Graded Computation Tree Logic (GCTL)

Alessandro Bianco Fabio Mogavero Aniello Murano

Universitá degli Studi di Napoli "Federico II"
http://people.na.infn.it/~{alessandrobianco,mogavero,murano}

In Proceedings of
LICS 2009

September 2009

Systems correctness

Let **S** be a system and **P** a desired behavior (specification).

Two very important problems:

- Model Checking: Is **S** correct w.r.t. **P**?
- Satisfiability: Is **P** a correct specification?

To answer to these questions, formal methods are used.

- **S** can be modelled by a labeled transition graph $\mathcal{K}$ (Kripke structure).
- **P** can be expressed as a temporal logic formula φ.

Then,

- Model Checking: $\mathcal{K} \models \varphi$?
- Satisfiability: Is there a $\mathcal{K}$ such that $\mathcal{K} \models \varphi$?

Systems specifications

Temporal logic: description of the temporal ordering of events!

Two main families of temporal logics:

- Linear-Time Temporal Logics (**LTL**)
  - Each moment in time has a unique possible future.
  - Useful for hardware specification.
- Branching-Time Temporal Logics (**CTL**, **CTL***, and μ-**CALCULUS**)
  - Each moment in time may split into various possible future.
  - Useful for software specification.

The μ-CALCULUS subsumes many logics, in particular, LTL, CTL, and CTL*.

Several extension of μ-CALCULUS have been considered.

One among all: the GRADED μ-CALCULUS, i.e., the μ-CALCULUS extended with graded modalities [“there are at least *n* successors such that...”].

Computational complexity

	M.C.	Sat.
LTL	PSPACE-COMPLETE	PSPACE-COMPLETE
CTL	PTIME-COMPLETE	EXPTIME-COMPLETE
CTL*	PSPACE-COMPLETE	2EXPTIME-COMPLETE
μ-CALCULUS	UPTIME $\cap$ COUPTIME	EXPTIME-COMPLETE
GRADED μ-CALCULUS[a][b]	UPTIME $\cap$ COUPTIME	EXPTIME-COMPLETE

Table: Computational complexity of Model Checking and Satisfiability.

[a] O. Kupferman, U. Sattler, and M. Vardi. The Complexity of the GRADED μ-CALCULUS, CADE'02.

[b] P. Bonatti, C. Lutz, A. Murano, and M. Vardi. The Complexity of Enriched μ-Calculi, ICALP'06 / LMCS'08.

μ-CALCULUS: very expressive but too low-level (hard to understand).

LTL, CTL, and CTL*: less expressive but much more human-friendly.

Our motivation

A very challenging issue is to extend the expressiveness of classical temporal logics to model more complex specifications, in a way that

- there is no extra cost on determine its decision problems,
- the resulting formal language is easy to use and understand.

A natural question: how could logics that allow to reasoning about path be affected by considering graded modalities?

Our proposal

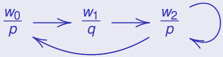
We investigate the extension of CTL with graded modalities (GCTL, for short).

Possible applications/connetions:

- XML query language;
- cyclomatic complexity;
- redundancy in a system.

There is a technical challenge involved with such an extension:

- the concept of grade have to relapse both on states and paths;
- it is easy to have structures with an infinite number of paths satisfying a given property (e.g., F q), so the concept of grade becomes unuseful.



$$w_0 \longrightarrow w_1 \longrightarrow w_2 \longrightarrow w_0/w_2 \longrightarrow w_1/w_0/w_2 \longrightarrow \cdots$$

We solve this problem using the concepts of minimality and conservativeness.

Outline

Syntax of GCTL* and GCTL

Definition

GCTL* *state* (φ) and *path* (ψ) *formulas* are built inductively as follows:

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

The simpler class of GCTL formulas is obtained by forcing each temporal operator, occurring in a formula, to be coupled with a path quantifier.

Since our semantics is defined on finite paths, the next-time operator X is no more the dual of itself, hence we have in the syntax both X and its dual $\tilde{\mathsf{X}}$.

For $g = 1$, we may write $\mathsf{E}\psi$ and $\mathsf{A}\psi$ instead of $\mathsf{E}^{\geq g}\psi$ and $\mathsf{A}^{<g}\psi$.

Informal meaning of $E^{\geq g}$ and $A^{<g}$

Definition

GCTL* *state* (φ) and *path* (ψ) *formulas* are built inductively as follows:

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

The simpler class of GCTL formulas is obtained by forcing each temporal operator, occurring in a formula, to be coupled with a path quantifier.

Informally, the graded quantifiers $E^{\geq g}\psi$ and $A^{<g}\psi$ can be read as

- $E^{\geq g}\psi$: there exist at least *g* paths that satisfy ψ,
- $A^{<g}\psi$: all but less than *g* paths satisfy ψ.

However, the domain on which the quantifiers range is not the class of all infinite paths, but that containing all finite, minimal, and conservative paths.

Kripke structures, paths, order, and minimality

Definition

A *Kripke structure* (KRIPKE, for short) is a tuple $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathrm{L} \rangle$ where:

1. AP: finite non-empty set of *atomic propositions*;
2. W: non-empty set of *worlds*;
3. $R \subseteq \mathrm{W} \times \mathrm{W}$: *transition* relation;
4. $\mathrm{L} : \mathrm{W} \mapsto 2^{\mathrm{AP}}$: *labeling* function.

A path π of a KRIPKE $\mathcal{K}$ is a finite sequence of states compatible with the transition relation R of $\mathcal{K}$.
A path π' is a *subpath* of π, formally $\pi' \preccurlyeq \pi$, iff the first is a prefix of the latter.
For a set of paths P, we say that π is *minimal* in P iff, for all $\pi' \in \mathrm{P}$, it holds that *(i)* $\pi \preccurlyeq \pi'$ or *(ii)* $\pi' \not\preccurlyeq \pi$.
By $\min(\mathrm{P})$ we denote the *antichain* (i.e., the set of minimal paths) of P w.r.t. $\preccurlyeq$.

Semantics of GCTL*

Definition

Given a KRIPKE $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathrm{L} \rangle$, a world $w \in \mathrm{W}$, and a GCTL* path formula ψ, it holds that:

1. $\mathcal{K}, w \models \mathsf{E}^{\geq g}\psi$ iff $|\min(\mathfrak{P}_A(\mathcal{K}, w, \psi))| \geq g$;
2. $\mathcal{K}, w \models \mathsf{A}^{<g}\psi$ iff $|\min(\mathrm{P}(\mathcal{K}, w) \setminus \mathfrak{P}_E(\mathcal{K}, w, \psi))| < g$;

where $\mathrm{P}(\mathcal{K}, w)$ is the set of finite paths of $\mathcal{K}$ starting in w and $\mathfrak{P}_A(\mathcal{K}, w, \psi)$ (resp., $\mathfrak{P}_E(\mathcal{K}, w, \psi)$) is the set of those paths that are (resp., non) *conservative* w.r.t. ψ (resp., $\neg\psi$).

$\pi \in \mathrm{P}(\mathcal{K}, w)$ is conservative w.r.t. ψ iff, for all $\pi' \in \mathrm{P}(\mathcal{K}, w)$, it holds that $\pi \preccurlyeq \pi'$ implies $\mathcal{K}, \pi, 0 \models \psi$, i.e., all paths extending π satisfy ψ.

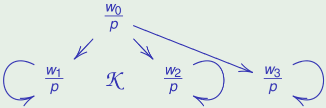
- $\mathfrak{P}_A(\mathcal{K}, w, \psi) = \mathrm{P}(\mathcal{K}, w) \setminus \mathfrak{P}_E(\mathcal{K}, w, \neg\psi)$.
- $\neg\mathsf{E}^{\geq g}\psi \equiv \mathsf{A}^{<g}\neg\psi$.

Minimality and conservativeness

Example (Minimality for F p)

$\mathfrak{P}_A(\mathcal{K}, w_0, \mathrm{F}\, p) = \mathrm{P}(\mathcal{K}, w_0)$,
$\min(\mathfrak{P}_A(\mathcal{K}, w_0, \mathrm{F}\, p)) = \{w_0\}$.
$\mathcal{K}, w_0 \models \mathrm{E}^{\geq 1}\mathrm{F}\, p$,
$\mathcal{K}, w_0 \not\models \mathrm{E}^{\geq 2}\mathrm{F}\, p$.

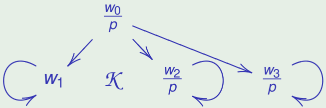
$\mathrm{P}(\mathcal{K}, w_0) = \{w_0 \cdot (w_1^* + w_2^* + w_3^*)\}$,



Example (Conservativeness for G p)

$\mathfrak{P}_A(\mathcal{K}, w_0, \mathrm{G}\, p) = \{w_0 \cdot (w_2^+ + w_3^+)\}$,
$\min(\mathfrak{P}_A(\mathcal{K}, w_0, \mathrm{G}\, p)) = \{w_0 \cdot (w_2 + w_3)\}$.
$\mathcal{K}, w_0 \models \mathrm{E}^{\geq 2}\mathrm{G}\, p$,
$\mathcal{K}, w_0 \not\models \mathrm{E}^{\geq 3}\mathrm{G}\, p$.

$\mathrm{P}(\mathcal{K}, w_0) = \{w_0 \cdot (w_1^* + w_2^* + w_3^*)\}$,



Counting nodes on trees

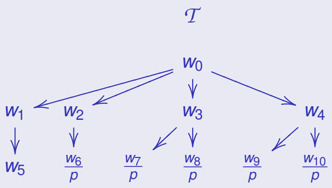
$$\begin{array}{ccccccccc} 5 & = & 1 & + & (2 & + & 2) \\ \underbrace{\{w_6, w_7, w_8, w_9, w_{10}\}}_{w_0} & = & \underbrace{\{w_6\}}_{w_2} & \underbrace{\cup}_{h_1 = 3} & (\underbrace{\{w_7, w_8\}}_{w_3} & \underbrace{\cup}_{h_2 = 2} & \underbrace{\{w_9, w_{10}\}}_{w_4}) \end{array}$$

$\mathcal{T}, w_0 \models \mathrm{E}^{\geq 5}\mathrm{F}\, p \rightsquigarrow \{w_6, w_7, w_8, w_9, w_{10}\}$.

$\mathcal{T}, w_0 \models \mathrm{E}^{\geq 3}\mathrm{X}\, \mathrm{E}^{\geq 1}\mathrm{F}\, p \rightsquigarrow \{w_2, w_3, w_4\}$,
$\mathcal{T}, w_0 \models \mathrm{E}^{\geq 2}\mathrm{X}\, \mathrm{E}^{\geq 2}\mathrm{F}\, p \rightsquigarrow \{w_3, w_4\}$.

$\mathcal{T}, w_0 \models \bigwedge_{i=1}^{5} \mathrm{E}^{\geq h_i}\mathrm{X}\, \mathrm{E}^{\geq i}\mathrm{F}\, p$.

$h_i = j$ means that there are j successors of w_0 from which i paths satisfying $\mathrm{F}\, p$ start ($h_3 = h_4 = h_5 = 0$).



One-step unfolding (I)

We want to prove a one-step unfolding property for $\mathsf{E}^{\geq g}\mathsf{X}\,\psi$.

Decompose g into all possible integer partitions:

$$\begin{array}{ccccccccccc} g & = & \underbrace{(1+\ldots+1)} & + & \underbrace{(2+\ldots+2)} & + & \ldots & + & \underbrace{g} \\ g & = & 1 * p_1 & + & 2 * p_2 & + & \ldots & + & g * p_g \end{array}$$

Sum all elements in the following way: $h_i = \sum_{j=i}^{g} p_j$.
Let $CP(g)$ be the set of all such finite sequences $\{h_i\}_i$.

Then, we have that $\mathsf{E}^{\geq g}\mathsf{X}\,\psi \equiv \bigvee_{\{h_i\}_i \in CP(g)} \bigwedge_{i=1}^{g} \mathsf{E}^{\geq h_i}\mathsf{X}\,\mathsf{E}^{\geq i}\psi$.

One-step unfolding (II)

The following properties hold, where φ, φ', and ψ are, respectively, two state and a path formula:

1. $E^{\geq g}(\varphi \wedge \psi) \equiv \varphi \wedge E^{\geq g}\psi$;
2. $E^{\geq g}(\varphi \vee \psi) \equiv \begin{cases} \varphi \vee E^{\geq g}\psi, & \text{if } g = 1; \\ \neg\varphi \wedge E^{\geq g}\psi, & \text{otherwise;} \end{cases}$
3. $\varphi \,U\, \varphi' \equiv \varphi' \vee (\varphi \wedge X(\varphi \,U\, \varphi'))$.

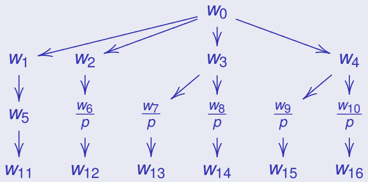
$$E^{\geq g}\varphi \,U\, \varphi' \equiv \begin{cases} \varphi' \vee (\varphi \wedge EX\,E(\varphi \,U\, \varphi')), & \text{if } g = 1; \\ \neg\varphi' \wedge \varphi \wedge \bigvee_{\{h_i\}_{i \in CP(g)}} \bigwedge_{i=1}^{g} E^{\geq h_i} X\, E^{\geq i}(\varphi \,U\, \varphi'), & \text{otherwise.} \end{cases}$$

Using this property, we are able to reduce GCTL to the GRADED μ-CALCULUS with an exponential blow-up (since $|CP(g)| = O(2^{\sqrt{g}})$).

Succinctness

Consider the property "in a tree, there exist at least g grandchildren of the root labeled with p, while all other nodes are not".

It is possible to express such a property with the following GCTL formulas of length linear in g: $\varphi = (\mathrm{E}^{\geq g}\mathrm{F}\, p) \wedge (\neg p) \wedge (\mathrm{AX}\, \neg p) \wedge (\mathrm{AX}\,\mathrm{AX}\,\mathrm{AX}\,\mathrm{AG}\, \neg p)$.



However, each GRADED μ-CALCULUS formulas equivalent to φ has to have an exponential size in the degree g.

Elementary model properties

GCTL*, as CTL*,

- is invariant under unwinding and partial unwinding,
- has the tree and finite model property.

However,

- it is not invariant under bisimulation,
- it is more expressive than CTL*.

All the above results also hold for GCTL.

Outline

Introduction to PATA

Partitioning alternating tree automata (PATA, for short) are symmetric automata running on infinite trees.

They are a generalization of alternating tree automata in such a way that the automaton can send copies of itself to a given number *n* of paths starting from the current node of the tree in input.

The run execution of PATA embed the one-step unfolding property.

Let $D_b^\varepsilon = \{\Diamond, \Box\} \times \mathbb{N}_b \cup \{\varepsilon\}$ be the set of abstract directions. Then, $D_b^\varepsilon \times Q$ is the set of moves that are allowed for a PATA, where Q is the set of states.

- (ε, q): change the state to q without changing the node of the input tree;
- $((\Diamond, g), q)$: there exists a set of successors of the current node in the input tree to which the state q is sent, with all degree summing up to g;
- $((\Box, g), q)$: dual of $((\Diamond, g), q)$.

Formal definition of PABT

Definition

A *partitioning alternating Büchi tree automaton* (PABT, for short) is a tuple $\mathcal{A} = \langle Q, \Sigma, b, \delta, q_0, b_0, F \rangle$ where:

1. Q: finite non-empty set of *states*;
2. Σ: finite non-empty set of *labels*;
3. $b \in \mathbb{N}$: is a *counting branching bound*;
4. $\delta : Q \times \mathbb{N}_b \times \Sigma \mapsto \mathrm{B}^+(D_b^\varepsilon \times Q)$ is a *transition function*;
5. $q_0 \in Q$: *initial state*;
6. $b_0 \in \mathbb{N}$: *initial branching degree*;
7. $F \subseteq Q \times \mathbb{N}_b$: *Büchi acceptance condition.*

Run of a PABT (I)

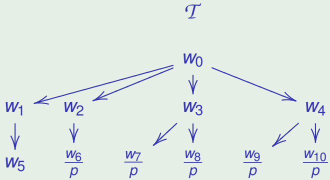
A run of a PABT is a $(T \times Q \times \mathbb{N}_b)$-labeled tree that is coherenth with the delta transition of the automaton, where T is the domain of the input tree $\mathcal{T}$.

Example

1. $\delta(q, g, \sigma) = \mathfrak{t}$, if $g = 1$ and $\sigma = \{p\}$;
2. $\delta(q, g, \sigma) = ((\Diamond, g), q)$, otherwise.

This automaton recognize all and only the trees having at least 5 paths reaching a node labeled with p.

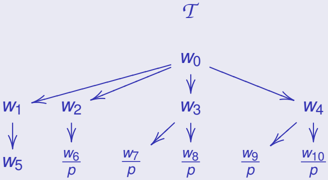
$\mathcal{A} = \langle \{q\}, \{\emptyset, \{p\}\}, 5, \delta, q, 5, \emptyset \rangle$

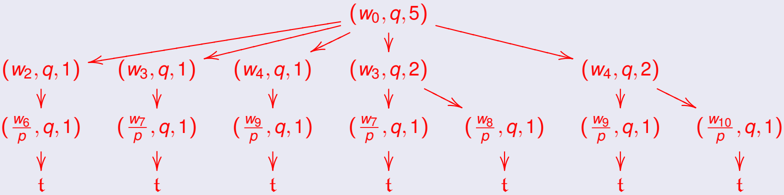


Run of a PABT (II)

$\mathcal{A} = \langle \{q\}, \{\emptyset, \{p\}\}, 5, \delta, q, 5, \emptyset \rangle$

1. $\delta(q, g, \sigma) = \mathfrak{t}$, if $g = 1$ and $\sigma = \{p\}$;
2. $\delta(q, g, \sigma) = ((\Diamond, g), q)$, otherwise.





Reduction to NBT

To evaluate the emptiness of a PABT, we first reduce it to an asymmetric non-deterministic tree automata and then we calculate the emptiness of the latter.

For the reduction we use an extension of the Miyano-Hayashi technique for tree automata. To do this, we have to face to two problems:

- PABT allows the use of ε-moves;
- there is no bound on the number of direction that a PABT can use.

The first problem is solved allocating in the NBT an apposite direction that collects all states of the PABT sent through an ε-move.

The second one is solved proving a bounded-width model property for PABT.

The emptiness for PABT is EXPTIME-COMPLETE.

Satisfiability of GCTL

The reduction of GCTL satisfiability to the emptiness of PABT is based on a variation of the classical one between CTL and ABT.

The set of states is the extended Fisher-Ladner closure of the formula.

The delta transition of the automaton is a transposition of the one-step unfolding properties of "until" and "release".

- $\delta(\langle\langle \varphi_1 \mathsf{U} \varphi_2 \rangle\rangle, 1, \sigma) = (\varepsilon, \varphi_2) \vee ((\varepsilon, \varphi_1) \wedge ((\Diamond, 1), \langle\langle \varphi_1 \mathsf{U} \varphi_2 \rangle\rangle))$.
- $\delta(\langle\langle \varphi_1 \mathsf{U} \varphi_2 \rangle\rangle, g, \sigma) = (\varepsilon, \neg\varphi_2) \wedge (\varepsilon, \varphi_1) \wedge ((\Diamond, g), \langle\langle \varphi_1 \mathsf{U} \varphi_2 \rangle\rangle)$, $g > 1$.

The satisfiability for GCTL is EXPTIME-COMPLETE.

Outline

Conclusion

In this work...

- we introduced GCTL*, i.e., CTL* augmented with Graded Quantifiers,
- we study some elementary model-theoretic properties of this logic:
    - one-step unfolding,
    - expressiveness,
    - succinctness,
    - tree and finite model property,
    - reduction to GRADED μ-CALCULUS,
- we introduce the Partitioning Alternating Tree Automata as a generalization of graded alternating tree automata and study its emptiness problem,
- finally, we show the decidability of satisfiability for GCTL using a reduction to the emptiness problem of PABT.

Outline

References

- Alessandro Ferrante, Aniello Murano: Enriched μ-Calculi Module Checking. FoSSaCS 2007: 183-197
- Alessandro Ferrante, Aniello Murano, Mimmo Parente: Enriched μ-Calculi Module Checking. Logical Methods in Computer Science 4(3) (2008)
- Alessandro Ferrante, Aniello Murano, Mimmo Parente: Enriched μ-Calculus Pushdown Module Checking. LPAR 2007: 438-453
- Alessandro Ferrante, Margherita Napoli, Mimmo Parente: Model Checking for Graded CTL. Fundam. Inform. 96(3): 323-339 (2009)
- Franz Baader et al: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003
- Erich Gradel: On The Restraining Power of Guards. J. Symb. Log. 64(4): 1719-1742 (1999)
- Orna Kupferman, Gila Morgenstern, Aniello Murano: Typeness for omega-regular Automata. Int. J. Found. Comput. Sci. 17(4): 869-884 (2006)
- Orna Kupferman, Ulrike Sattler, Moshe Y. Vardi: The Complexity of the Graded μ-Calculus. CADE 2002: 423-437
- Orna Kupferman, Moshe Y. Vardi, Pierre Wolper: An automata-theoretic approach to branching-time model checking. J. ACM 47(2): 312-360 (2000)
- Fabio Mogavero, Aniello Murano: Branching-Time Temporal Logics with Minimal Model Quantifiers. Developments in Language Theory 2009: 396-409
- Alessandro Bianco, Fabio Mogavero, Aniello Murano: Graded Computation Tree Logic. LICS 2009: 342-351
- Piero A. Bonatti, Carsten Lutz, Aniello Murano, Moshe Y. Vardi: The Complexity of Enriched Mu-Calculi. Logical Methods in Computer Science 4(3) (2008)