Counting CTL

Antoine Meyer (1)

Joint work with: François Laroussinie (2), Eudes Petonnet (2)

(1) LIGM, Univ. Paris Est – CNRS
(2) LIAFA / Univ. Paris 7 – CNRS

ACTS II, Chennai, Feb. 3rd 2010

Counting in formulas

Consider an ATM, and the property:
"Three mistakes forbid cash retrieval"

Counting in formulas

Consider an ATM, and the property:
"Three mistakes forbid cash retrieval"

- In CTL:

$$\neg \mathsf{EF}\Big(\mathtt{error} \wedge \mathsf{EXEF}\big(\mathtt{error} \wedge \mathsf{EXEF}(\mathtt{error} \wedge \mathsf{EF}\mathtt{money})\big)\Big)$$

or: $\neg \mathsf{EF}\Big(\mathtt{error} \wedge \mathsf{EF}_s\big(\mathtt{error} \wedge \mathsf{EF}_s(\mathtt{error} \wedge \mathsf{EF}\mathtt{money})\big)\Big)$

Counting in formulas

Consider an ATM, and the property:
"Three mistakes forbid cash retrieval"

- In CTL:

$$\neg\mathsf{EF}\Big(\mathtt{error} \wedge \mathsf{EXEF}\big(\mathtt{error} \wedge \mathsf{EXEF}(\mathtt{error} \wedge \mathsf{EF}\mathtt{money})\big)\Big)$$

or: $$\neg\mathsf{EF}\Big(\mathtt{error} \wedge \mathsf{EF}_s\big(\mathtt{error} \wedge \mathsf{EF}_s(\mathtt{error} \wedge \mathsf{EF}\mathtt{money})\big)\Big)$$

- With counting:

$$\neg\mathsf{EF}_{[\sharp\mathtt{error}\geq 3]}\,\mathtt{money}$$

Counting in formulas

"Whenever the PIN is locked, at least three erroneous attempts have been made"

Counting in formulas

"Whenever the PIN is locked, at least three erroneous attempts have been made"

- In CTL:

$$\neg \mathsf{E} \neg \mathtt{error} \mathsf{U} \mathtt{lock} \wedge \neg \mathsf{E} \neg \mathtt{error} \mathsf{U} (\mathtt{error} \wedge \mathsf{EXE} \neg \mathtt{error} \mathsf{U} \mathtt{lock}) \wedge \neg \mathsf{E} \neg \mathtt{error} \mathsf{U} \big(\mathtt{error} \wedge \mathsf{EXE} \neg \mathtt{error} \mathsf{U} (\mathtt{error} \wedge \mathsf{EXE} \neg \mathtt{error} \mathsf{U} \mathtt{lock}) \big)$$

Counting in formulas

"Whenever the PIN is locked, at least three erroneous attempts have been made"

- In CTL:

$$\neg \mathsf{E} \neg \mathtt{error} \mathsf{U} \mathtt{lock} \wedge \neg \mathsf{E} \neg \mathtt{error} \mathsf{U} (\mathtt{error} \wedge \mathsf{EXE} \neg \mathtt{error} \mathsf{U} \mathtt{lock}) \wedge \neg \mathsf{E} \neg \mathtt{error} \mathsf{U} \big(\mathtt{error} \wedge \mathsf{EXE} \neg \mathtt{error} \mathsf{U} (\mathtt{error} \wedge \mathsf{EXE} \neg \mathtt{error} \mathsf{U} \mathtt{lock})\big)$$

- With counting:

$$\neg \mathsf{EF}_{[\sharp \mathtt{error} \leq 2]}\, \mathtt{lock}$$

Counting in formulas

Other examples:

$\mathsf{EF}_{[\sharp \mathsf{EX}\texttt{Pb} < 2 \,\wedge\, \sharp\texttt{ok} > 10]}\texttt{P}, \quad \mathsf{EF}_{[\sharp\texttt{ok} - \sharp\texttt{bad} > 10]}\texttt{P}, \quad \mathsf{AG}_{[10 \cdot \sharp\texttt{ok} < 300 \cdot \sharp\texttt{bad}]}\,\bot, \ldots$

Counting in formulas

Other examples:

$\mathsf{EF}_{[\sharp \mathsf{EX}\,\mathtt{Pb} < 2 \,\wedge\, \sharp\mathtt{ok} > 10]}\mathtt{P}, \quad \mathsf{EF}_{[\sharp\mathtt{ok} - \sharp\mathtt{bad} > 10]}\mathtt{P}, \quad \mathsf{AG}_{[10 \cdot \sharp\mathtt{ok} < 300 \cdot \sharp\mathtt{bad}]}\,\bot, \ldots$

CCTL = CTL + counting constraints of the form

$$\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i - \sum_{i=1}^{m} \beta_i \cdot \sharp\psi_i \sim k$$

(and all sensible restrictions: $\ell = 1, m = 0, m = 1, \alpha_i = \beta_i = 1, \ldots$)

Counting temporal logics

- LTL with regular expressions containing quantitative constraints [Emerson,Trefler 97] $\rightsquigarrow$ exponential algorithms in $|\Phi|$ and the *value* of constants.
- CTL with constraints (with parameters) [Emerson, Trefler 99]
  Constraints as positive boolean combinations of $\sum_i P_i \leq c$
  - Model-checking $E\_U\_$ is NP-complete
  - Polynomial algorithm given for a restricted logic
- Branching-time temporal logic with general counting constraints (using freeze variables): undecidable [Yang, Mok, Wang 97].
- LTL and CTL with Presburger constraints [Bouajjani, Echahed, Habermehl 95] for infinite state processes
- (timed extensions. . .)

Outline

1. CCTL
2. Expressiveness
3. Model checking
4. Freeze variables

Outline

1. CCTL
2. Expressiveness
3. Model checking
4. Freeze variables

Counting CTL

Given $\ell, k \in \mathbb{N}$, $k' \in \mathbb{Z}$ and $\sim \in \{<, \leq, =, \geq, >\}$, we define:

$$\mathcal{C}_0 \ni C ::= \sharp\varphi \sim k$$

$$\mathcal{C}_1 \ni C ::= (\textstyle\sum_{i=1}^{\ell} \sharp\varphi_i) \sim k$$

$$\alpha\mathcal{C}_1 \ni C ::= (\textstyle\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i) \sim k$$

$$\alpha_i \in \mathbb{N}$$

$$\mathcal{C}_2 \ni C ::= (\sharp\varphi - \sharp\psi) \sim k'$$

$$\mathcal{C}_3 \ni C ::= (\textstyle\sum_{i=1}^{\ell} \pm\sharp\varphi_i) \sim k'$$

$$\alpha\mathcal{C}_3 \ni C ::= (\textstyle\sum_{i=1}^{\ell} \beta_i \cdot \sharp\varphi_i) \sim k$$

$$\beta_i \in \mathbb{Z}$$

Counting CTL

Given $\ell, k \in \mathbb{N}$, $k' \in \mathbb{Z}$ and $\sim \in \{<, \leq, =, \geq, >\}$, we define:

$$\mathcal{C}_0 \ni C ::= \sharp\varphi \sim k \qquad \mathcal{C}_2 \ni C ::= (\sharp\varphi - \sharp\psi) \sim k'$$

$$\mathcal{C}_1 \ni C ::= (\textstyle\sum_{i=1}^{\ell} \sharp\varphi_i) \sim k \qquad \mathcal{C}_3 \ni C ::= (\textstyle\sum_{i=1}^{\ell} \pm\sharp\varphi_i) \sim k'$$

$$\alpha\mathcal{C}_1 \ni C ::= (\textstyle\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i) \sim k \qquad \alpha\mathcal{C}_3 \ni C ::= (\textstyle\sum_{i=1}^{\ell} \beta_i \cdot \sharp\varphi_i) \sim k$$

$$\alpha_i \in \mathbb{N} \qquad \beta_i \in \mathbb{Z}$$

For each $\mathcal{C}$, $\mathcal{B}(\mathcal{C})$ = boolean combinations of constraints in $\mathcal{C}$

Definition

Let $\mathcal{C}$ be a set of constraints as above, the syntax of $\mathrm{CCTL}_{\mathcal{C}}$ is:

$$\varphi, \psi ::= P \mid \varphi \wedge \psi \mid \neg\varphi \mid \mathsf{E}\varphi\mathsf{U}_{[C]}\psi \mid \mathsf{A}\varphi\mathsf{U}_{[C]}\psi$$

where $P \in \mathrm{AP}$ (atomic propositions), $C \in \mathcal{C}$

Counting CTL

CCTL formulas are interpreted over states of Kripke structures

$$\mathcal{S} = \langle Q, R, \ell \rangle$$

- Q is a finite set of states
- $R \subseteq Q \times Q$ is a complete edge relation
- $\ell : Q \to 2^{\mathsf{AP}}$ is a labeling of states with atomic propositions

No costs, no weights, (no probabilities), no time . . . !

Semantics of constraints

Let π be a finite run, $\pi \models C$ depends on the interpretation of $\sharp\varphi$ over π:

$$|\pi|_\varphi \stackrel{\text{def}}{=} \left|\{j \mid 0 \le j \le |\pi| \ \wedge\ \pi(j) \models \varphi\}\right|$$

CCTL semantics

$$\begin{array}{lll} q \models \mathsf{E}\varphi\mathsf{U}_{[C]}\psi & \text{iff} & \exists \rho \in \mathsf{Runs}(q),\ \exists k \ge 0, \rho(k) \models \psi, \\ & & \rho_{|k-1} \models C,\ \text{and } \forall 0 \le i < k,\ \rho(i) \models \varphi \\ q \models \mathsf{A}\varphi\mathsf{U}_{[C]}\psi & \text{iff} & \forall \rho \in \mathsf{Runs}(q),\ \exists k \ge 0,\ \rho(k) \models \psi, \\ & & \rho_{|k-1} \models C,\ \text{and } \forall 0 \le i < k,\ \rho(i) \models \varphi \end{array}$$

Examples of formulas

- $\mathsf{EX}\,\varphi \stackrel{\text{def}}{=} \mathsf{EF}_{[\sharp\top=1]}\,\varphi$

Examples of formulas

- $\mathsf{EX}\,\varphi \stackrel{\text{def}}{=} \mathsf{EF}_{[\sharp\top=1]}\,\varphi$
- $\mathsf{E}\varphi\mathsf{U}_{[C]}\psi \stackrel{\text{def}}{=} \mathsf{EF}_{[C\wedge\sharp(\neg\varphi)=0]}\psi$

Examples of formulas

- $\mathsf{EX}\,\varphi \stackrel{\text{def}}{=} \mathsf{EF}_{[\sharp\top=1]}\,\varphi$
- $\mathsf{E}\varphi\mathsf{U}_{[C]}\psi \stackrel{\text{def}}{=} \mathsf{EF}_{[C\wedge\sharp(\neg\varphi)=0]}\psi$
- $\mathsf{E}\varphi\mathsf{U}_{<5}\psi \stackrel{\text{def}}{=} \mathsf{E}\varphi\mathsf{U}_{[\sharp\mathtt{tick}<5]}\psi$ – (TCTL over KSs with tick).

Examples of formulas

- $\mathsf{EX}\,\varphi \stackrel{\text{def}}{=} \mathsf{EF}_{[\sharp\top=1]}\,\varphi$
- $\mathsf{E}\varphi\mathsf{U}_{[C]}\psi \stackrel{\text{def}}{=} \mathsf{EF}_{[C\wedge\sharp(\neg\varphi)=0]}\psi$
- $\mathsf{E}\varphi\mathsf{U}_{<5}\psi \stackrel{\text{def}}{=} \mathsf{E}\varphi\mathsf{U}_{[\sharp\mathtt{tick}<5]}\psi$ – (TCTL over KSs with tick).
- For an ATM: "it is not possible to get money when three mistakes are made in the same session":

$$\mathsf{AG}\left(\neg\mathsf{EF}_{[\sharp\mathtt{error}\geq 3\wedge\sharp\mathtt{reset}=0]}\mathtt{money}\right)$$

Examples of formulas

- $\mathsf{EX}\,\varphi \stackrel{\text{def}}{=} \mathsf{EF}_{[\sharp\top=1]}\,\varphi$
- $\mathsf{E}\varphi\mathsf{U}_{[C]}\psi \stackrel{\text{def}}{=} \mathsf{EF}_{[C\wedge\sharp(\neg\varphi)=0]}\psi$
- $\mathsf{E}\varphi\mathsf{U}_{<5}\psi \stackrel{\text{def}}{=} \mathsf{E}\varphi\mathsf{U}_{[\sharp\mathtt{tick}<5]}\psi$ – (TCTL over KSs with tick).
- For an ATM: "it is not possible to get money when three mistakes are made in the same session":

$$\mathsf{AG}\big(\neg\mathsf{EF}_{[\sharp\mathtt{error}\geq 3\wedge\sharp\mathtt{reset}=0]}\mathtt{money}\big)$$

- $\mathsf{AG}\,(\,\mathsf{EF}_{[\sharp(\mathsf{EX}\mathtt{alarm})\leq 5]}\mathtt{init}\,)$ "It is always possible to reach `init` along a path where less than 5 states have an `alarm` state as successor."

Examples of formulas

- The bounded waiting property with bound 10 for a mutual exclusion algorithm with n processes:

$$\mathsf{AG} \bigwedge_i \left(\mathtt{request}_i \Rightarrow \neg \mathsf{EF}_{[\sum_{j \neq i} \sharp \mathtt{CS}_j > 10 \wedge \sharp \mathtt{CS}_i = 0]} \top\right)$$

Examples of formulas

- The bounded waiting property with bound 10 for a mutual exclusion algorithm with n processes:

$$\mathsf{AG} \bigwedge_i \left(\mathtt{request}_i \Rightarrow \neg \mathsf{EF}_{[\sum_{j \neq i} \sharp \mathtt{CS}_j > 10 \wedge \sharp \mathtt{CS}_i = 0]} \top\right)$$

- "The number of receive events can not exceed the number of send events":

$$\mathsf{AG}_{[\sharp send - \sharp receive < 0]} \bot$$

Examples of formulas

- The bounded waiting property with bound 10 for a mutual exclusion algorithm with n processes:

$$\mathsf{AG} \bigwedge_i \left(\mathtt{request}_i \Rightarrow \neg \mathsf{EF}_{[\sum_{j \neq i} \sharp \mathtt{CS}_j > 10 \wedge \sharp \mathtt{CS}_i = 0]} \top \right)$$

- "The number of receive events can not exceed the number of send events":

$$\mathsf{AG}_{[\sharp send - \sharp receive < 0]} \bot$$

- Quantitative fairness: "The φ_i's occur infinitely often along every run and there is no sub-run where φ_1 holds for more than 10 states and φ_2 holds for less than 4 states":

$$\mathsf{AG}\,\mathsf{AF}_{[\bigwedge_i 5 \leq \sharp \varphi_i \leq 10]} \top$$

Outline

1. CCTL
2. Expressiveness
3. Model checking
4. Freeze variables

$\mathcal{B}(\alpha\mathcal{C}_1) : \bigwedge \bigvee \sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i \sim k$ $\alpha\mathcal{C}_2 : \sharp\varphi - \sharp\psi \sim k$

Proposition

Any $CCTL_{\mathcal{B}(\alpha\mathcal{C}_1)}$ formula can be translated into CTL.

Idea: manually count occurrences of events using nested U modalities and consider all possible shuffles of such occurrences.

Expressiveness

$\mathcal{B}(\alpha\mathcal{C}_1) : \bigwedge \bigvee \sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i \sim k$ $\alpha\mathcal{C}_2 : \sharp\varphi - \sharp\psi \sim k$

Proposition

Any $CCTL_{\mathcal{B}(\alpha\mathcal{C}_1)}$ formula can be translated into CTL.

Idea: manually count occurrences of events using nested U modalities and consider all possible shuffles of such occurrences.

Proposition

The $CCTL_{\mathcal{C}_2}$ formula $\varphi = AG_{[\sharp A - \sharp B < 0]}\bot$ cannot be translated into CTL.

Idea: the set of models of any CTL formula can be recognized by an alternating tree automaton. This is not the case for φ.

Succinctness

$$\mathcal{B}(\alpha\mathcal{C}_1) : \bigwedge \bigvee \sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i \sim k$$

$\mathsf{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$ formulas can be translated into CTL, but in these constraints, there are three potential sources of concision:

- Binary encoding of constants
- Boolean combinations in constraints
- Sums of counting expressions

Succinctness

$$\mathcal{B}(\alpha\mathcal{C}_1) : \bigwedge \bigvee \sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i \sim k$$

$\mathsf{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$ formulas can be translated into CTL, but in these constraints, there are three potential sources of concision:

- Binary encoding of constants
- Boolean combinations in constraints
- Sums of counting expressions

Only the first two yield an exponential improvement in succinctness

Succinctness – Binary encoding

$\mathcal{C}_0 : \sharp\varphi \sim k$

The previous translation of $\mathsf{EF}_{[\sharp A = k]} B$ into CTL yields an exponential formula (it uses k nested modalities)

Succinctness – Binary encoding

$\mathcal{C}_0 : \sharp\varphi \sim k$

The previous translation of $\mathsf{EF}_{[\sharp A = k]} B$ into CTL yields an exponential formula (it uses k nested modalities)

Proposition

$CCTL_{\mathcal{C}_0}$ can be exponentially more succinct than CTL

Idea: TCTL formulas $\mathsf{EF}_{<k} A$ and $\mathsf{EF}_{>k} A$ do not admit any equivalent CTL formula of temporal height less than k [Laroussinie, Schnoebelen, Turuani 01]

Succinctness – Boolean combinations

$\mathcal{B}(\mathcal{C}_0) : \bigwedge \bigvee \sharp\varphi \sim k$

Proposition

$CCTL_{\mathcal{B}(\mathcal{C}_0)}$ with unary encoding of integers can be exponentially more succinct than CTL.

Succinctness – Boolean combinations

$\mathcal{B}(\mathcal{C}_0) : \bigwedge \bigvee \sharp\varphi \sim k$

Proposition

$CCTL_{\mathcal{B}(\mathcal{C}_0)}$ with unary encoding of integers can be exponentially more succinct than CTL.

Idea: any CTL formula equivalent to ψ:

$$\psi \;=\; \mathsf{E}(\mathsf{F}\, P_0 \wedge \ldots \wedge \mathsf{F}\, P_n)$$

must be of length exponential in n [Wilke 99, Adler, Immerman 03]

$$\psi \;\equiv\; \mathsf{EF}_{[\bigwedge_i \sharp P_i \geq 1]} \top$$

(binary encoding of constants not needed)

Succinctness – Sums

$\mathcal{C}_1 : \sum_i \sharp\varphi_i \sim k$

Proposition

For every formula $\Phi \in CCTL_{\mathcal{C}_1}$ with unary encoding, there exists an equivalent CTL formula of DAG-size polynomial in $|\Phi|$.

Succinctness – Sums

$\mathcal{C}_1 : \sum_i \sharp\varphi_i \sim k$

Proposition

For every formula $\Phi \in CCTL_{\mathcal{C}_1}$ with unary encoding, there exists an equivalent CTL formula of DAG-size polynomial in $|\Phi|$.

Example: $\Phi = \mathsf{EF}_{\sum_i \sharp P_i = K} A$ is equivalent to Ψ_K with:

$$\Psi_k \stackrel{\text{def}}{=} \mathsf{E}\big(\textstyle\bigwedge_i \bar{P}_i\big)\mathsf{U}\big(\textstyle\bigvee_i P_i \wedge \beta_{k,1,\bot}\big) \qquad (k > 0)$$

$$\Psi_0 \stackrel{\text{def}}{=} \mathsf{E}\big(\textstyle\bigwedge_i \bar{P}_i\big)\mathsf{U}\, A \qquad \Psi_{-1} \stackrel{\text{def}}{=} \bot$$

$$\beta_{k,i,\epsilon} \stackrel{\text{def}}{=} (P_i \wedge \beta_{k-1,i+1,\top}) \vee (\bar{P}_i \wedge \beta_{k,i+1,\epsilon}) \qquad (i < n)$$

$$\beta_{k,n,\top} \stackrel{\text{def}}{=} (P_n \wedge \mathsf{EX}\, \Psi_{k-1}) \vee (\bar{P}_n \wedge \mathsf{EX}\, \Psi_k)$$

$$\beta_{k,n,\bot} \stackrel{\text{def}}{=} P_n \wedge \mathsf{EX}\, \Psi_{k-1}$$

Comparison with Past

Counting constraints deal with past events !

We could use past-time modalities:

$$\mathsf{AG}\big(\texttt{money} \;\Rightarrow\; \neg\, \mathsf{F}_s^{-1}(\texttt{error} \wedge \mathsf{F}_s^{-1}(\texttt{error} \wedge \mathsf{F}_s^{-1}\texttt{error}))\big)$$

Comparison with Past

Counting constraints deal with past events !

We could use past-time modalities:

$$\mathsf{AG}\big(\texttt{money} \Rightarrow \neg\, \mathsf{F}_s^{-1}(\texttt{error} \wedge \mathsf{F}_s^{-1}(\texttt{error} \wedge \mathsf{F}_s^{-1}\texttt{error}))\big)$$

+ Past-time modalities allow us to express properties over the ordering of the events.
+ They (often) increase the expressive power (compared to CTL).
+ Boolean combinations are directly handled. . .

− Counting constraints are still more succinct.
− Complexity (model-checking $\mathsf{CTL} + \mathsf{F}^{-1}$ is PSPACE-complete)

Outline

1. CCTL
2. Expressiveness
3. Model checking
4. Freeze variables

Model checking $\mathrm{CCTL}_{\mathcal{C}_0}$ and $\mathrm{CCTL}_{\mathcal{C}_1}$

$\mathrm{CCTL}_{\mathcal{C}_0}$: $\sharp\varphi \sim k$

$\mathrm{CCTL}_{\mathcal{C}_1}$: $(\sum_{i=1}^{\ell} \sharp\varphi_i) \sim k$

Theorem

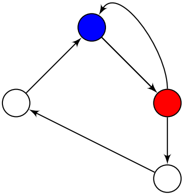
Model-checking $CCTL_{\mathcal{C}_1}$ and $CCTL_{\mathcal{C}_0}$ is P*-complete*

Idea: Reduction to a model-checking problem for TCTL formulas over Kripke structures with 0/1 durations

Model checking $\mathrm{CCTL}_{\mathcal{C}_0}$ and $\mathrm{CCTL}_{\mathcal{C}_1}$

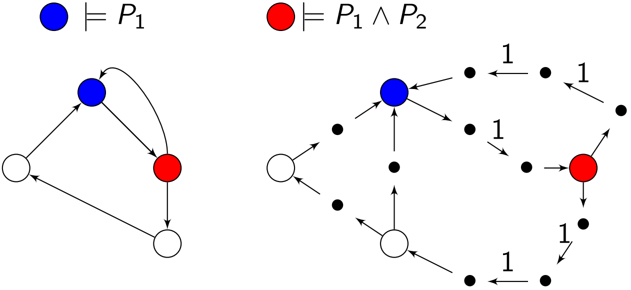
Example: $\mathcal{S} = (Q, R, \ell)$, and $\Phi = \mathsf{E}\varphi\mathsf{U}_{[\sharp P_1 + \sharp P_2 \sim k]}\psi$

$\models P_1$ $\models P_1 \wedge P_2$



Model checking $\mathsf{CCTL}_{\mathcal{C}_0}$ and $\mathsf{CCTL}_{\mathcal{C}_1}$

Example: $\mathcal{S} = (Q, R, \ell)$, and $\Phi = \mathsf{E}\varphi\mathsf{U}_{[\sharp P_1 + \sharp P_2 \sim k]}\psi$



Model-checking $\mathrm{CCTL}_{\mathcal{C}_1}$

Proof: $\mathcal{S} = (Q, R, \ell)$, and $\Phi = \mathsf{E}\psi\mathsf{U}_{[C]}\psi'$ with $C \stackrel{\text{def}}{=} \sum_{i=1}^{\ell} \sharp\varphi_i \sim k$

$\forall q \in Q$: $|q|_C \stackrel{\text{def}}{=} \big|\{i \mid q \models \varphi_i\}\big|$

We build the $\mathrm{DKS}^{0/1}$ $\mathcal{S}' = (Q', R', \ell')$ as follows:

- $Q' \stackrel{\text{def}}{=} Q \cup \bigcup_{q \in Q} \{q_i \mid 0 \leq i \leq |q|_C\}$,
- $R' \stackrel{\text{def}}{=} \{q \xrightarrow{0} q_0\} \cup \{q_i \xrightarrow{1} q_{i+1} \mid i < |q|_C\}$
  $\cup \{q_n \xrightarrow{0} q' \mid (q, q') \in R, n = |q|_C\}$,
- $\ell'(q_i) = \varnothing$ and $\ell'(q) = \ell(q) \cup \{\mathtt{ok}\}$

$\rho \models_{\mathcal{S}} \psi\mathsf{U}_{[C]}\psi'$ if and only if $\tilde{\rho} \models_{\mathcal{S}'} (\mathtt{ok} \Rightarrow \psi)\mathsf{U}_{[\sim k]}(\mathtt{ok} \wedge \psi')$

Model-checking $\mathsf{CCTL}_{\mathcal{C}_2}$

$$\mathsf{CCTL}_{\mathcal{C}_2} : (\sharp\varphi - \sharp\psi) \sim k$$

Theorem

The model-checking problem for $\mathsf{CCTL}_{\mathcal{C}_2}$ is P*-complete*

Model-checking $\mathrm{CCTL}_{\mathcal{C}_2}$

$\mathrm{CCTL}_{\mathcal{C}_2} : (\sharp\varphi - \sharp\psi) \sim k$

Theorem

The model-checking problem for $CCTL_{\mathcal{C}_2}$ is P*-complete*

Let $\mathcal{S} \stackrel{\text{def}}{=} (Q, R, \ell)$

Case 1: $\Phi \stackrel{\text{def}}{=} \mathsf{E}\varphi' \mathsf{U}_{[C]} \psi'$ with $C \stackrel{\text{def}}{=} (\sharp\varphi - \sharp\psi) \sim k$

$\forall q \in Q$, we define $|q|_C \in \{-1, 0, 1\}$

Let $G_{\mathcal{S}} = (S', R', w)$ be the weighted graph such that:

- S' contains only S states satisfying $\mathsf{E}\varphi' \mathsf{U} \psi'$;
- R' is R restricted to $S' \times S'$;
- $w(q, q') \stackrel{\text{def}}{=} |q|_C$ if $q \models \varphi'$, and 0 otherwise

Model-checking $\mathrm{CCTL}_{\mathcal{C}_2}$

$C \stackrel{\text{def}}{=} (\sharp\varphi - \sharp\psi) \leq k$: shortest paths in $G_{\mathcal{S}}$ + reachability of negative cycles

$C \stackrel{\text{def}}{=} (\sharp\varphi - \sharp\psi) = k$: with $k \geq 0$

Compute $R_k \stackrel{\text{def}}{=} \{(q, q') \in S'^2 \mid \exists\sigma,\ |q\sigma q'|_C = k\}$ as follows:

- $R_k = R_{\lfloor k/2 \rfloor} \cdot R_{\lfloor k/2 \rfloor} \cdot R_{(k \mod 2)}$
- $R_1 \stackrel{\text{def}}{=} R_0 \cdot \xrightarrow{1} \cdot R_0$
- R_0 is the least solution of:

$$X = (\xrightarrow{0})^* \cup X \cdot (\xrightarrow{1} \cdot X \cdot \xrightarrow{-1} \cup \xrightarrow{-1} \cdot X \cdot \xrightarrow{1}) \cdot X$$

$\Rightarrow q \models \Phi$ iff $(q, q') \in R_k$ for some q' satisfying ψ'

Model-checking $\mathrm{CCTL}_{\mathcal{C}_2}$

$C \stackrel{\text{def}}{=} (\sharp\varphi - \sharp\psi) \leq k$: shortest paths in $G_{\mathcal{S}}$ + reachability of negative cycles

$C \stackrel{\text{def}}{=} (\sharp\varphi - \sharp\psi) = k$: with $k \geq 0$

Compute $R_k \stackrel{\text{def}}{=} \{(q, q') \in S'^2 \mid \exists\sigma,\ |q\sigma q'|_C = k\}$ as follows:

- $R_k = R_{\lfloor k/2 \rfloor} \cdot R_{\lfloor k/2 \rfloor} \cdot R_{(k \mod 2)}$
- $R_1 \stackrel{\text{def}}{=} R_0 \cdot \xrightarrow{1} \cdot R_0$
- R_0 is the least solution of:

$$X = (\xrightarrow{0})^* \cup X \cdot (\xrightarrow{1} \cdot X \cdot \xrightarrow{-1} \cup \xrightarrow{-1} \cdot X \cdot \xrightarrow{1}) \cdot X$$

$\Rightarrow q \models \Phi$ iff $(q, q') \in R_k$ for some q' satisfying ψ'

Case 2: $\Phi \stackrel{\text{def}}{=} \mathrm{EG}_{[C \wedge \sharp\varphi' = 0]}\psi'$: ...

Model-checking $\mathsf{CCTL}_{\mathcal{C}_3}$

$$\mathsf{CCTL}_{\mathcal{C}_3} : \left(\sum_{i=1}^{\ell} \pm \cdot \sharp\varphi_i\right) \sim k$$

Theorem

The model-checking problem for and $CCTL_{\mathcal{C}_3}$ is P*-complete*

Each state contributes to a cost $d \in \{-M, \ldots, M\}$ with $M \leq |C|$: same technique as previously

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_0)}$

$\mathcal{B}(\mathcal{C}_0) : \bigwedge \bigvee \sharp\varphi \sim k$

Theorem

The model-checking problem for $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_0)}$ is Δ_2^{P}-hard

Reduction from SNSAT (derived from the reduction done for CTL^+ [Laroussinie, Markey, Schnoebelen 01]

SNSAT: collection of equations $z_i = \exists \bar{X}.\varphi_i(z_1, \ldots, z_{i-1}, \bar{X})$

Model-checking CCTL$_{\alpha\mathcal{C}_1}$

$\alpha\mathcal{C}_1 : (\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i) \sim k,\ \alpha_i \in \mathbb{N}$

Theorem

The model-checking problem for CCTL$_{\alpha\mathcal{C}_1}$ is Δ_2^{P}-hard

Model-checking CCTL$_{\alpha\mathcal{C}_1}$

$\alpha\mathcal{C}_1$: $(\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i) \sim k$, $\alpha_i \in \mathbb{N}$

Theorem

The model-checking problem for CCTL$_{\alpha\mathcal{C}_1}$ is Δ_2^{P}-hard

Reduction from the model-checking problem for TCTL over Kripke structures with integer durations (DKS)

Let $\mathcal{S} = (Q, R_{\mathcal{S}}, \ell)$ be a DKS
For every transition $q \xrightarrow{k} q'$ in $\mathcal{S}$, we add a new state between q and q' and labeled with only P_k
The TCTL formula $\mathsf{E}\varphi\mathsf{U}_{\sim m}\psi$ is replaced by:

$$\mathsf{E}(\mathtt{ok} \Rightarrow \widetilde{\varphi})\, \mathsf{U}_{[C]}\, (\mathtt{ok} \wedge \widetilde{\psi}) \qquad \text{with} \qquad C \stackrel{\text{def}}{=} \sum_{d \in W} d \cdot \sharp P_d \sim m$$

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$

$\mathcal{B}(\alpha\mathcal{C}_1) : \bigvee\bigwedge(\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i) \sim k$, $\alpha_i \in \mathbb{N}$

Theorem

The model-checking problem for $\mathrm{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$ is in Δ_2^{P}

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$

$$\mathcal{B}(\alpha\mathcal{C}_1) : \bigvee\bigwedge(\textstyle\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i) \sim k,\ \alpha_i \in \mathbb{N}$$

Theorem

The model-checking problem for $\mathrm{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$ is in Δ_2^{P}

Based on the Parikh image of the runs satisfying $\mathrm{EF}_{[C]}\psi$

- we can assume that $|\rho|$ is in $O(|Q| \cdot 2^{|C|})$;
- check in polynomial time that a guessed Parikh image corresponds to some path;
- check that it verifies the formula

For $\mathrm{EG}_{[C]}\psi$ we are looking at infinite runs, but $(\sum \alpha_i \cdot \sharp\varphi_i) \sim k$ may change its truth value at most twice

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_2)}$

$\mathcal{B}(\mathcal{C}_2) : \bigwedge\bigvee(\sharp\varphi - \sharp\psi) \sim k$

Theorem

The model-checking problem for $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_2)}$ is undecidable

Reduction from the halting problem of a two-counter machine $\mathcal{M}$: $\mathcal{M}$ does not halt *if and only if* $q_1 \models_{\mathcal{S}_\mathcal{M}} \mathsf{EG}_{[C]}\bot$ with:

$$
\begin{aligned}
C &\stackrel{\text{def}}{=} (\sharp\mathsf{halt} \geq 1) \ \vee C_{bad} \\
C_{bad} &\stackrel{\text{def}}{=} \bigvee_{X \in \{\mathtt{C},\mathtt{D}\}}\big(\quad (\sharp\varphi_X^+ - \sharp\varphi_X^- < 0) \\
&\qquad\qquad \vee (\sharp\varphi_X^+ - \sharp\varphi_X^- > 0 \wedge \sharp\mathsf{ko}_X - \sharp\mathsf{ok}_X > 0)\big)
\end{aligned}
$$

Outline

1. CCTL
2. Expressiveness
3. Model checking
4. Freeze variables

CCTL with freeze variables

Definition

Let V be a set of variables.

$$\mathsf{CCTL}^{\mathsf{v}} \ni \varphi, \psi ::= P \mid \varphi \wedge \psi \mid \neg\varphi \mid z[\psi].\varphi \mid C \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

where $P \in \mathsf{AP}$ and C is a constraint $\sum_{i=1}^{\ell} \alpha_i \cdot z_i \sim c$
with $z_i \in V$, $\alpha_i, c \in \mathbb{N}$, and $\sim \in \{<, \leq, =, \geq, >\}$.

CCTL with freeze variables

Definition

Let V be a set of variables.

$$\mathsf{CCTL}^{\mathsf{v}} \ni \varphi, \psi ::= P \mid \varphi \wedge \psi \mid \neg\varphi \mid z[\psi].\varphi \mid C \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

where $P \in \mathsf{AP}$ and C is a constraint $\sum_{i=1}^{\ell} \alpha_i \cdot z_i \sim c$
with $z_i \in V$, $\alpha_i, c \in \mathbb{N}$, and $\sim \in \{<, \leq, =, \geq, >\}$.

For example:

$$\mathsf{EF}_{[\sharp P \leq 5 \wedge \sharp P' > 2]} A \equiv z[P].z'[P'].\mathsf{EF}(z \leq 5 \wedge z' > 2 \wedge A)$$

CCTL with freeze variables

Definition

Let V be a set of variables.

$\mathsf{CCTL}^{\mathsf{v}} \ni \varphi, \psi ::= P \mid \varphi \wedge \psi \mid \neg\varphi \mid z[\psi].\varphi \mid C \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$

where $P \in \mathsf{AP}$ and C is a constraint $\sum_{i=1}^{\ell} \alpha_i \cdot z_i \sim c$
with $z_i \in V$, $\alpha_i, c \in \mathbb{N}$, and $\sim \in \{<, \leq, =, \geq, >\}$.

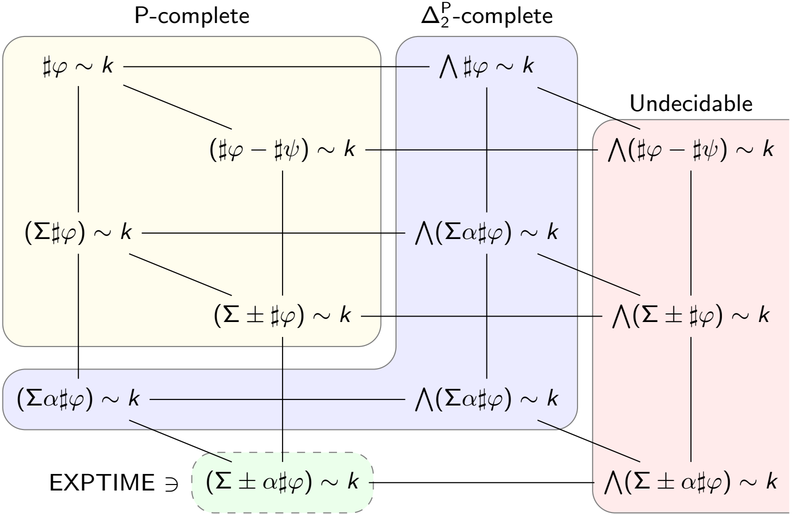
For example:

$$\mathsf{EF}_{[\sharp P \leq 5 \wedge \sharp P' > 2]} A \;\equiv\; z[P].z'[P'].\mathsf{EF}(z \leq 5 \wedge z' > 2 \wedge A)$$

Theorem

Model checking closed $CCTL^{\mathsf{v}}$ formulas is PSPACE*-complete.*

Conclusion


P-complete
Δ_2^P-complete
Undecidable
$\sharp\varphi \sim k$
$\bigwedge \sharp\varphi \sim k$
$(\sharp\varphi - \sharp\psi) \sim k$
$\bigwedge(\sharp\varphi - \sharp\psi) \sim k$
$(\Sigma\sharp\varphi) \sim k$
$\bigwedge(\Sigma\alpha\sharp\varphi) \sim k$
$(\Sigma \pm \sharp\varphi) \sim k$
$\bigwedge(\Sigma \pm \sharp\varphi) \sim k$
$(\Sigma\alpha\sharp\varphi) \sim k$
$\bigwedge(\Sigma\alpha\sharp\varphi) \sim k$
EXPTIME $\ni$
$(\Sigma \pm \alpha\sharp\varphi) \sim k$
$\bigwedge(\Sigma \pm \alpha\sharp\varphi) \sim k$