at most exponential in the size of φ . Moreover, in some cases the blow-up is unavoidable.*

Proof: (Sketch) The equivalences shown in Lemma III-C suggest a reduction of GCTL to graded μ -calculus. Since this involves the use of the function $\text{ex}(\psi, g)$ of exponential size, we obtain an exponential translation. For the lower bound, consider the property “in a tree, there exist just g grandchildren of the root labeled with p , while all other nodes are not”. Such a property can be described by the GCTL formula $\varphi = (E^{\geq g}Fp) \wedge \neg p \wedge AX \neg p \wedge AXAXAXAG \neg p$. We claim that a graded μ -calculus formula φ' requires exponential size to express the same property. Let L be the set (of exponential size) of all the tree models of φ . For

each $1 \leq i \leq n$, let L_i be the set of trees having exactly i root successors labeled with p and all remaining nodes not labeled with p . For each L_i , it is possible to find in φ' a graded μ -calculus subformula φ'_i whose models are in L_i . Now, we recall that directly from the definition of GCTL models, trees in L are made in such a way that the n nodes labeled with p , present at the second level of it, are grouped as children of root children, w.r.t. a possible partitioning of n (i.e., a set in $\mathcal{P}(n)$). As an example, L contains the tree having two distinct root children x and y , where x has $n-1$ children labeled with p and y has just one node labeled with p . One can see that a tree in L can have several subtrees rooted at level 1 coming from different L_i s. Moreover, to get all trees in L , one has to consider all possible combination of trees from the sets L_i s. Since each of these correspond to a φ'_i , it means that every φ' requires an exponential disjunction of combinations of φ'_i . ■

By Theorem III-D and the fact that for graded μ -calculus the satisfiability problem is solvable in EXPTIME [18], we immediately get that the satisfiability problem for GCTL is decidable and solvable in 2EXPTIME, as reported in the following corollary. However, in the next section we improve this result by showing that the satisfiability problem for GCTL is solvable in EXPTIME, by exploiting an automata-theoretic approach that deeply makes use of the idea behind the function $\text{ex}(\psi, g)$ introduced in Lemma III-C.

Corollary III-E. *The satisfiability problem for GCTL is decidable and solvable in 2EXPTIME.*

We conclude this section by showing some interesting and simple properties about GCTL. First of all, by using a proof by induction we show that this logic is invariant under the unwinding of a model. Directly from this, we obtain that GCTL also enjoys the tree model property. Moreover, by the reduction to the graded μ -calculus (Theorem III-D), we directly obtain that GCTL has the finite model property². Straightforwardly, it also holds that GCTL is not invariant under bisimulation among models, since counting is not bisimilar-invariant at all. Directly from this, we obtain that GCTL is more expressive than CTL, since the latter is invariant under bisimulation. All these properties are reported in the following theorem.

Theorem III-F. *For GCTL it holds that (i) it is invariant under unwinding; (ii) it has the tree model property; (iii) it has the finite model property; (iv) is not invariant under bisimulation; and (v) it is more expressive than CTL.*

IV. PARTITIONING BÜCHI TREE AUTOMATA

Nondeterministic automata on infinite trees are an extension of nondeterministic automata on infinite words and

¹The μ -calculus is a well-known modal logic augmented with fixed point operators [17]. The graded μ -calculus extends the μ -calculus with graded state quantifiers [18], [6].

²To the best of our knowledge, the proof of the finite model property for the graded μ -calculus is not present in the literature. However, using the reduction to the classical μ -calculus explained in [18] it is possible to derive the property directly from that of the classical logic.

finite trees (see [30] for an introduction). *Alternating automata* [26] are a generalization of nondeterministic automata that embody the same concept of alternation as Turing machines [7]. Intuitively, while a nondeterministic automaton that visits a node of the input tree sends exactly one copy of itself to each of the successors of the node, an alternating automaton can send several copies of itself to the same successor. *Symmetric automata* [16], [33] are a variation of classical (asymmetric) alternating automata in which it is not necessary to specify the direction (i.e., the choice of the successors) of the tree on which a copy is sent. In fact, through three generalized directions (ϵ -moves, existential moves, and universal moves), it is possible to send a copy of the automaton, starting from a node of the input tree, to the same node, to some of its successors, or to all its successors. Hence, the automaton does not distinguish directions. As a generalization of symmetric automata, *graded alternating tree automata* (GATA, for short) have also been introduced [18]. In this framework, the automaton can send copies of itself to a given number n of successors, either in existential or universal way, without specifying which successors these exactly are. Moreover, a GATA can also send a copy of itself to the reading node by pursuing an ϵ -move.

Here, we consider *partitioning alternating tree automata* (PATA, for short) as a generalization of GATA in such a way that the automaton can send copies of itself to a given number n of paths, starting from the current node. As we show later, for each GCTL formula φ , it is possible to build a PATA that accepts all and only the tree models of φ . The key idea is to extend GATA's runs by also labeling their nodes with a natural number, with the aim of collecting "graded path information". We give an idea on how a PATA \mathcal{A} works w.r.t. the logic GCTL through an example.

First, note that \mathcal{A} uses as states all possible subformulas of the considered formula³, as it is done classically in the automata-theoretic approach to solve decidability problems in logic [19]. Now, suppose that the automaton is in the node x of an input tree T and in state $E^{\geq g}\psi$, where ψ is also a GCTL path formula. Then, in a state corresponding to ψ , the automaton sends $n \leq g$ copies of itself to n successors of x with degrees g_1, \dots, g_n that sum to g . One can note that this sequence of n degrees is a partition of the number g . The degree g_i associated to a successor x_i of x denotes that at least g_i minimal paths starting from x_i have to satisfy ψ and the automaton takes care of it through the transition function. In more details, we identify the set of n directions relative to successors of x w.r.t. the degrees $\{g_1, \dots, g_n\}$ by means of a decreasing chain $\{M_1, \dots, M_{n+1}\}$, such that for each i , it holds that $M_i \setminus M_{i+1}$ contains all directions of x that are associated with a degree i . Clearly, there could be

³More precisely, the automaton uses, as states, an extended definition of the Fischer-Ladner. See proof of Theorem 3 in [5] for a formal definition.

different possible chains satisfying such a property and each one induces a different run of \mathcal{A} on T . As a particular case, \mathcal{A} sends g copies of itself to g distinct successors of x on choosing $|M_1| = g$ and, for each $i > 1$, $M_i = \emptyset$.

The formal definition of a PATA along with the Büchi acceptance condition (PABT, for short) follows. In particular, we give a definition without any constraint on the use of its labeling degrees, which allows to introduce a more general class of automata, independently from the logic we consider here. Note that by the definition we give, the automaton on its own cannot enforce that multiple successors in which it is sent are all distinct. However, we can force this by means of the transition function. First, we introduce some extra notation. With $B^+(X)$ we denote the sets of *positive Boolean formulas* over X (i.e., Boolean formulas built from elements in X using \wedge and \vee) where we also allow the formulas t (true) and f (false). For a set $X' \subseteq X$ and a formula $\phi \in B^+(X)$, we say that X' satisfies ϕ , $X' \models \phi$, iff the assigning of true to elements in X' and false to elements in $X \setminus X'$ makes ϕ true. With D_b and D_b^ϵ we denote the sets $\{\diamond, \square\} \times \mathbb{N}_{(b)+}$ and $D_b \cup \{\epsilon\}$, respectively. Intuitively, these two sets represent the generalized directions that one can use, through the transition function, to describe the behavior of the automaton. For brevity, we often write $\langle g \rangle$ and $[g]$ instead of (\diamond, g) and (\square, g) , respectively.

Definition IV-A. (PABT) A partitioning alternating Büchi tree automaton is a tuple $\mathcal{A} = \langle Q, \Sigma, b, \delta, q_0, g_0, F \rangle$, where Q is a finite set of states, Σ is a finite input alphabet, $b \in \mathbb{N}$ is a counting branching bound, $\delta : Q \times \mathbb{N}_{(b)+} \times \Sigma \mapsto B^+(D_b^\epsilon \times Q)$ is a transition function, $q_0 \in Q$ is an initial state, $g_0 \in \mathbb{N}$ is an initial branching degree, and $F \subseteq Q \times \mathbb{N}_{(b)}$ is a Büchi acceptance condition, which is defined later.

The behavior of a PABT is described by means of a run. As for classical alternating automata, given a PABT $\mathcal{A} = \langle Q, \Sigma, b, \delta, q_0, g_0, F \rangle$ and a Σ -labeled tree $\langle T, \text{inp} \rangle$ in input, a run $\langle T_r, \text{run} \rangle$ of \mathcal{A} on $\langle T, \text{inp} \rangle$ is induced by the sets of pairs $S \subseteq D_b^\epsilon \times Q$ satisfying its transition function δ . Here, we first give an intuition of such a run through an example. Suppose that \mathcal{A} , while reading a node x of T labeled with σ , is in a state q with degree g at the node y of the run, and $\delta(q, g, \sigma) = (\epsilon, q_1) \wedge (\langle 3 \rangle, q_2) \vee ([2], q_3)$. Also, suppose that x has three successors $\{x \cdot 0, x \cdot 1, x \cdot 2\}$. Consider now $S = \{(\epsilon, q_1), (\langle 3 \rangle, q_2)\}$ satisfying $\delta(q, g, \sigma)$. Accordingly, \mathcal{A} can send a copy of itself to node x in the state q_1 (by performing an ϵ -move) and three copies of itself in the state q_2 to three paths through either one, two, or all successors of x . Now, suppose that we want to send two copies of \mathcal{A} through one successor and one through another. This can be characterized by taking $M_1 = \{0, 1\}$, $M_2 = \{1\}$, and $M_3 = M_4 = \emptyset$. Consequently, the run has three successors $\{y \cdot 0, y \cdot 1, y \cdot 2\}$, one labeled with $(x, q_1, 0)$ (for the ϵ -move), another labeled with $(x \cdot 0, q_2, 1)$, and the last one labeled with $(x \cdot 1, q_2, 2)$.

We now give the formal definition of a run. To this aim, we first formally define the sets $\{M_i\}_i^{g+1}$ introduced above, through a function spart , useful to define the required splitting among paths. Then, we introduce a function exec that allows to construct all possible execution steps.

Definition IV-B. (Splitting partition function) A splitting partition function $\text{spart} : (D, d) \in 2^{\mathbb{N}} \times D_b \mapsto \text{spart}(D, d) \in 2^{(2^{\mathbb{N}})^+}$ maps a set D and a direction d into a set of decreasing chains $\{M_i\}_i$ of subset of D ($M_i \subseteq D$ and $M_i \supseteq M_{i+1}$) such that:

- 1) if $d = \langle g \rangle$, then for all $\{M_i\}_i^{g+1} \in \text{spart}(D, d) \subseteq (2^D)^{g+1}$, it holds that $M_{g+1} = \emptyset$ and there is $\{h_i\}_i^g \in \mathcal{CP}(g)$ such that $|M_j| = h_j$, for all $j \in \mathbb{N}_{(g)+}$;
- 2) if $d = [g]$, then for all $\{M_i\}_i^{g+1} \in \text{spart}(D, d) \subseteq (2^D)^{g+1}$, it holds that $M_1 = D$ and for all $\{h_i\}_i^g \in \mathcal{CP}(g)$ there is $j \in \mathbb{N}_{(g)+}$ such that $|M_{j+1}| < h_j$.

Differently from GATA, one can see that in general the sets $\text{spart}(D, \langle g \rangle)$ and $\text{spart}(D, [g])$ are not the dual of each other. This is due to the fact that in PATA, for a considered node x , we may want to check properties along paths starting in x , instead of just looking at the successors of x , as it is done in GATA. This induces, in the $d = \langle g \rangle$ case, to take care of just g paths (on which we check that a certain property holds), while in the $d = [g]$ case we have to take care of all paths (i.e., that in less than g paths the property may or may not hold, while in all the remaining ones it must hold).

We now give the formal definition of the function exec . Let \mathbb{N}_ε denote the set $\mathbb{N} \cup \{\varepsilon\}$.

Definition IV-C. (Execution function) An execution function $\text{exec} : (S, D) \in 2^{D_b^\varepsilon \times Q} \times 2^{\mathbb{N}_\varepsilon} \mapsto \text{exec}(S, D) \in 2^{2^{\mathbb{N}_\varepsilon \times Q \times \mathbb{N}_{(b)}}}$ maps the two sets S and D into the set of all possible subsets of $\mathbb{N}_\varepsilon \times Q \times \mathbb{N}_{(b)}$, called configurations of the execution, such that, for all sets $E \in 2^{\mathbb{N}_\varepsilon \times Q \times \mathbb{N}_{(b)}}$ we have $E \in \text{exec}(S, D)$ iff for all pairs $(d, q) \in S$ it holds that:

- 1) if $d = \varepsilon$ then $(\varepsilon, q, 0) \in E$;
- 2) if either $d = \langle g \rangle$ or $d = [g]$ then there is $\{M_i\}_i^{g+1} \in \text{spart}(D, d)$ such that for all $i \in \mathbb{N}_{(g)+}$ and direction $x \in M_i \setminus M_{i+1}$, it holds that $(x, q, i) \in E$.

The above function exec allows us to give the following definition of PABT's run in a very concise and elegant way. First, we introduce some extra notations. Let $X' \subseteq X^*$ be a set of words on X and $x \in X^*$. Then, we denote by $\text{succ}_{X'}(x)$ the set of *successor words* of x in X' , i.e., $\text{succ}_{X'}(x) = \{x \cdot a \in X' \mid a \in \mathbb{N}\}$ and by $\text{dir}_{X'}(x)$ the set of *direction* of x in X' , i.e., $\text{dir}_{X'}(x) = \{a \in \mathbb{N} \mid x \cdot a \in X'\}$. Now, let $f : X' \mapsto X''$. We use $\text{inf}(f)$ to refer to the set $\{x \in X'' \mid |f^{-1}(x)| = \omega\}$, i.e., the set of elements of X'' that f uses infinitely often as labels for elements in X' , and $f|_{X''}$ to indicate the restriction of f to X'' , i.e., $f|_{X''} : X'' \mapsto X''$, where $X'' \subseteq X'$. In the following we

also write $S \models \delta(q, g, \sigma)$ to denote that S is a set of tuples $(d, q) \in D_b^\varepsilon \times Q$ that satisfies $\delta(q, g, \sigma)$.

Definition IV-D. (Run of a PABT) A run of a PABT $\mathcal{A} = \langle Q, \Sigma, b, \delta, q_0, g_0, F \rangle$ on a Σ -labeled tree $\langle T, \text{inp} \rangle$ is a $(T \times Q \times \mathbb{N}_{(b)})$ -labeled full tree $\langle T_r, \text{run} \rangle$ such that:

- 1) $\text{run}(\varepsilon) = (\varepsilon, q_0, g_0)$;
- 2) for all $y \in T_r$ with $\text{run}(y) = (x, q, g)$, there exist a set $S \subseteq D_b^\varepsilon \times Q$, where $S \models \delta(q, g, \text{inp}(x))$, and a set $E \in \text{exec}(S, \text{dir}_T(x))$ such that for all configurations $(d, q', g') \in E$ there is a node $y' \in \text{succ}_{T_r}(y)$ such that $\text{run}(y') = (x \cdot d, q', g')$.

The run $\langle T_r, \text{run} \rangle$ is accepting iff all its infinite paths satisfy the acceptance condition, i.e., for all paths $\pi \preceq T_r$, with $|\pi| = \omega$, it holds that $\text{inf}(\text{run}|_\pi) \cap T \times F \neq \emptyset$. A tree $\langle T, \text{inp} \rangle$ is accepted by \mathcal{A} iff there is an accepting run of \mathcal{A} on it. By $\mathcal{L}(\mathcal{A})$ we denote the language accepted by the automaton \mathcal{A} , i.e., the set of all input trees that \mathcal{A} accepts. \mathcal{A} is said to be empty if $\mathcal{L}(\mathcal{A}) = \emptyset$. The emptiness problem for \mathcal{A} is to decide whether $\mathcal{L}(\mathcal{A}) = \emptyset$.

By extending a construction given in [19], we obtain the following result.

Theorem IV-E. Given a GCTL formula φ with degree b , we can construct in time $\mathcal{O}(|\varphi|)$ a PABT \mathcal{A}_φ , with $\mathcal{O}(|\varphi|)$ states and counting branching bound b , such that $\mathcal{L}(\mathcal{A}_\varphi)$ is exactly the set of all tree models of φ .

Proof: (Sketch) The automaton \mathcal{A}_φ is defined as the tuple $\langle \text{ecl}(\varphi), 2^{\text{AP}}, \text{deg}(\varphi), \delta, \varphi, 0, F \rangle$, where $\text{ecl}(\varphi)$ is the Fisher-Ladner-closure extended in order to deal with graded path modalities. The acceptance condition F is the set of all pairs $(\langle \varphi_1 R \varphi_2 \rangle, 1)$ and $([\varphi_1 R \varphi_2], 1)$ of $\text{ecl}(\varphi) \times \mathbb{N}_{(b)}$. The transition function extends that introduced in [19] for CTL, along with the extra graded path modalities. Its formal definition is reported in [5]. Here, we only give an intuition of it through a couple of examples. Finally, a formal proof of the correctness of the whole construction, also reported in [5], follows naturally, by extending that (by induction on the structure of the formula) used for CTL.

First, recall that δ is a function from $\text{ecl}(\varphi) \times \mathbb{N}_{(b)} \times 2^{\text{AP}}$ into $\text{B}^+(D_b^\varepsilon \times \text{ecl}(\varphi))$. Consider now the state formula $\varphi = E^{\geq g} X \varphi'$. This formula is true on a tree model rooted at a node x having at least g distinct successors of x satisfying φ' . This is ensured through the δ in two successive steps. First, starting from the state $E^{\geq g} X \varphi'$, the δ gives the formula $(\langle g \rangle, \langle \varphi' \rangle)$, which intends to send to g successors (not necessarily distinct) the check of the satisfaction of φ' . Then, from state $\langle \varphi' \rangle$ we have to ensure that each of such successor nodes, say y , contributes to the satisfiability of exactly one φ' (intuitively one degree of φ). Therefore, on reading y , if the degree associated with the state $\langle \varphi' \rangle$ is greater than 1, the δ returns false, otherwise, with an ε -move, we move to state φ' . Accordingly, in the δ we use as counting branching positive numbers to indicate formulas'

degrees which have to be accomplished along paths and use as a convention 0 if we have none to accomplish. In particular, ϵ -moves always give 0 as counting branching.

As another example, consider the state formula $\varphi = E^{\geq g}(\varphi_1 \cup \varphi_2)$. This formula is true on a tree model rooted at a node x having at least g distinct minimal paths satisfying $\varphi_1 \cup \varphi_2$. As in CTL, the path formula $\varphi_1 \cup \varphi_2$ is true on a path if φ_2 is immediately true, or φ_1 is immediately true and then the formula $\varphi_1 \cup \varphi_2$ is satisfied on the successor node. Moreover, the quantifier $E^{\geq g}$ requires that there are at least g of such (minimal) paths. Therefore, if $g = 1$ the δ proceeds as in CTL. Conversely, if $g > 1$ we have to force φ_2 to not be immediately true (otherwise, we have less than g minimal paths satisfying the formula). Thus, we use the δ to ensure that φ_1 is immediately true and that g successive paths (but not necessarily all distinct) satisfy $\varphi_1 \cup \varphi_2$. Iteratively, the δ keeps using the above idea up to all states corresponding to the formula $\varphi_1 \cup \varphi_2$ are sent to next nodes with counting branching 1. This ensures that the considered tree model has at least g minimal paths satisfying the formula $\varphi_1 \cup \varphi_2$. Note that if less than g of such paths exist in the tree model, then the automaton keeps regenerating infinitely often the state corresponding to the until formula. Such a tree is then not accepting as this state is not in F . It is worth noting that, the above iteration upon the until states inherits the fixed point idea of the function $\text{ex}(\psi, g)$ introduced in Lemma III-C. In particular, we formally embed it into the δ through the formula $((1), \langle \varphi_1 \cup \varphi_2 \rangle)$ (see the formal definition of δ in [5] for details). This is a key step in our construction, since it allows to treat the exponential blow-up induced by this function by only using a constant rule into the δ . ■

In the remaining part of this section, we illustrate how the emptiness problem for PABT can be solved in EXPTIME. To gain this result, we use a technical extension of the Miyano and Hayashi technique [23] for tree automata [25], which has been deeply used in the literature for translating asymmetric Büchi automata to nondeterministic ones in exponential time. Here, we use this technique to translate with the same blow-up a PABT into nondeterministic Büchi tree automata (NBT, for short). Roughly speaking, this means that we manage to combine the exponential blow-up induced by the alternation and that induced by the permutations of all possible splitting degrees in only one exponential blow-up. This technique is illustrated in the next theorem.

Theorem IV-F. *Let \mathcal{A} be a PABT with n states and counting branching bound b . Then, there exists a NBT \mathcal{A}' with $2^{2n \cdot (b+1)}$ states and $n \cdot b(b+1)/2$ directions such that \mathcal{A} is not empty iff \mathcal{A}' is not.*

Proof: (Sketch) The NBT \mathcal{A}' guesses a subset construction applied to a run of the PABT. At a given node x of a run of \mathcal{A}' , it keeps in its memory the set of states in which the various copies of \mathcal{A} visit the node x in the guessed run. In order to make sure that every infinite path

visits accepting states in F infinitely often, \mathcal{A}' keeps track of states that “owe” a visit to F . The fact that PABT are symmetric, however, requires further non-trivial work, indeed, differently from the classical approach, we have to convert the symmetric automaton \mathcal{A} into a nondeterministic one. This is because, while for symmetric automata there is bijective correspondence between direction of both the input and output automaton, in our case we have to build this correspondence by looking at the δ of the input automaton. The extra problems are: (i) \mathcal{A} can perform ϵ -moves and (ii) \mathcal{A} does not have an upper bound on the number of directions it uses. The first problem is solved by using in \mathcal{A}' an opposite direction that collects all states of \mathcal{A} sent through ϵ -moves during a given execution. We face the second problem by using the following property of PABT’s: if \mathcal{A} accepts a tree T , it must accept also a tree T' with branching degree at most equal to $d' = n \cdot b(b+1)/2$. This holds since, in each state q and degree g at a node x of the input tree, a set S that satisfies the $\delta(q, g, \text{inp}(x))$ can contain at most $|Q \times \mathbb{N}_{(b+)}|$ pairs of the kind $(\langle g' \rangle, q')$. So, we can split each of such a pair in at most g' nodes of degree 1 and then, for each state q' , we can have at most $b(b+1)/2$ distinct successors of x . Therefore, it is possible to construct a relative run of \mathcal{A}' by restricting our attention only to trees with degree at most d' . The full construction of \mathcal{A}' is reported in [5]. We conclude this proof sketch by only giving some intuition for the transition relation of \mathcal{A}' .

Suppose $\mathcal{A} = \langle \{q_0, q_1\}, \{a\}, 2, \delta, q_0, 0, F \rangle$, where the δ contains $\delta(q_0, 0, a) = (\epsilon, q_0) \wedge (\langle 2 \rangle, q_1)$. Hence, the degree bound d' for \mathcal{A} is 6. Also, suppose that \mathcal{A}' is in the state (H, H') , with $H = \{(q_0, 0)\}$ and $H' = \emptyset$. Now, as in the classical case, the set $\{(\epsilon, q_0), (\langle 2 \rangle, q_1)\}$ satisfies the relation δ for all states in H , but in our construction we can not use this set directly to build δ' , since all these tuples contain an additional information (the degree) that we must use to split (H, H') in all possible successors. Accordingly to this fact indeed, we have the following two possibilities: either \mathcal{A}' sends a copy of itself to one child with degree 2 or to two children with degree 1. Moreover, in both cases \mathcal{A}' also sends a witness of the ϵ -move to direction d' . ■

Recall that for the NBT \mathcal{A}' with Q' as state set and branching degree d' the emptiness problem is solvable in PTIME [32] and, precisely, in $\mathcal{O}(|Q'|^{2d'})$ (we directly consider the one-letter automaton associated to \mathcal{A}'). Then, by Theorem IV-F, the following result follows.

Theorem IV-G. *The emptiness problem for a PABT \mathcal{A} with n states and counting branching bound b can be decided in time $2^{\mathcal{O}(n^2 \cdot b^3)}$.*

By Theorems IV-E and IV-G, and by $n = |\text{ecl}(\varphi)| = \mathcal{O}(|\varphi|)$ and $b = \text{deg}(\varphi) = \mathcal{O}(|\varphi|)$, we get that the satisfiability problem for GCTL is in EXPTIME and precisely solvable in time $2^{\mathcal{O}(|\varphi|^5)}$. Since GCTL subsumes CTL, the following holds.

Theorem IV-H. *The satisfiability problem for GCTL is EXPTIME-COMPLETE.*

V. DISCUSSION

Graded modalities refine classical existential and universal quantifiers by specifying the number of elements for which the existential requirement should hold/universal requirement may not hold. Earlier work studied the extension of the μ -calculus by graded modalities and shown that the complexity of the satisfiability problem stays EXPTIME in the graded setting. In this paper, we have introduced and investigated a (semantic) fragment of the graded μ -calculus, that is GCTL, which extends CTL with graded path quantifiers under a suitable concept of minimality and conservativeness over paths. In particular, we have shown an exponential translation from GCTL to graded μ -calculus, and we have proved that in some cases this blow-up is unavoidable, making GCTL even more appealing in practice.

One of the main features of this logic is the capability to express properties that are weaker than those definable with the universal quantifications $A\psi$ and stronger than those definable with the existential quantifications $E\psi$. In “*planning in nondeterministic domain*” [9], [8], for examples, the use of strong planning (i.e., all the goals have to be satisfied by all the computations) and weak planning (i.e., all the goals have to be satisfied by some computation) are two extreme ways to achieve a given purpose. With our logic, we are able to express “graded specification” that can be considered as a compromise between strong and weak planning, which are also forced to be minimal (succinct).

As interesting results about GCTL, we have shown that although this logic is more expressive than CTL, it retains an EXPTIME satisfiability procedure. This result has been achieved by exploiting an automata-theoretic approach via the introduction of a new automata model, that is PATA. As an immediate consequence, by using a classical product-automata construction [19] through PATA’s, one can also get that the model checking problem is as easy as CTL, i.e. it stays in PTIME. We postpone this to future work. However, we recall that model checking for GCTL without the concepts of minimality and conservativeness has been already solved in [13].

Other directions for future work regard the investigation of graded path modalities along with more complex logics, such as CTL^* , i.e., to investigate $GCTL^*$. We believe that it is not hard to extend to this logic the properties shown for GCTL in Theorem III-F (expressiveness, tree and finite model properties). On the contrary, to solve the satisfiability problem for $GCTL^*$ is less than immediate as the automata model we have considered in this paper for GCTL is not appropriate. Indeed, by using a theoretic-automata approach similar to that one used for GCTL, we can reduce the satisfiability problem for $GCTL^*$ to the emptiness problem of PATA, but with an acceptance condition stronger than Büchi,

such as the parity one [25]. Unfortunately, the technique we have shown to translate PABT into NBT is not appropriate for parity automata. However, by using a technique based on promises and strategies, as it was done in [18], we conjecture that PATA along with a parity condition can be translated in exponential-time into an alternating parity tree automaton. Then, by using the fact that for the latter the emptiness problem is solvable in exponential-time, we get that the satisfiability problem for $GCTL^*$ is solvable in $2EXPTIME$, thus not harder than that for CTL^* . By exploiting a similar idea of that used for graded μ -calculus, one could also show that $GCTL^*$ is equivalent to CTL^* augmented with graded world modalities (Counting- CTL^* [24]) and we conjecture that $GCTL^*$ is exponentially more succinct than Counting- CTL^* (for GCTL and Counting-CTL, this result holds by simply applying the same idea used for the translation from GCTL to the graded μ -calculus). This result is important as it was shown in [24] that Counting- CTL^* is equivalent to *monadic path logic*, which is MSO with set quantifications restricted to paths.

Acknowledgement. We wish to thank Martin Lange and Orna Kupferman for their helpful comments on a preliminary version of the paper.

REFERENCES

- [1] T.M. Apostol. *Introduction to Analytic Number Theory*. Springer-Verlag, 1976.
- [2] M. Arenas, P. Barceló, and L. Libkin. Combining Temporal Logics for Querying XML Documents. In *ICDT’07*, LNCS 4353, pages 359–373. Springer-Verlag, 2007.
- [3] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [4] P. Barceló and L. Libkin. Temporal Logics over Unranked Trees. In *LICS*, pages 31–40. IEEE Computer Society, 2005.
- [5] A. Bianco, F. Mogavero, and A. Murano. Graded Computation Tree Logic. Technical report. Updated version: <http://www.fabiomogavero.com/tworks/carticles/bmm09.pdf>.
- [6] P.A. Bonatti, C. Lutz, A. Murano, and M.Y. Vardi. The Complexity of Enriched μ -Calculi. *LMCS*, 4(3:11):1–27, 2008.
- [7] A.K. Chandra, D. Kozen, and L.J. Stockmeyer. Alternation. *JACM*, 28(1):114–133, 1981.
- [8] A. Cimatti, M. Pistore, M. Roveri, and P. Traverso. Weak, Strong, and Strong Cyclic Planning via Symbolic Model Checking. *AI*, 147(1-2):35–84, 2003.
- [9] A. Cimatti, M. Roveri, and P. Traverso. Strong Planning in Non-Deterministic Domains via Model Checking. In *AIPS’98*, pages 36–43, 1998.

- [10] E.M. Clarke and E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71. Springer-Verlag, 1981.
- [11] E.A. Emerson and J.Y. Halpern. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *JCSS*, 30(1):1–24, 1985.
- [12] E.A. Emerson and J.Y. Halpern. “Sometimes” and “Not Never” Revisited: On Branching Versus Linear Time. *JACM*, 33(1):151–178, 1986.
- [13] A. Ferrante, M. Napoli, and M. Parente. CTL Model-Checking with Graded Quantifiers. In *ATVA'08*, LNCS 5311, pages 18–32. Springer-Verlag, 2008.
- [14] K. Fine. In So Many Possible Worlds. *NDJFL*, 13:516–520, 1972.
- [15] Erich Grädel. On The Restraining Power of Guards. *JSL*, 64(4):1719–1742, 1999.
- [16] D. Janin and I. Walukiewicz. Automata for the Modal μ -Calculus and Related Results. In *MFCS'95*, LNCS 969, pages 552–562. Springer-Verlag, 1995.
- [17] D. Kozen. Results on the Propositional μ -Calculus. *TCS*, 27:333–354, 1983.
- [18] O. Kupferman, U. Sattler, and M.Y. Vardi. The Complexity of the Graded μ -Calculus. In *CADE'02*, LNCS 2392, pages 423–437. Springer-Verlag, 2002.
- [19] O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.
- [20] L. Lamport. “Sometime” is Sometimes “Not Never”: On the Temporal Logic of Programs. In *POPL'80*, pages 174–185, 1980.
- [21] L. Libkin and C. Sirangelo. Reasoning About XML with Temporal Logics and Automata. In *LPAR'08*, LNCS 5330, pages 97–112, 2008.
- [22] T.J. McCabe. A Complexity Measure. *TSE*, 2:308–320, 1976.
- [23] S. Miyano and T. Hayashi. Alternating Finite Automata on ω -Words. *TCS*, 32:321–330, 1984.
- [24] F. Moller and A.M. Rabinovich. Counting on CTL*: On the Expressive Power of Monadic Path Logic. *IC*, 184(1):147–159, 2003.
- [25] A. Mostowski. Regular Expressions for Infinite Trees and a Standard Form of Automata. In *CT'84*, LNCS 208, pages 157–168. Springer-Verlag, 1984.
- [26] D.E. Muller and P.E. Schupp. Alternating Automata on Infinite Trees. *TCS*, 54(2-3):267–276, 1987.
- [27] A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57, 1977.
- [28] A. Pnueli. The Temporal Semantics of Concurrent Programs. *TCS*, 13:45–60, 1981.
- [29] M. Schmidt-Schauß and G. Smolka. Attributive Concept Descriptions with Complements. *AI*, 48(1):1–26, 1991.
- [30] W. Thomas. Automata on Infinite Objects. In *Handbook of Theoretical Computer Science (vol. B)*, pages 133–191. MIT Press, 1990.
- [31] S. Tobies. PSpace Reasoning for Graded Modal Logics. *JLC*, 11(1):85–106, 2001.
- [32] M.Y. Vardi and P. Wolper. Automata-Theoretic Techniques for Modal Logics of Programs. *JCSS*, 32(2):183–221, 1986.
- [33] T. Wilke. CTL+ is Exponentially More Succinct than CTL. In *FSTTCS'99*, pages 110–121. Springer-Verlag, 1999.

APPENDIX

Given a Kripke structure $\mathcal{K} = \langle AP, W, R, L \rangle$ and $w \in W$, for all GCTL* state formulas φ , the relation $\mathcal{K}, w \models \varphi$, is inductively defined as follows.

- 1) $\mathcal{K}, w \models p$, with $p \in AP$, iff $p \in L(w)$.
- 2) For state formulas φ , φ_1 , and φ_2 , it holds:
 - (a) $\mathcal{K}, w \models \neg\varphi$ iff not $\mathcal{K}, w \models \varphi$, that is $\mathcal{K}, w \not\models \varphi$;
 - (b) $\mathcal{K}, w \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{K}, w \models \varphi_1$ and $\mathcal{K}, w \models \varphi_2$;
 - (c) $\mathcal{K}, w \models \varphi_1 \vee \varphi_2$ iff $\mathcal{K}, w \models \varphi_1$ or $\mathcal{K}, w \models \varphi_2$.
- 3) For a path formula ψ and a natural number g , it holds:
 - (a) $\mathcal{K}, w \models A^{<g}\psi$ iff $|\text{mins}(\text{pth}(\mathcal{K}, w) \setminus \mathfrak{P}_E(\mathcal{K}, w, \psi))| < g$;
 - (b) $\mathcal{K}, w \models E^{\geq g}\psi$ iff $|\text{mins}(\mathfrak{P}_A(\mathcal{K}, w, \psi))| \geq g$;
where $\mathfrak{P}_A(\mathcal{K}, w, \psi) = \{\pi \in \text{pth}(\mathcal{K}, w) \mid \forall \pi' \in \text{pth}(\mathcal{K}, w) : \pi \preceq \pi' \text{ implies } \mathcal{K}, \pi', 0 \models \psi\}$ and $\mathfrak{P}_E(\mathcal{K}, w, \psi) = \{\pi \in \text{pth}(\mathcal{K}, w) \mid \exists \pi' \in \text{pth}(\mathcal{K}, w) : \pi \preceq \pi' \text{ and } \mathcal{K}, \pi', 0 \models \psi\}$.

For all GCTL* path formulas ψ , paths $\pi \in \text{pth}(\mathcal{K})$, and natural numbers $k < |\pi|$, the relation $\mathcal{K}, \pi, k \models \psi$ is inductively defined as follows.

- 4) $\mathcal{K}, \pi, k \models \varphi$, with φ state formula, iff $\mathcal{K}, \pi(k) \models \varphi$.
- 5) For path formulas ψ , ψ_1 , and ψ_2 , it holds:
 - (a) $\mathcal{K}, \pi, k \models \neg\psi$ iff not $\mathcal{K}, \pi, k \models \psi$, that is $\mathcal{K}, \pi, k \not\models \psi$;
 - (b) $\mathcal{K}, \pi, k \models \psi_1 \wedge \psi_2$ iff $\mathcal{K}, \pi, k \models \psi_1$ and $\mathcal{K}, \pi, k \models \psi_2$;
 - (c) $\mathcal{K}, \pi, k \models \psi_1 \vee \psi_2$ iff $\mathcal{K}, \pi, k \models \psi_1$ or $\mathcal{K}, \pi, k \models \psi_2$.
 - (d) $\mathcal{K}, \pi, k \models X\psi$ iff $k < |\pi| - 1$ and $\mathcal{K}, \pi, (k + 1) \models \psi$;
 - (e) $\mathcal{K}, \pi, k \models \tilde{X}\psi$ iff $k = |\pi| - 1$ or $\mathcal{K}, \pi, (k + 1) \models \psi$;
 - (f) $\mathcal{K}, \pi, k \models \psi_1 U \psi_2$ iff there is an index i , with $k \leq i < |\pi|$, such that $\mathcal{K}, \pi, i \models \psi_2$ and, for all indexes j with $k \leq j < i$, it holds $\mathcal{K}, \pi, j \models \psi_1$;
 - (g) $\mathcal{K}, \pi, k \models \psi_1 R \psi_2$ iff for all indexes i , with $k \leq i < |\pi|$, it holds $\mathcal{K}, \pi, i \models \psi_2$ or there is an index j with $k \leq j < i$, such that $\mathcal{K}, \pi, j \models \psi_1$.