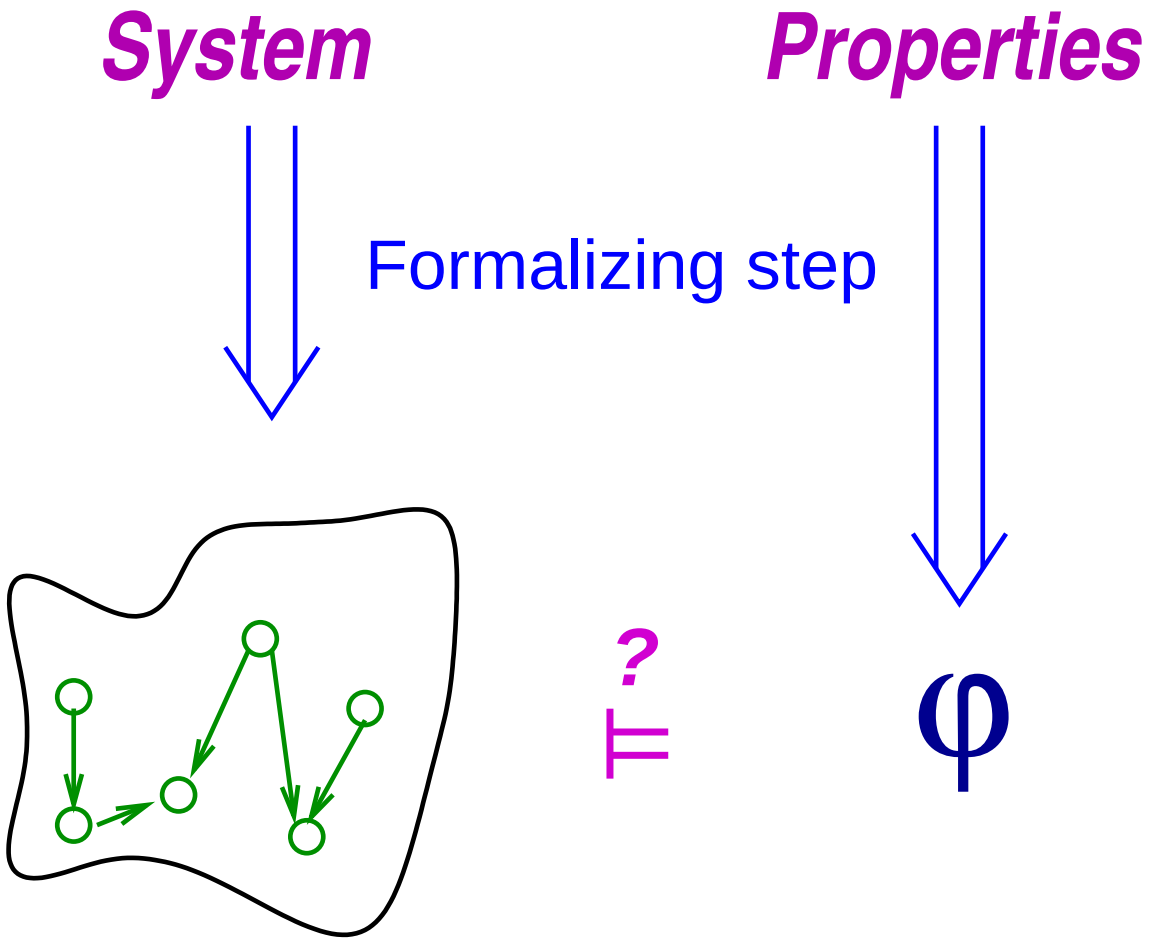
Counting Temporal Logics

François Laroussinie (1)
Joint work with: Antoine Meyer (2), Eudes Petonnet (1)
FOSSACS'10 & TIME'10

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

September 2010

Model checking



Classical framework

The system is modeled with an automaton, a Kripke structure, a Petri net,...

The specification is stated with a Temporal Logic.

The correctness of the system corresponds to a set of TL formulas that have to be satisfied by the model.

Classical framework

The system is modeled with an automaton, a Kripke structure, a Petri net,...

The specification is stated with a Temporal Logic.

The correctness of the system corresponds to a set of TL formulas that have to be satisfied by the model.

Then just press the button ! (and wait...)

Classical framework

The system is modeled with an automaton, a Kripke structure, a Petri net,...

The specification is stated with a Temporal Logic.

The correctness of the system corresponds to a set of TL formulas that have to be satisfied by the model.

Then just press the button ! (and wait...)

Choose the right models !

Temporal logics for the specification

Classical Temporal Logics (CTL, LTL,. . .) are very convenient to specify reactive systems

- good expressive power,
- natural semantics,
- succinctness,
- efficient (. . .) decision procedures (and tools !),
- nice extensions (timed TL, probabilistic TL,. . .).

Temporal logics for the specification

Classical Temporal Logics (CTL, LTL,...) are very convenient to specify reactive systems

- good expressive power,
- natural semantics,
- succinctness,
- efficient (...) decision procedures (and tools !),
- nice extensions (timed TL, probabilistic TL,...).

examples...

Linear-time Temporal Logic

Formulas built from AP, $\wedge$, $\neg$, U, X, S, X^{-1}, ...

- $\varphi \mathsf{U} \psi$: φ holds for every suffix until ψ is satisfied.
- $\mathsf{F}\varphi \stackrel{\text{def}}{=} \top \mathsf{U} \varphi$: "eventually φ"
- $\mathsf{G}\psi \stackrel{\text{def}}{=} \neg \mathsf{F} \neg \varphi$: "always φ"
- $\mathsf{X}\psi$: "The next state satisfies φ"

Formulas are interpreted over infinite executions: $q_0 \cdot q_1 \cdot q_2 \cdots$

$\mathcal{S} \models \Phi \Leftrightarrow \Phi$ holds for any run in $\mathcal{S}$.

- $\mathsf{G}(\texttt{Pb} \Rightarrow \mathsf{F}\,\texttt{Alarm})$
- $(\mathsf{GF}\,\texttt{P}_1) \Rightarrow (\mathsf{GF}\,\texttt{P}_2)$

Branching-time Temporal Logic

Formulas contain also path quantifiers: E and A.

Formulas are interpreted over states of Kripke structures.

$\mathcal{S} \models \Phi \Leftrightarrow \Phi$ holds for q_{init}.

- $\mathsf{E}\left(\mathtt{P}_1\ \mathsf{U}\ (\mathsf{AX}\ \mathtt{P}_2)\right)$
- $\mathsf{AG}(\mathtt{Pb} \Rightarrow \mathsf{AF}\ \mathtt{Alarm})$
- $\mathsf{AG}(\mathtt{Pb} \Rightarrow \mathsf{EF}\ \mathtt{Alarm})$

Counting. . .

Here our aim is to make the specifications of counting properties easy to write in TL.

Adding counting constraints in formulas

Consider an ATM.

"three mistakes forbid cash retrieval"

Adding counting constraints in formulas

Consider an ATM.

"three mistakes forbid cash retrieval"

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

Adding counting constraints in formulas

Consider an ATM.

"three mistakes forbid cash retrieval"

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

$$\neg \mathsf{EF}\Big(\texttt{error} \wedge \mathsf{EF}_s\Big(\texttt{error} \wedge \mathsf{EF}_s(\texttt{error} \wedge \mathsf{EF}\texttt{money})\Big)\Big)$$

Adding counting constraints in formulas

Consider an ATM.

"three mistakes forbid cash retrieval"

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

$$\neg \mathsf{EF}\Big(\texttt{error} \wedge \mathsf{EF}_s\Big(\texttt{error} \wedge \mathsf{EF}_s(\texttt{error} \wedge \mathsf{EF}\texttt{money})\Big)\Big)$$

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

Adding counting constraints in formulas

Consider an ATM.

"three mistakes forbid cash retrieval"

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

$$\neg \mathsf{EF}\Big(\texttt{error} \wedge \mathsf{EF}_s\Big(\texttt{error} \wedge \mathsf{EF}_s(\texttt{error} \wedge \mathsf{EF}\texttt{money})\Big)\Big)$$

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

$$\neg \mathsf{E}\neg\texttt{error}\mathsf{U}\texttt{lock} \wedge$$
$$\neg \mathsf{E}\neg\texttt{error}\mathsf{U}(\texttt{error} \wedge \mathsf{EXE}\neg\texttt{error}\mathsf{U}\texttt{lock}) \wedge$$
$$\neg \mathsf{E}\neg\texttt{error}\mathsf{U}\Big(\texttt{error} \wedge \mathsf{EXE}\neg\texttt{error}\mathsf{U}(\texttt{error} \wedge \ldots \wedge \mathsf{EXE}\neg\texttt{error}\mathsf{U}\texttt{lock})\Big)$$

With counting constraints

"three mistakes forbid cash retrieval"

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

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

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

Counting in formulas

Other examples :

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

Counting in formulas

Other examples :

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

Objectives:
study the extensions of LTL and LTL with counting constraints of the form. . .

$$\bigwedge \bigvee \Big(\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i \;\sim k \Big)$$

with $\alpha_i \in \mathbb{Z}$. . . and all sensible restrictions. . .

Counting extensions of TL

- LTL with some regular expressions with quantitative constraints [ET97] $\rightsquigarrow$ exponential algo. in $|\Phi|$ and the *value* of constants.
- CTL with constraints (with parameters) [ET99].
  Constraints are positive BC of $\sum_i P_i \leq c$ with $P_i \in \mathrm{AP}$:
  - Model-checking E_U_ is shown to be NP-complete;
  - A polynomial algorithm is given for a restricted logic.
- A branching-time TL with general counting constraint (using freeze var.) is undecidable [YMW97].
- LTL and CTL with Presburger constraints [BEH95b, BEH95a] for some infinite state processes.
- Sugar/PSL [psl03]: extension of LTL.
- (timed extensions...)

Outline

Outline

Kripke structures

CTL formulas are interpreted over the states of a Kripke structure:

AP = a set of atomic propositions.

$\mathcal{S} = \langle Q, q_0, \rightarrow, \ell \rangle$

- Q is finite set of states,
- $q_0 \in Q$ is the initial state,
- $\rightarrow \subseteq Q \times Q$ is total accessibility relation, and
- $\ell : Q \rightarrow 2^{\mathsf{AP}}$ is a labeling of states with atomic propositions.

Kripke structures

CTL formulas are interpreted over the states of a Kripke structure:

AP = a set of atomic propositions.

$\mathcal{S} = \langle Q, q_0, \rightarrow, \ell \rangle$

- Q is finite set of states,
- $q_0 \in Q$ is the initial state,
- $\rightarrow \subseteq Q \times Q$ is total accessibility relation, and
- $\ell : Q \rightarrow 2^{\mathsf{AP}}$ is a labeling of states with atomic propositions.

An execution: $\rho : s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \cdots$

$\rho(i) \stackrel{\text{def}}{=} s_i, \quad \rho^i \stackrel{\text{def}}{=} s_i \rightarrow s_{i+1} \cdots, \quad \rho_{|i} \stackrel{\text{def}}{=} s_0 \rightarrow \cdots \rightarrow s_i$

Kripke structures

CTL formulas are interpreted over the states of a Kripke structure:

AP = a set of atomic propositions.

$\mathcal{S} = \langle Q, q_0, \rightarrow, \ell \rangle$

- Q is finite set of states,
- $q_0 \in Q$ is the initial state,
- $\rightarrow \subseteq Q \times Q$ is total accessibility relation, and
- $\ell : Q \rightarrow 2^{\mathsf{AP}}$ is a labeling of states with atomic propositions.

An execution: $\rho : s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \cdots$
$\rho(i) \stackrel{\text{def}}{=} s_i, \quad \rho^i \stackrel{\text{def}}{=} s_i \rightarrow s_{i+1} \cdots, \quad \rho_{|i} \stackrel{\text{def}}{=} s_0 \rightarrow \cdots \rightarrow s_i$

no cost, no weight, no time . . . !

CTL with counting constraints

Definition

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

where $P \in \text{AP}$ and C is a constraint of the form:

$$\bigwedge \bigvee \left(\sum_{i=1}^{\ell} \alpha_i \cdot \sharp\varphi_i \sim k \right)$$

with $\varphi_i \in \text{CCTL}$

Semantics of CTL

$\mathcal{S} = \langle Q, q_0, \rightarrow, \ell \rangle$

Semantics

$$
\begin{array}{lll}
q \models P & \text{iff} & P \in \ell(q) \\
q \models \varphi \wedge \psi & \text{iff} & q \models \varphi \text{ and } q \models \psi \\
q \models \neg\varphi & \text{iff} & q \not\models \varphi \\
q \models \mathsf{EX}\varphi & \text{iff} & \exists q \rightarrow q', \text{s.t. } q' \models \varphi \\
q \models \mathsf{E}\varphi\mathsf{U}\psi & \text{iff} & \exists \rho \in \mathsf{Runs}(q), \text{s.t.} \exists k \geq 0, \rho(k) \models \psi, \text{ and} \\
 & & \forall 0 \leq i < k,\ \rho(i) \models \varphi \\
q \models \mathsf{A}\varphi\mathsf{U}\psi & \text{iff} & \forall \rho \in \mathsf{Runs}(q), \exists k \geq 0, \text{s.t. } \rho(k) \models \psi, \text{ and} \\
 & & \forall 0 \leq i < k,\ \rho(i) \models \varphi
\end{array}
$$

CTL with counting constraints

Semantics

$$q \models \mathsf{E}\varphi\mathsf{U}_{[C]}\psi \quad \text{iff} \quad \exists \rho \in \mathsf{Runs}(q),\ \exists k \geq 0, \rho(k) \models \psi,\ \rho_{|k-1} \models C,\ \text{and } \forall 0 \leq i < k,\ \rho(i) \models \varphi$$

$$q \models \mathsf{A}\varphi\mathsf{U}_{[C]}\psi \quad \text{iff} \quad \forall \rho \in \mathsf{Runs}(q),\ \exists k \geq 0,\ \rho(k) \models \psi,\ \rho_{|k-1} \models C,\ \text{and } \forall 0 \leq i < k,\ \rho(i) \models \varphi$$

Semantics of constraints Let π be a finite run.

$\pi \models C$ is based on the interpretation of $\sharp\varphi$ over π: $\pi[\sharp\varphi]$

$$\pi[\sharp\varphi] \stackrel{\text{def}}{=} |\{j \mid 0 \leq j \leq |\pi| \ \wedge\ \pi(j) \models \varphi\}|$$

Abbreviations. . .

- "It is possible to reach Φ and C":

$$\mathsf{EF}_{[C]}\,\Phi \stackrel{\text{def}}{=} \mathsf{E}\top\,\mathsf{U}_{[C]}\,\Phi$$

- "Along every path, one can reach Φ and C":

$$\mathsf{AF}_{[C]}\,\Phi \stackrel{\text{def}}{=} \mathsf{A}\top\,\mathsf{U}_{[C]}\,\Phi$$

Abbreviations...

- "It is possible to reach Φ and C":

$$\mathsf{EF}_{[C]}\,\Phi \stackrel{\text{def}}{=} \mathsf{E}\top\mathsf{U}_{[C]}\,\Phi$$

- "Along every path, one can reach Φ and C":

$$\mathsf{AF}_{[C]}\,\Phi \stackrel{\text{def}}{=} \mathsf{A}\top\mathsf{U}_{[C]}\,\Phi$$

- "Always: C implies Φ":

$$\mathsf{AG}_{[C]}\,\Phi \stackrel{\text{def}}{=} \neg\mathsf{EF}_{[C]}\neg\Phi$$

- "There is a path along which C implies Φ":

$$\mathsf{EG}_{[C]}\,\Phi \stackrel{\text{def}}{=} \neg\mathsf{AF}_{[C]}\neg\Phi$$

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

- $\mathsf{EX}\,\varphi \stackrel{\text{def}}{=} \mathsf{EF}_{[\sharp\top=1]}\,\varphi$
- $\mathsf{E}\varphi\mathsf{U}_{[C]}\psi \stackrel{\text{def}}{=} \mathsf{EF}_{[C\wedge\sharp(\neg\varphi)=0]}\psi$
- we can reach the alarm state in less than 50 transitions:

$$\mathsf{EF}_{[\sharp\top<50]}\texttt{Alarm}$$

Examples of formulas

- $\mathsf{EX}\,\varphi \stackrel{\text{def}}{=} \mathsf{EF}_{[\sharp\top=1]}\,\varphi$
- $\mathsf{E}\varphi\mathsf{U}_{[c]}\psi \stackrel{\text{def}}{=} \mathsf{EF}_{[c\wedge\sharp(\neg\varphi)=0]}\psi$
- we can reach the alarm state in less than 50 transitions:

$$\mathsf{EF}_{[\sharp\top<50]}\mathtt{Alarm}$$

- we can reach the alarm state in less than 50 "ticks":

$$\mathsf{EF}_{[\sharp\mathtt{tick}<50]}\mathtt{Alarm}$$

Timed CTL over KSs with tick: $\mathsf{E}\varphi\mathsf{U}_{<k}\psi \stackrel{\text{def}}{=} \mathsf{E}\varphi\mathsf{U}_{[\sharp\mathtt{tick}<k]}\psi$

Examples of formulas

- For an ATM: “it is not possible to get money when three mistakes are made in the same session”:

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

Examples of formulas

- 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":

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

Examples of formulas

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

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

Examples of formulas

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

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

- Ratio: "there is a path leading to P s.t. the ratio of `error` states is less than 1 percent."

Examples of formulas

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

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

- Ratio: "there is a path leading to P s.t. the ratio of `error` states is less than 1 percent."

Examples of formulas

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

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

- Ratio: "there is a path leading to P s.t. the ratio of `error` states is less than 1 percent."

$$\mathsf{EF}_{[100\cdot\sharp\texttt{error}-\sharp\top<0]}P$$

(Constraints "$\frac{\sharp P}{\sharp P'} \sim k$" can easily be expressed.)

Outline

Expressiveness

Constraints: $\bigwedge \bigvee \Sigma_{i=1}^{l} \alpha_i \cdot \sharp\varphi_i \sim k$

Proposition

CCTL with positive coefficients can be translated into CTL.

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

Expressiveness

Constraints: $\bigwedge \bigvee \Sigma_{i=1}^{l} \alpha_i \cdot \sharp\varphi_i \sim k$

Proposition

CCTL with positive coefficients can be translated into CTL.

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

Proposition

The formula $\varphi = \mathsf{AG}_{[\sharp A - \sharp B < 0]} \bot$ cannot be translated into CTL.

Idea: the set of models of any CTL formula can be recognized by a Büchi alternating tree automaton. For φ, we need a counter for any path.

Diagonal constraints increase the expressive power of CTL.

Succinctness

CCTL formulas with constraints $\bigwedge \bigvee \Sigma_{i=1}^{l} \alpha_i \cdot \sharp \varphi_i \sim k$ with positive coefficients can be translated into CTL, but in these constraints, there are three potential sources of concision:

- the binary encoding of constants,
- the Boolean combinations in the constraints,
- and the sums.

Succinctness

CCTL formulas with constraints $\bigwedge \bigvee \Sigma_{i=1}^{l} \alpha_i \cdot \sharp \varphi_i \sim k$ with positive coefficients can be translated into CTL, but in these constraints, there are three potential sources of concision:

- the binary encoding of constants,
- the Boolean combinations in the constraints,
- and the sums.

Only the two first yield an exponential improvement in succinctness.

Succinctness – binary encoding of the constants

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

Succinctness – binary encoding of the constants

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

Proposition CCTL with constraints $\sharp\varphi \sim k$ can be exponentially more succinct than CTL.

Idea: TCTL formulas $\mathrm{EF}_{<k}A$ and $\mathrm{EF}_{>k}A$ do not admit any equivalent CTL formula of temporal height less than k [LST03].

Succinctness – Boolean combinations

Proposition CCTL with constraints $\bigwedge(\sharp\varphi_i \sim k_i)$ and unary encoding of integers can be exponentially more succinct than CTL.

Succinctness – Boolean combinations

Proposition CCTL with constraints $\bigwedge(\sharp\varphi_i \sim k_i)$ and 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 [Wil99, Al03].

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

(no need of binary encoding of constants)

Succinctness – Sums

Proposition For every formula $\Phi \in$ CCTL with constraints of the form $\sum \sharp\varphi_i \sim k$ and with unary encoding of integers, there exists an equivalent CTL formula of DAG-size polynomial in $|\Phi|$.

Succinctness – Sums

Proposition For every formula $\Phi \in \mathrm{CCTL}$ with constraints of the form $\sum \sharp\varphi_i \sim k$ and with unary encoding of integers, there exists an equivalent CTL formula of DAG-size polynomial in $|\Phi|$.

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

Succinctness – Sums

Proposition For every formula $\Phi \in \mathrm{CCTL}$ with constraints of the form $\sum \sharp\varphi_i \sim k$ and with unary encoding of integers, 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:

$$\begin{aligned}
\Psi_k &\stackrel{\text{def}}{=} \mathsf{E}\Big(\bigwedge_i \bar{P}_i\Big)\mathsf{U}\Big(\bigvee_i P_i \wedge \beta_{k,1,\bot}\Big) && (k > 0)\\
\Psi_0 &\stackrel{\text{def}}{=} \mathsf{E}\Big(\bigwedge_i \bar{P}_i\Big)\mathsf{U}\, A\\
\Psi_{-1} &\stackrel{\text{def}}{=} \bot
\end{aligned}$$

$$\begin{aligned}
\beta_{k,i,\epsilon} &\stackrel{\text{def}}{=} (P_i \wedge \beta_{k-1,i+1,\top}) \vee (\bar{P}_i \wedge \beta_{k,i+1,\epsilon}) && (i < n)\\
\beta_{k,n,\top} &\stackrel{\text{def}}{=} (P_n \wedge \mathsf{EX}\,\Psi_{k-1}) \vee (\bar{P}_n \wedge \mathsf{EX}\,\Psi_k)\\
\beta_{k,n,\top} &\stackrel{\text{def}}{=} P_n \wedge \mathsf{EX}\,\Psi_{k-1}
\end{aligned}$$

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 CTL + F^{-1} is PSPACE-Complete)

Outline

Model checking algorithms

The complexity of the model checking problem depends on the type of the constraints.
Complexity ranges from P-complete to. . . undecidability.

Constraints

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

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

$$\mathcal{C}_1: \quad \Big(\sum_{i=1}^{l} \sharp\varphi_i\Big) \sim k$$

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

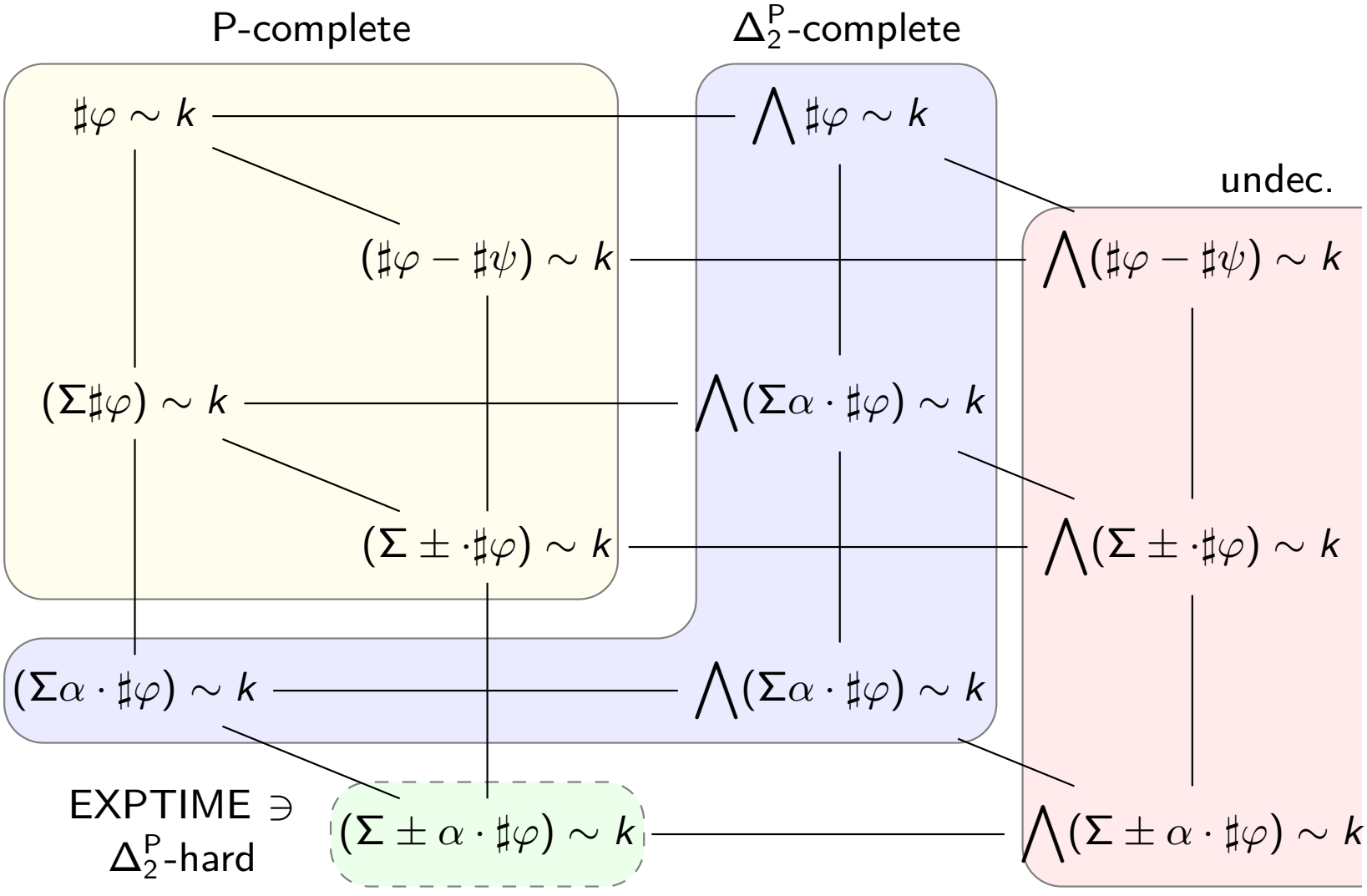
$$\mathcal{C}_2: \quad (\sharp\varphi - \sharp\psi) \sim k'$$

$$\mathcal{C}_3: \quad \Big(\sum_{i=1}^{l} \pm \cdot \sharp\varphi_i\Big) \sim k'$$

$$\alpha\mathcal{C}_3: \quad \Big(\sum_{i=1}^{l} \beta_i \cdot \sharp\varphi_i\Big) \sim k' \qquad \beta_i \in \mathbb{Z}$$

And $\mathcal{B}(\mathcal{C})$ is the set of Boolean combinations of constr. in $\mathcal{C}$.

Conclusion


P-complete
Δ_2^{P}-complete
undec.
$\sharp\varphi \sim k$
$\bigwedge \sharp\varphi \sim k$
$(\sharp\varphi - \sharp\psi) \sim k$
$\bigwedge(\sharp\varphi - \sharp\psi) \sim k$
$(\Sigma\sharp\varphi) \sim k$
$\bigwedge(\Sigma\alpha \cdot \sharp\varphi) \sim k$
$(\Sigma \pm \cdot\sharp\varphi) \sim k$
$\bigwedge(\Sigma \pm \cdot\sharp\varphi) \sim k$
$(\Sigma\alpha \cdot \sharp\varphi) \sim k$
$\bigwedge(\Sigma\alpha \cdot \sharp\varphi) \sim k$
EXPTIME $\ni$
Δ_2^{P}-hard
$(\Sigma \pm \alpha \cdot \sharp\varphi) \sim k$
$\bigwedge(\Sigma \pm \alpha \cdot \sharp\varphi) \sim k$

Outline

Model checking $\mathrm{CCTL}_{\mathcal{C}_1}$: $(\Sigma \sharp \varphi_i) \sim k$

Theorem Model-checking $\mathrm{CCTL}_{\mathcal{C}_1}$ is P-complete.

Reduction to a model-checking problem for TCTL formula over Kripke structures with $0/1$ durations.

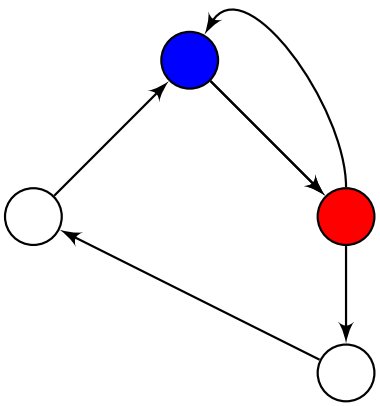
Model checking $\mathrm{CCTL}_{\mathcal{C}_1}$: $(\Sigma \sharp \varphi_i) \sim k$

Theorem Model-checking $\mathrm{CCTL}_{\mathcal{C}_1}$ is P-complete.

Reduction to a model-checking problem for TCTL formula over Kripke structures with 0/1 durations.

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

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



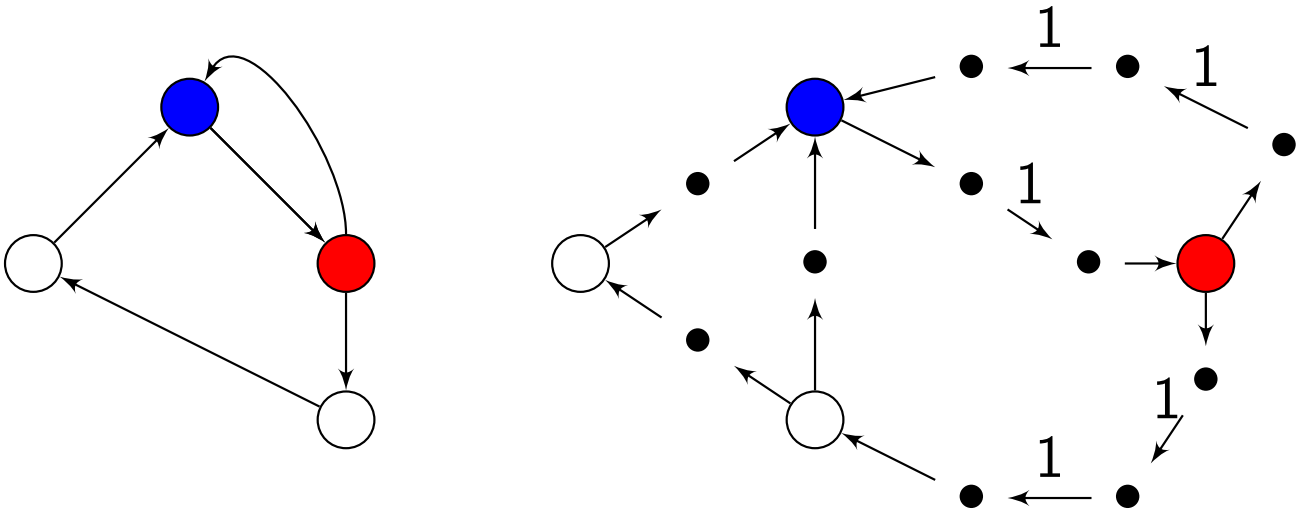
Model checking $\mathrm{CCTL}_{\mathcal{C}_1}$: $(\Sigma \sharp \varphi_i) \sim k$

Theorem Model-checking $\mathrm{CCTL}_{\mathcal{C}_1}$ is P-complete.

Reduction to a model-checking problem for TCTL formula over Kripke structures with 0/1 durations.

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

$\bullet \models P_1$ $\qquad$ $\bullet \models P_1 \wedge P_2$



$\Phi' = \mathsf{E}(\neg \bullet \Rightarrow \varphi)\mathsf{U}_{\sim k}(\neg \bullet \wedge \psi) \in \mathrm{TCTL}$

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

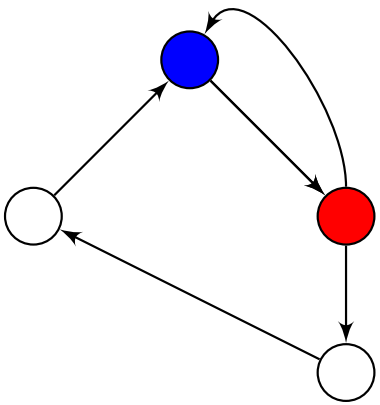
For $\mathrm{CCTL}_{\mathcal{C}_3}$ (*i.e.* with constraint of the form "$(\Sigma_{i=1}^{l} \pm \cdot \sharp\varphi_i) \sim k$'"), the previous encoding gives a DKS with durations in $\{-1, 0, 1\}$:

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

For $\mathrm{CCTL}_{\mathcal{C}_3}$ (*i.e.* with constraint of the form "$(\Sigma_{i=1}^{l} \pm \cdot \sharp\varphi_i) \sim k$'"), the previous encoding gives a DKS with durations in $\{-1, 0, 1\}$:

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

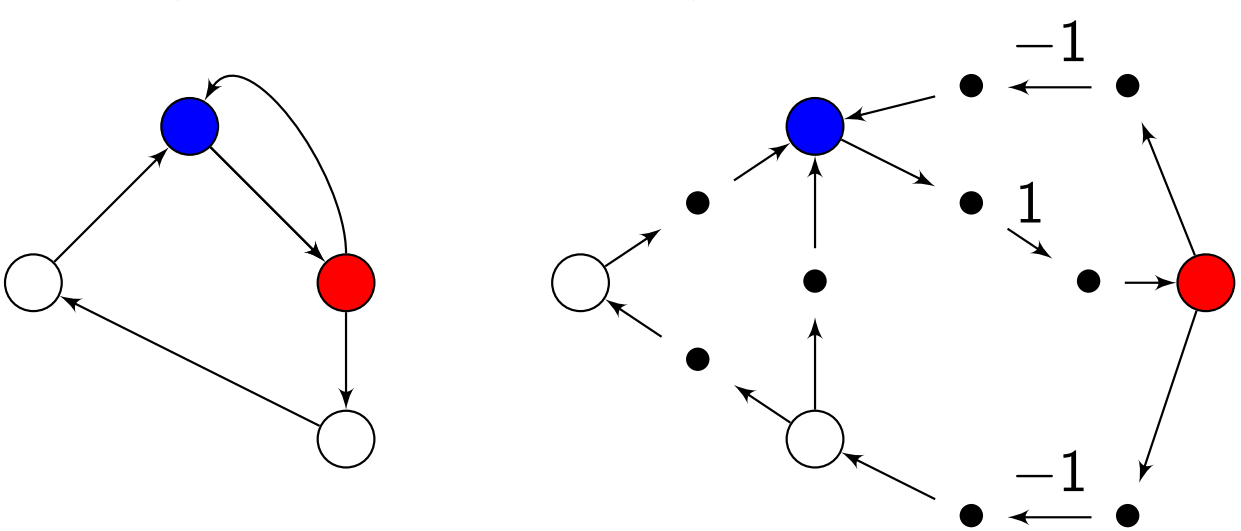
$\models P_1$ $\models P_2$



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

For $\mathrm{CCTL}_{\mathcal{C}_3}$ (*i.e.* with constraint of the form "$(\Sigma_{i=1}^{l} \pm \cdot \sharp\varphi_i) \sim k$'"), the previous encoding gives a DKS with durations in $\{-1, 0, 1\}$:

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



Reduction to a model-checking problem for TCTL over $\mathrm{DKS}^{-1/0/1}$.

Model-checking TCTL over $\mathrm{DKS}^{-1/0/1}$

Theorem

Model-checking TCTL over $\mathrm{DKS}^{-1/0/1}$ is P-complete.

Model-checking TCTL over $\mathrm{DKS}^{-1/0/1}$

Theorem

Model-checking TCTL over $\mathrm{DKS}^{-1/0/1}$ is P-complete.

- $\Phi \stackrel{\text{def}}{=} \mathrm{E}\varphi\mathrm{U}_{\leq k}\psi$: shortest paths + reachability of negative cycles.

Model-checking TCTL over $\mathrm{DKS}^{-1/0/1}$

Theorem

Model-checking TCTL over $\mathrm{DKS}^{-1/0/1}$ is P-complete.

- $\Phi \stackrel{\text{def}}{=} \mathrm{E}\varphi\mathrm{U}_{\leq k}\psi$: shortest paths + reachability of negative cycles.
- $\Phi \stackrel{\text{def}}{=} \mathrm{E}\varphi\mathrm{U}_{=k}\psi$:

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

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

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

Add constraints for $\varphi \ldots \rightsquigarrow R'_k$

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

Model-checking TCTL over $\mathrm{DKS}^{-1/0/1}$

Theorem

Model-checking TCTL over $\mathrm{DKS}^{-1/0/1}$ is P-complete.

- $\Phi \stackrel{\text{def}}{=} \mathrm{E}\varphi\mathrm{U}_{\leq k}\psi$: shortest paths + reachability of negative cycles.
- $\Phi \stackrel{\text{def}}{=} \mathrm{E}\varphi\mathrm{U}_{=k}\psi$:

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

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

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

Add constraints for $\varphi \ldots \rightsquigarrow R'_k$
$\Rightarrow q \models \Phi$ iff $(q, q') \in R'_k$ for some q' satisfying ψ'.

- $\Phi \stackrel{\text{def}}{=} \mathrm{A}\varphi\mathrm{U}_{\sim k}\psi$: ...

Model-checking $\mathrm{CCTL}_{\mathcal{C}_3}$: $(\Sigma \pm \cdot \sharp \varphi_i) \sim k$

Corollary Model-checking $\mathrm{CCTL}_{\mathcal{C}_3}$ is P-complete.

Outline

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_0)}$: $\bigwedge \bigvee \sharp\varphi \sim k$

Theorem

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_0)}$ is Δ_2^{P}-hard.

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_0)}$: $\bigwedge\bigvee\sharp\varphi \sim k$

Theorem

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_0)}$ is Δ_2^{P}-hard.

Reduction from SNSAT:

$$\begin{cases} z_1 \stackrel{\text{def}}{=} \exists X_1.\varphi_1(X_1) \\ z_2 \stackrel{\text{def}}{=} \exists X_2.\varphi_2(z_1, X_2) \\ \qquad\qquad \dots \\ z_n \stackrel{\text{def}}{=} \exists X_n.\varphi_n(z_1, \dots, z_{n-1}, X_n) \end{cases}$$

Question: $z_n = \top$?

Model-checking $\mathrm{CCTL}_{\alpha\mathcal{C}_1}$: $(\Sigma_{i=1}^{l}\alpha_i \cdot \sharp\varphi_i) \sim k,\ \alpha_i \in \mathbb{N}$

Theorem

Model-checking $\mathrm{CCTL}_{\alpha\mathcal{C}_1}$ is Δ_2^{P}-hard.

Model-checking $\mathrm{CCTL}_{\alpha\mathcal{C}_1}$: $(\Sigma_{i=1}^{l}\alpha_i \cdot \sharp\varphi_i) \sim k,\ \alpha_i \in \mathbb{N}$

Theorem

Model-checking $\mathrm{CCTL}_{\alpha\mathcal{C}_1}$ is Δ_2^{P}-hard.

Reduction from the MC pb for TCTL over KSs with durations.

Let $\mathcal{S} = (Q, R_{\mathcal{S}}, \ell)$ be a DKS.

"$q \xrightarrow{k} q'$ in $\mathcal{S}$" is replaced by "$q(\mathtt{ok}) \to q_{aux}(P_k) \to q'(\mathtt{ok})$"

The TCTL formula $\Phi \stackrel{\text{def}}{=} \mathrm{E}\varphi\mathrm{U}_{\sim m}\psi$ is replaced by $\widetilde{\Phi}$:

$$\mathrm{E}(\mathtt{ok} \Rightarrow \widetilde{\varphi})\ \mathrm{U}_{[C]}\ (\mathtt{ok} \wedge \widetilde{\psi})$$

with $C \stackrel{\text{def}}{=} \sum_{d \in W} d \cdot \sharp P_d \sim m$.

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$: $\bigvee\bigwedge(\Sigma_{i=1}^{l}\alpha_i \cdot \sharp\varphi_i) \sim k$, $\alpha_i \in \mathbb{N}$

Theorem

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$ is in Δ_2^{P}.

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$: $\bigvee\bigwedge(\Sigma_{i=1}^{l}\alpha_i \cdot \sharp\varphi_i) \sim k$, $\alpha_i \in \mathbb{N}$

Theorem

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\alpha\mathcal{C}_1)}$ 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;
- verify that it verifies the formula.

And for $\mathrm{EG}_{[C]}\psi$, we look for an infinite run but $(\Sigma\alpha_i \cdot \sharp\varphi_i) \sim k$ may change its truth value at most twice along the run...

Outline

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_2)}$: $\bigwedge \bigvee (\sharp\varphi - \sharp\psi) \sim k$

Theorem

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

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_2)}$: $\bigwedge\bigvee(\sharp\varphi - \sharp\psi) \sim k$

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 $\mathcal{M}$.

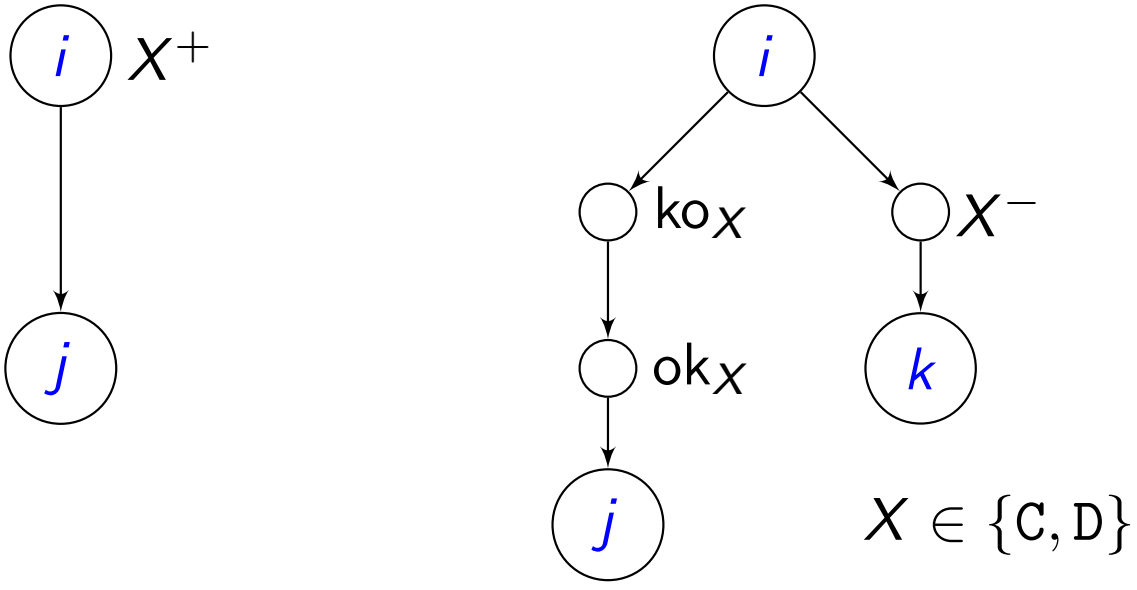
Two counters: `C` and `D`, and 5 kinds of instructions:

- i : `C++, goto` j
- i : `D++, goto` j
- i : `if C=0 then goto` j `else C--, goto` k
- i : `if D=0 then goto` j `else D--, goto` k
- i : `halt`

Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_2)}$: $\bigwedge\bigvee(\sharp\varphi - \sharp\psi) \sim k$

From $\mathcal{M}$, we build the following Kripke structure:

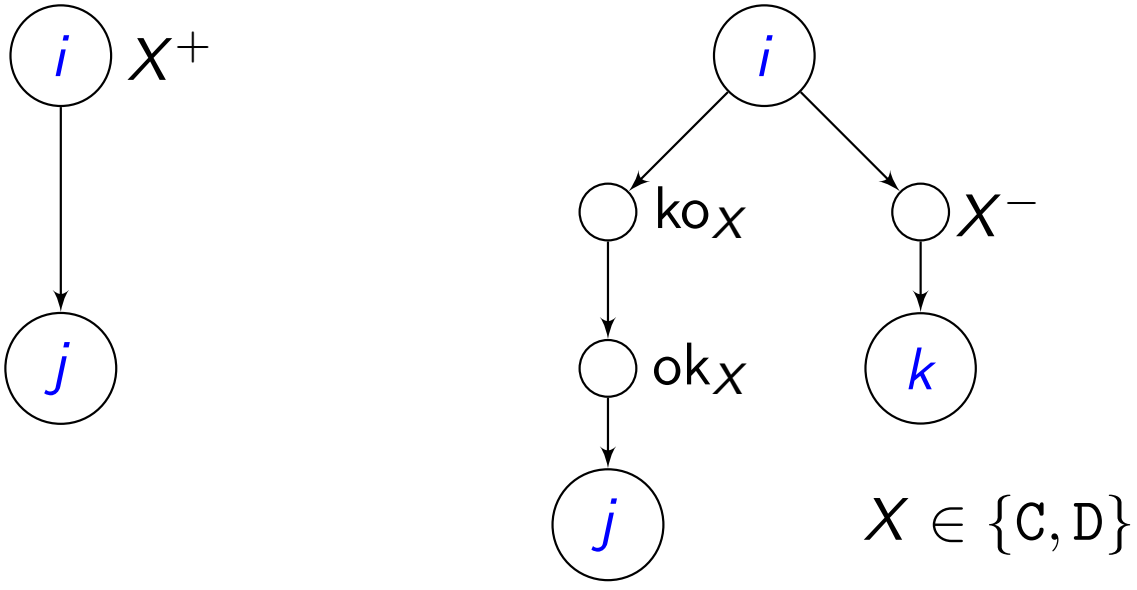
i : `X++, goto` j $\quad$ i : `if X=0 then goto` j `else X--, goto` k



Model-checking $\mathrm{CCTL}_{\mathcal{B}(\mathcal{C}_2)}$: $\bigwedge\bigvee(\sharp\varphi - \sharp\psi) \sim k$

From $\mathcal{M}$, we build the following Kripke structure:

i : `X++, goto` j $\quad$ i : `if X=0 then goto` j `else X--, goto` k

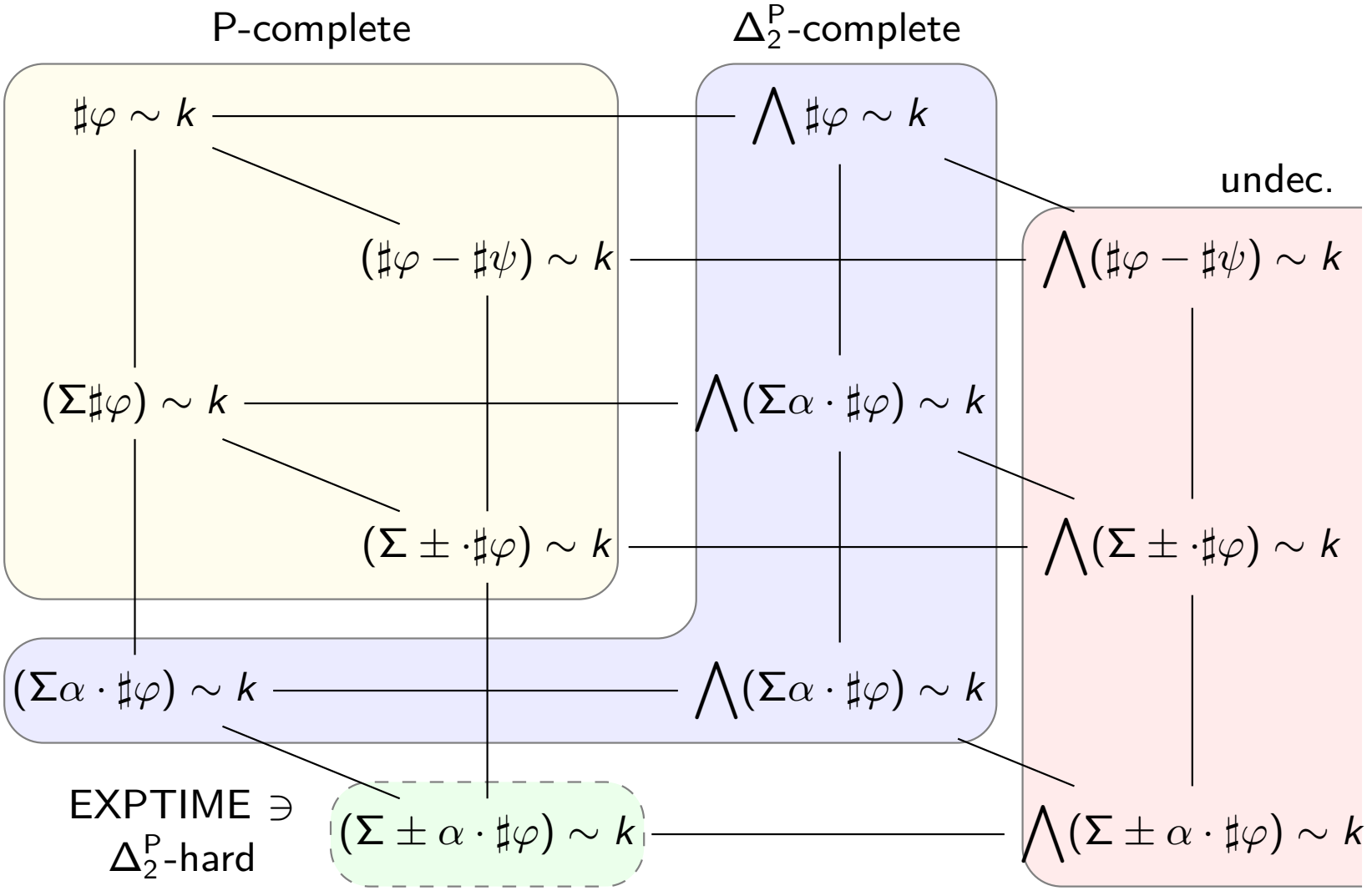


$\mathcal{M}$ does not halt *if and only if* $q_1 \models_{\mathcal{S}_\mathcal{M}} \mathsf{EG}_{[C]}\bot$ with:

$$C \stackrel{\text{def}}{=} (\sharp\mathsf{halt} \geq 1) \ \vee C_{bad}$$

$$C_{bad} \stackrel{\text{def}}{=} \bigvee_{X\in\{\mathtt{C},\mathtt{D}\}} \Big((\sharp X^+ - \sharp X^- < 0) \vee (\sharp X^+ - \sharp X^- > 0 \wedge \sharp \mathsf{ko}_X - \sharp \mathsf{ok}_X > 0)\Big)$$

Conclusion


P-complete
Δ_2^{P}-complete
undec.
$\sharp\varphi \sim k$
$\bigwedge \sharp\varphi \sim k$
$(\sharp\varphi - \sharp\psi) \sim k$
$\bigwedge(\sharp\varphi - \sharp\psi) \sim k$
$(\Sigma\sharp\varphi) \sim k$
$\bigwedge(\Sigma\alpha \cdot \sharp\varphi) \sim k$
$(\Sigma \pm \cdot\sharp\varphi) \sim k$
$\bigwedge(\Sigma \pm \cdot\sharp\varphi) \sim k$
$(\Sigma\alpha \cdot \sharp\varphi) \sim k$
$\bigwedge(\Sigma\alpha \cdot \sharp\varphi) \sim k$
EXPTIME $\ni$
Δ_2^{P}-hard
$(\Sigma \pm \alpha \cdot \sharp\varphi) \sim k$
$\bigwedge(\Sigma \pm \alpha \cdot \sharp\varphi) \sim k$

Outline

LTL + counting constraints

Syntax

$$LTL \ni \varphi, \psi \ ::= \ P \mid \varphi \wedge \psi \mid \neg\varphi \mid \varphi \mathsf{U} \psi \mid \mathsf{X}\varphi$$

where $P \in \mathsf{AP}$.

LTL + counting constraints

Syntax

$$LTL \ni \varphi, \psi \ ::= \ P \mid \varphi \wedge \psi \mid \neg\varphi \mid \varphi \mathsf{U} \psi \mid \mathsf{X}\varphi$$

where $P \in \mathsf{AP}$.

Semantics $\rho : q_1 \to q_2 \to \cdots$

$$\begin{array}{lll}
\rho \models P & \text{iff} & P \in \ell(q_1) \\
\rho \models \varphi \wedge \psi & \text{iff} & \rho \models \varphi \text{ and } \rho \models \psi \\
\rho \models \neg\varphi & \text{iff} & \rho \not\models \varphi \\
\rho \models \mathsf{X}\varphi & \text{iff} & \rho^1 \models \varphi \\
\rho \models \varphi \mathsf{U} \psi & \text{iff} & \exists k \geq 0, \rho^k \models \psi, \text{ and } \forall 0 \leq i < k, \ \rho^i \models \varphi
\end{array}$$

LTL + counting constraints

Syntax

$$LTL \ni \varphi, \psi \ ::= \ P \mid \varphi \wedge \psi \mid \neg\varphi \mid \varphi \mathsf{U}_{[C]} \psi \mid \mathsf{X}\varphi$$

where $P \in \mathsf{AP}$.

Semantics $\rho : q_1 \to q_2 \to \cdots$

$$\begin{array}{lll}
\rho \models P & \text{iff} & P \in \ell(q_1) \\
\rho \models \varphi \wedge \psi & \text{iff} & \rho \models \varphi \text{ and } \rho \models \psi \\
\rho \models \neg\varphi & \text{iff} & \rho \not\models \varphi \\
\rho \models \mathsf{X}\varphi & \text{iff} & \rho^1 \models \varphi \\
\rho \models \varphi \mathsf{U} \psi & \text{iff} & \exists k \geq 0, \rho^k \models \psi, \text{ and } \forall 0 \leq i < k, \ \rho^i \models \varphi \\
\rho \models \varphi \mathsf{U}_{[C]} \psi & \text{iff} & \exists k \geq 0, \rho^k \models \psi, \ \ \rho, k-1 \models C \\
 & & \text{and } \forall 0 \leq i < k, \ \rho^i \models \varphi
\end{array}$$

LTL + counting constraints

Syntax

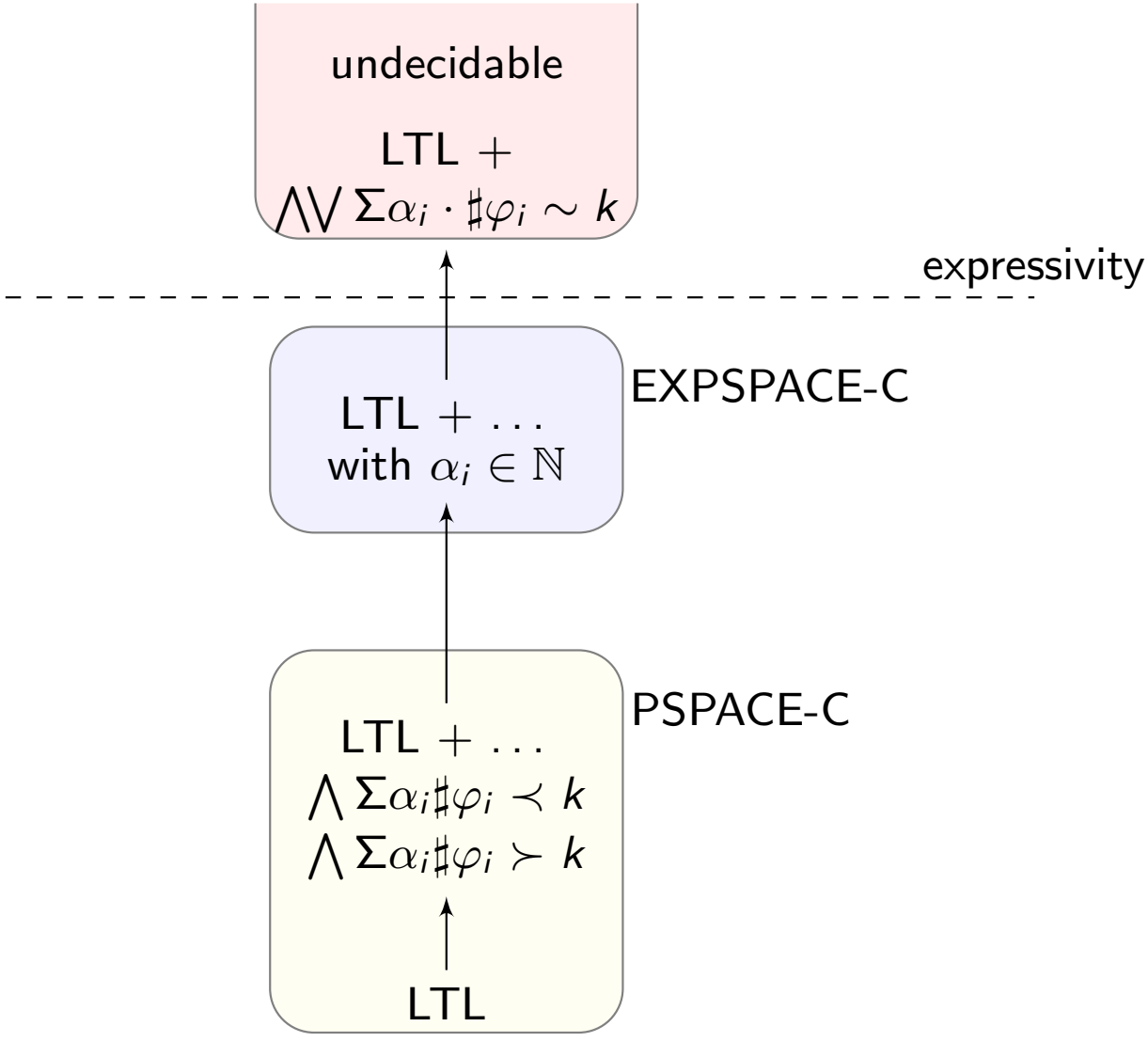
$$LTL \ni \varphi, \psi \ ::= \ P \mid \varphi \wedge \psi \mid \neg\varphi \mid \varphi \mathsf{U}_{[C]} \psi \mid \mathsf{X}\varphi$$

where $P \in \mathsf{AP}$.

Semantics $\rho : q_1 \to q_2 \to \cdots$

$$\begin{array}{lll}
\rho \models P & \text{iff} & P \in \ell(q_1) \\
\rho \models \varphi \wedge \psi & \text{iff} & \rho \models \varphi \text{ and } \rho \models \psi \\
\rho \models \neg\varphi & \text{iff} & \rho \not\models \varphi \\
\rho \models \mathsf{X}\varphi & \text{iff} & \rho^1 \models \varphi \\
\rho \models \varphi \mathsf{U} \psi & \text{iff} & \exists k \geq 0, \rho^k \models \psi, \text{ and } \forall 0 \leq i < k, \ \rho^i \models \varphi \\
 & \text{iff} &
\end{array}$$

Overview of the results for LTL



Linear-time case

CLTL $\stackrel{\text{def}}{=}$ *LTL*+ constraints with "positive" coefficients.
Use standard techniques:

- alternating Büchi automata to recognize the models of a formula.
  (for CLTL, $\mathcal{A}_\Phi$ is exponential in $|\Phi|$.)
- satisfiability and model-checking
- removing equality makes procedures more efficient.
- couting along a unique path is easier...

Outline

Conclusion

- Other semantics are possible. . . For example, the cumulative semantics.
  $\mathrm{EF}^{c}_{[\sharp\top \leq k_1]}(P_1 \wedge \mathrm{EF}^{c}_{[\sharp\top \leq k_2]} P_2)$: "there is a run with less than k_2 transitions leading to P_2 and along this run there is some P_1 located at less than k_1 transitions from the initial state"

Conclusion

- Other semantics are possible. . . For example, the cumulative semantics.
  $\mathrm{EF}^{\mathrm{c}}_{[\sharp\top \leq k_1]}(P_1 \wedge \mathrm{EF}^{\mathrm{c}}_{[\sharp\top \leq k_2]} P_2)$: "there is a run with less than k_2 transitions leading to P_2 and along this run there is some P_1 located at less than k_1 transitions from the initial state"
- Freeze variables can also be used instead of subscripts in Until modalities.

Conclusion

- Other semantics are possible... For example, the cumulative semantics.
  $\mathrm{EF}^{\mathrm{c}}_{[\sharp\top \leq k_1]}(P_1 \wedge \mathrm{EF}^{\mathrm{c}}_{[\sharp\top \leq k_2]} P_2)$: "there is a run with less than k_2 transitions leading to P_2 and along this run there is some P_1 located at less than k_1 transitions from the initial state"
- Freeze variables can also be used instead of subscripts in Until modalities.
- CCTL*...

Conclusion

- Other semantics are possible... For example, the cumulative semantics.
  $\mathrm{EF}^{\mathrm{c}}_{[\sharp\top \leq k_1]}(P_1 \wedge \mathrm{EF}^{\mathrm{c}}_{[\sharp\top \leq k_2]} P_2)$: "there is a run with less than k_2 transitions leading to P_2 and along this run there is some P_1 located at less than k_1 transitions from the initial state"
- Freeze variables can also be used instead of subscripts in Until modalities.
- CCTL*...

- See in practice.

Conclusion

- Other semantics are possible... For example, the cumulative semantics.
  $\mathrm{EF}^{\mathrm{c}}_{[\sharp\top \leq k_1]}(P_1 \wedge \mathrm{EF}^{\mathrm{c}}_{[\sharp\top \leq k_2]} P_2)$: "there is a run with less than k_2 transitions leading to P_2 and along this run there is some P_1 located at less than k_1 transitions from the initial state"
- Freeze variables can also be used instead of subscripts in Until modalities.
- CCTL*...

- See in practice.
- Consider other operators (modulo,...).

Conclusion

- Other semantics are possible... For example, the cumulative semantics.
  $\mathrm{EF}^{c}_{[\sharp\top \leq k_1]}(P_1 \wedge \mathrm{EF}^{c}_{[\sharp\top \leq k_2]} P_2)$: "there is a run with less than k_2 transitions leading to P_2 and along this run there is some P_1 located at less than k_1 transitions from the initial state"
- Freeze variables can also be used instead of subscripts in Until modalities.
- CCTL*...

- See in practice.
- Consider other operators (modulo,...).
- Consider properties over infinite runs (ratio).


M. Adler and N. Immerman.
An $n!$ lower bound on formula size.
ACM Transactions on Computational Logic, 4(3):296–314, 2003.


A. Bouajjani, R. Echahed, and P. Habermehl.
On the verification problem of nonregular properties for nonregular processes.
In *Proc. 10th LICS*, pages 123–133. IEEE Comp. Soc. Press, 1995.


A. Bouajjani, R. Echahed, and P. Habermehl.
Verifying infinite state processes with sequential and parallel composition.
In *Proc. 22nd POPL*, pages 95–106, 1995.


E. A. Emerson and R. J. Trefler.
Generalized quantitative temporal reasoning: An automata-theoretic approach.
In *Proc. 7th TAPSOFT*, volume 1214 of *LNCS*, pages 189–200. Springer, 1997.


E. A. Emerson and R. J. Trefler.
Parametric quantitative temporal reasoning.
In *Proc. 14th LICS*, pages 336–343. IEEE Comp. Soc. Press, 1999.


F. Laroussinie, Ph. Schnoebelen, and M. Turuani.
On the expressivity and complexity of quantitative branching-time temporal logics.
Theor. Comput. Sci., 297(1–3):297–315, 2003.


Property Specification Language Reference Manual, Version 1.1, 2003.
`http://www.eda-stds.org/vfv/docs/PSL-v1.1.pdf`.


T. Wilke.
CTL$^+$ is exponentially more succinct than CTL.
In *Proc. 19th FSTTCS*, volume 1738 of *LNCS*, pages 110–121. Springer, 1999.


J. Yang, A. K. Mok, and F. Wang.
Symbolic model checking for event-driven real-time systems.

ACM Transactions on Programming Languages and Systems, 19(2):386–412, 1997.