Graded Computation Tree Logic

Alessandro Bianco  Fabio Mogavero  Aniello Murano

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

24th IEEE Symposium on Logic in Computer Science
Los Angeles, California, USA, August 11-14, 2009
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 $K$ (Kripke structure).
- $P$ can be expressed as a temporal logic formula $\varphi$.

Then,

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

<table>
<thead>
<tr>
<th></th>
<th>M.C.</th>
<th>Sat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTL</td>
<td>PSPACE-COMPLETE</td>
<td>PSPACE-COMPLETE</td>
</tr>
<tr>
<td>CTL</td>
<td>PTIME-COMPLETE</td>
<td>EXPTIME-COMPLETE</td>
</tr>
<tr>
<td>CTL*</td>
<td>PSPACE-COMPLETE</td>
<td>2EXPTIME-COMPLETE</td>
</tr>
<tr>
<td>µ-CALCULUS</td>
<td>UPTIME ∩ COUPTIME</td>
<td>EXPTIME-COMPLETE</td>
</tr>
<tr>
<td>GRADED µ-CALCULUS$^a_b$</td>
<td>UPTIME ∩ COUPTIME</td>
<td>EXPTIME-COMPLETE</td>
</tr>
</tbody>
</table>

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.


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/connections:
- 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 useless.

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

1. Graded Computation Tree Logic
   - Syntax and Semantics
   - Properties

2. Partitioning Alternating Tree Automata
   - Structure
   - Emptiness

3. Conclusion
Syntax of GCTL* and GCTL

**Definition**

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

1. φ ::= p | ¬φ | φ ∧ φ | φ ∨ φ | E≥gψ | A<gψ,
2. ψ ::= φ | ¬ψ | ψ ∧ ψ | ψ ∨ ψ | Xψ | ˜Xψ | ψ U ψ | ψ R ψ.

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 ˜X.

For g = 1, we may write Eψ and Aψ instead of E≥gψ and A<gψ.
**Informal meaning of $E^{\geq g}$ and $A^{< g}$**

**Definition**

GCTL* **state** ($\varphi$) and **path** ($\psi$) **formulas** are built inductively as follows:

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

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 $\psi$,
- $A^{< g} \psi$: all but less than $g$ paths satisfy $\psi$.

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 $\mathcal{K} = \langle \text{AP}, W, R, L \rangle$ where:

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

A path $\pi$ of a KRIPKE $\mathcal{K}$ is a finite sequence of states compatible with the transition relation $R$ of $\mathcal{K}$.

A path $\pi'$ is a subpath of $\pi$, formally $\pi' \preceq \pi$, iff the first is a prefix of the latter.

For a set of paths $P$, we say that $\pi$ is minimal in $P$ iff, for all $\pi' \in P$, it holds that (i) $\pi \preceq \pi'$ or (ii) $\pi' \not\preceq \pi$.

By $\text{min}(P)$ we denote the antichain (i.e., the set of minimal paths) of $P$ w.r.t. $\preceq$. 
Semantics of GCTL*

Definition

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

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

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

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

- $\mathcal{P}_A(\mathcal{K}, w, \psi) = \mathcal{P}(\mathcal{K}, w) \setminus \mathcal{P}_E(\mathcal{K}, w, \neg \psi)$.
- $\neg E^g \psi \equiv A^g \neg \psi$. 
Minimality and conservativeness

Example (Minimality for F p)
\[
\mathcal{P}_A(K, w_0, F p) = P(K, w_0), \quad P(K, w_0) = \{ w_0 \cdot (w_1^* + w_2^* + w_3^*) \},
\]
\[
\text{min}(\mathcal{P}_A(K, w_0, F p)) = \{ w_0 \}.
\]
\[
K, w_0 \models E \geq 1 F p, \\
K, w_0 \not\models E \geq 2 F p.
\]

Example (Conservativeness for G p)
\[
\mathcal{P}_A(K, w_0, G p) = \{ w_0 \cdot (w_2^* + w_3^*) \}, \quad P(K, w_0) = \{ w_0 \cdot (w_1^* + w_2^* + w_3^*) \},
\]
\[
\text{min}(\mathcal{P}_A(K, w_0, G p)) = \{ w_0 \cdot (w_2 + w_3) \}.
\]
\[
K, w_0 \models E \geq 2 G p, \\
K, w_0 \not\models E \geq 3 G p.
\]
Counting nodes on trees

\[
\begin{align*}
5 & = 1 + \bigcup (2 + 2) \\
\{w_6, w_7, w_8, w_9, w_{10}\} & = \{w_6\} \bigcup \{w_7, w_8\} \bigcup \{w_9, w_{10}\} \\
\end{align*}
\]

\begin{itemize}
    \item \(\mathcal{T}, w_0 \models E^{\geq 5} F p \iff \{w_6, w_7, w_8, w_9, w_{10}\}\).
    \\
    \item \(\mathcal{T}, w_0 \models E^{\geq 3} X E^{\geq 1} F p \iff \{w_2, w_3, w_4\}\),
    \\
    \item \(\mathcal{T}, w_0 \models E^{\geq 2} X E^{\geq 2} F p \iff \{w_3, w_4\}\).
\end{itemize}

\(h_i = j\) means that there are \(j\) successors of \(w_0\) from which \(i\) paths satisfying \(F p\) start (\(h_3 = h_4 = h_5 = 0\)).
One-step unfolding (I)

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

Decompose $g$ into all possible integer partitions:

$$g = (1 + \ldots + 1) + (2 + \ldots + 2) + \ldots + g$$

$$g = 1 \cdot p_1 + 2 \cdot p_2 + \ldots + g \cdot p_g$$

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 $E^{\geq g}X\psi \equiv \bigvee \{h_i\}_i \in CP(g) \land_{i=1}^{g} E^{\geq h_i}X E^{\geq i}\psi$. 
Properties

One-step unfolding (II)

The following properties hold, where $\varphi$, $\varphi'$, and $\psi$ are, respectively, two state and a path formula:

1. $E^{\geq g}(\varphi \land \psi) \equiv \varphi \land E^{\geq g} \psi$;

2. $E^{\geq g}(\varphi \lor \psi) \equiv \begin{cases} \varphi \lor E^{\geq g} \psi, & \text{if } g = 1; \\ \neg \varphi \land E^{\geq g} \psi, & \text{otherwise}; \end{cases}$

3. $\varphi U \varphi' \equiv \varphi' \lor (\varphi \land X (\varphi U \varphi'))$.

Using this property, we are able to reduce GCTL to the GRADED $\mu$-CALCULUS with an exponential blow-up (since $|CP(g)| = O(2^{\sqrt{g}})$).
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 $g$: $\varphi = (E^{\geq g} F p) \land (\neg p) \land (AX \neg p) \land (AX AX AX AG \neg p)$.

However, each GRADED $\mu$-CALCULUS formulas equivalent to $\varphi$ has to have an exponential size in the degree $g$. 
Elementary model properties

GCTL*, as CTL*,
- is invariant under \textit{unwinding} and \textit{partial unwinding},
- has the \textit{tree} and \textit{finite model property}.

However,
- it is not invariant under \textit{bisimulation},
- it is \textit{more expressive} than CTL*.

All the above results also hold for GCTL.
Outline

1. Graded Computation Tree Logic
   - Syntax and Semantics
   - Properties

2. Partitioning Alternating Tree Automata
   - Structure
   - Emptiness

3. Conclusion
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 $D_b^\varepsilon = \{\lozenge, \square\} \times \mathbb{N}_b \cup \{\varepsilon\}$ 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.

- $(\varepsilon, q)$: change the state to $q$ without changing the node of the input tree;
- $((\lozenge, g), q)$: 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 $g$;
- $((\square, g), q)$: dual of $((\lozenge, g), q)$.
Formal definition of PABT

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. $\Sigma$: finite non-empty set of labels;
3. $b \in \mathbb{N}$: is a counting branching bound;
4. $\delta : Q \times \mathbb{N}_b \times \Sigma \mapsto B^+(D^c_b \times Q)$ 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.
A run of a PABT is a \((T \times Q \times \mathbb{N}_b)\)-labeled tree that is coherent with the delta transition of the automaton, where \(T\) is the domain of the input tree \(T\).

Example

1. \(\delta(q, g, \sigma) = t\), if \(g = 1\) and \(\sigma = \{p\}\);

2. \(\delta(q, g, \sigma) = ((\diamond, g), q)\), otherwise.

This automaton recognizes all and only the trees having at least 5 paths reaching a node labeled with \(p\).
Run of a PABT (II)

\[ A = \langle \{q\}, \{\emptyset, \{p\}\}, 5, \delta, q, 5, \emptyset \rangle \]

1. \[ \delta(q, g, \sigma) = t \text{, if } g = 1 \text{ and } \sigma = \{p\}; \]
2. \[ \delta(q, g, \sigma) = ((\Diamond, g), q), \text{ otherwise.} \]
To evaluate the emptiness of a PABT, we first reduce it to an asymmetric nondeterministic 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 $\varepsilon$-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 $\varepsilon$-move.

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

The emptiness for PABT is $\text{EXPTime-Complete}$. 
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”.

\[
\delta(\langle\varphi_1 U \varphi_2\rangle, 1, \sigma) = (\varepsilon, \varphi_2) \lor ((\varepsilon, \varphi_1) \land ((\diamond, 1), \langle\varphi_1 U \varphi_2\rangle)).
\]

\[
\delta(\langle\varphi_1 U \varphi_2\rangle, g, \sigma) = (\varepsilon, \neg \varphi_2) \land (\varepsilon, \varphi_1) \land ((\diamond, g), \langle\varphi_1 U \varphi_2\rangle), \ g > 1.
\]

The satisfiability for GCTL is \textbf{ExpTime-Complete}. 
Outline

1. Graded Computation Tree Logic
   - Syntax and Semantics
   - Properties

2. Partitioning Alternating Tree Automata
   - Structure
   - Emptiness

3. 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 $\mu$-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.
Thank you very much for your attention!
I hope my talk was enough interesting for you.