Decidable Problems for Counter Systems

Day 2
Linear-Time Temporal Logics

Stéphane Demri

demri@lsv.ens-cachan.fr

LSV, ENS Cachan, CNRS, INRIA

ESSLLI 2010, Copenhagen, August 2010

Plan of the talk

- Yesterday's lecture:
  - Classes of counter systems (Minsky machines, CA, VASS, VAS, relational CA, etc.) and standard decision problems.
  - Presburger arithmetic and semilinear sets.
- Standard LTL
- Logic $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ for counter systems
- Presburger LTL
- LTL with registers

Specifying existence of runs in temporal logic

- Repeated reachability can be obviously expressed by $\mathsf{G}\ \mathsf{F}\ q_f$.

- Initialized VASS $(\mathcal{V}, (q, \vec{z}))$ is unbounded iff there is a run $(q, \vec{z}) \xrightarrow{*} (q', \vec{y}) \xrightarrow{*} (q', \vec{y'})$ with $\vec{y} \prec \vec{y'}$ for some q'.

- In temporal logic lingua:

$$(\mathcal{V}, (q, \vec{z})) \models \mathsf{E}\ \exists\, \mathsf{y}_1, \ldots, \mathsf{y}_n\ \mathsf{F}(\bigwedge_{i=1}^{n} \mathsf{x}_i = \mathsf{y}_i \wedge \mathsf{F}(\bigwedge_{i=1}^{n} \mathsf{x}_i \geq \mathsf{y}_i \wedge \bigvee_{i=1}^{n} \mathsf{x}_i > \mathsf{y}_i))$$

- Linear-time temporal logics offer genericity and fragments can be easily designed.

Temporal Modalities in Computations

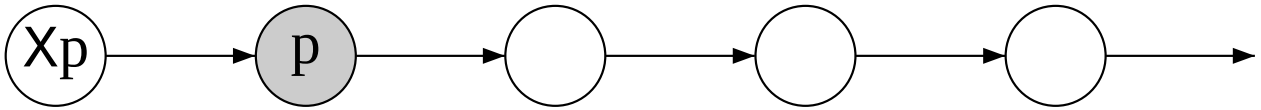
Modalities

- Temporal logics contain modalities with a temporal interpretation.
- A modality is a syntactic object (term) that modifies the relationships between a predicate and a subject.
- In the sentence “Tomorrow, it will rain”, the term “Tomorrow” is a temporal modality.
- The temporal modalities allow one to speak about the sequencing of configurations along a run.
- Standard temporal modalities:
  - `X` (“neXt”),
  - `F` (“sometimes”),
  - `G` (“always”).
- We also use the Boolean connectives: $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction) and $\Rightarrow$ (material implication).

“next” and “sometimes”

- $\mathsf{X}\varphi$ states that the next state satisfies φ.

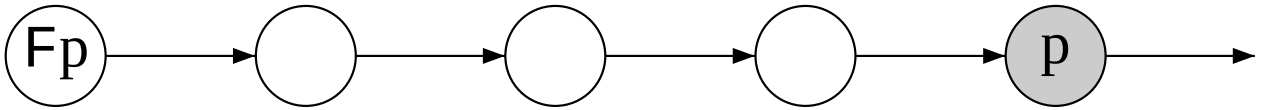
  $\mathsf{X}p$: next-time p



  For example, $\varphi \vee \mathsf{X}\varphi$ states that φ is satisfied now or in the next state.

- $\mathsf{F}p$ announces that a future state satisfies φ without specifying which state, and $\mathsf{G}\varphi$ that all the future states satisfy φ.

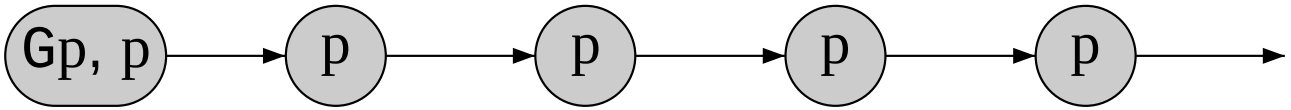
  $\mathsf{F}p$: sometimes p



“always” and “until”

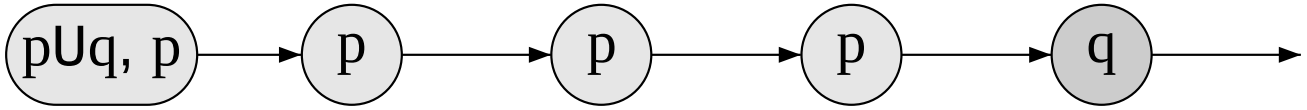
- The operator G is the dual of F: whatever the formula φ may be, if φ is always satisfied, then it is not true that $\neg\varphi$ will some day be satisfied, and conversely.
  ($\mathsf{G}\varphi$ and $\neg\mathsf{F}\neg\varphi$ are equivalent.)

  Gp: always p



- The U operator is richer and more complicated than the combinator F. $\varphi_1 \mathsf{U} \varphi_2$ states that φ_1 is true until φ_2 is true.

  pUq: p until q



$\mathsf{G}(\mathtt{alert} \Rightarrow \mathsf{F}\ \mathtt{halt})$ can be refined with

$$\mathsf{G}(\mathtt{alert} \Rightarrow (\mathtt{alarm}\ \mathsf{U}\ \mathtt{halt})).$$

Abbreviations

- The $\mathtt{F}$ combinator is a special case of $\mathtt{U}$: $\mathtt{F}\varphi$ and $\mathtt{true}\ \mathtt{U}\ \varphi$ are equivalent.
- Weak until operator $\mathtt{W}$.
- $\varphi_1 \mathtt{W} \varphi_2$ expresses "$\varphi_1 \mathtt{U} \varphi_2$", but without the inevitable occurrence of φ_2 and if φ_2 never occurs, then φ_1 remains true forever.
- $\varphi_1 \mathtt{W} \varphi_2$ is equivalent to $\mathtt{G}\varphi_1 \vee (\varphi_1 \mathtt{U} \varphi_2)$.

Plain LTL

LTL syntax

- LTL formulae:

$$\varphi, \psi ::= \mathrm{p} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \psi$$

- Atomic formulae are propositional variables.

- Later, control states or arithmetical constraints about counter values are considered at the atomic level.

- LTL modelsρ are ω-sequences of propositional valuations of the form $\rho : \mathbb{N} \to \mathcal{P}(\mathrm{PROP})$.

Satisfaction relation (formal semantics)

- $\rho, i \models \mathrm{p} \overset{\text{def}}{\Leftrightarrow} \mathrm{p} \in \rho(i)$,
- $\rho, i \models \neg\varphi \overset{\text{def}}{\Leftrightarrow} \rho, i \not\models \varphi$,
- $\rho, i \models \varphi_1 \wedge \varphi_2 \overset{\text{def}}{\Leftrightarrow} \rho, i \models \varphi_1$ and $\rho, i \models \varphi_2$,
- $\rho, i \models \mathsf{X}\varphi \overset{\text{def}}{\Leftrightarrow} \rho, i+1 \models \varphi$,
- $\rho, i \models \varphi_1 \mathsf{U} \varphi_2 \overset{\text{def}}{\Leftrightarrow}$ there is $j \geq i$ such that $\rho, j \models \varphi_2$ and $\rho, k \models \varphi_1$ for all $i \leq k < j$.

$\mathsf{F}\varphi \overset{\text{def}}{=} \top \mathsf{U} \varphi$, $\mathsf{G}\varphi \overset{\text{def}}{=} \neg \mathsf{F} \neg \varphi$, $\varphi \Rightarrow \psi \overset{\text{def}}{=} \neg\varphi \vee \psi$, etc.

About LTL

- Models(φ): set of models ρ such that $\rho, 0 \models \varphi$.
- Models can be viewed as ω-words over the alphabet $\mathcal{P}(\mathrm{PROP})$.
- Models(φ) can be effectively represented by a Büchi automaton $\mathcal{A}_\varphi$.
- Satisfiability and model-checking (see later) are PSPACE-complete problems [Sistla & Clarke, JACM 85].

A Brief Introduction to Büchi Automata

Automata-based approach

- Automata-based approach: to reduce logical problems into automata-based decision problems in order to take advantage of known results from automata theory.

- Another view: a means to transform declarative statements (formulae) into operational devices (automata).

- Standard target problems on automata:
  - the nonemptiness problem checks whether an automaton admits at least one accepting computation,
  - the universality problem checking whether an automaton accepts everything,
  - the inclusion problem checking whether the language accepted by automaton $\mathcal{A}$ is included in the language accepted by automaton $\mathcal{B}$.

Desirable properties

- The reduction should be conceptually simple.
- The computational complexity of the automata-based target problem should be well-characterized.
- Preferrably, the reduction might allow to obtain the optimal complexity for the source logical problem.
- A pioneering work by R. Büchi [Buchi, 62] in which Büchi automata are shown equivalent to formulae in monadic second-order logics (MSO).

Definition for Büchi automata

- Büchi automaton: finite-state automaton except that ω-words are accepted instead of finite words.

- The set of final states is used as an acceptance condition for ω-words.

- Büchi automaton $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$:

  - Σ is a finite alphabet,

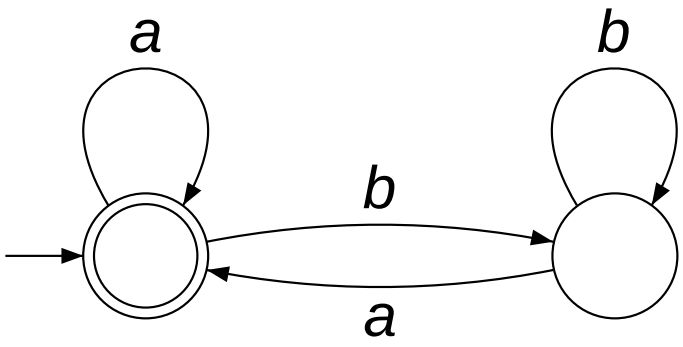
  - Q is a finite set of states,

  - $Q_0 \subseteq Q$ is the set of *initial* states,

  - the transition relation δ is a subset of $Q \times \Sigma \times Q$,

  - $F \subseteq Q$ is a set of final states.

Recognizing infinitely many a's



- Run $\rho = q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \ldots$: for $i \geq 0$, $(q_i, a_i, q_{i+1}) \in \delta$.
- Run ρ is successful: $q_0 \in Q_0$ and some state of F is repeated infinitely often in ρ.
- Label of ρ: $\sigma = a_0 a_1 \cdots \in \Sigma^\omega$.
- $\mathcal{A}$ accepts $\mathrm{L}(\mathcal{A})$ made of ω-words $\sigma \in \Sigma^\omega$ such that there exists a successful run of $\mathcal{A}$ with label σ.

Generalized Büchi automata

- Standard generalization: conjunctions of classical Büchi conditions.

- Generalized Büchi automaton (GBA)
  $\mathcal{A} = (\Sigma, Q, Q_0, \delta, \{F_1, \ldots, F_k\})$ with $F_1, \ldots, F_k \subseteq Q$.

- Successful run ρ of $\mathcal{A}$: the first state is initial and there is a state in F_i that is repeated infinitely often in ρ for $1 \leq i \leq n$.

- Let $\mathcal{A} = (\Sigma, Q, Q_0, \delta, \{F_1, \ldots, F_k\})$ be a GBA.
  One can compute (in logarithmic space in the size of $\mathcal{A}$) a Büchi automaton $\mathcal{A}^b = (\Sigma, Q^b, Q_0^b, \delta^b, F^b)$ such that $\mathrm{L}(\mathcal{A}^b) = \mathrm{L}(\mathcal{A})$.

Proof

- GBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, \{F_1, \ldots, F_k\})$.
- $\mathcal{A}^b$ is made of k copies of $\mathcal{A}$ and simulates the generalized accepting condition by passing from one copy to another.
- $Q^b \stackrel{\text{def}}{=} Q \times \{1, \ldots k\}$; $Q_0^b \stackrel{\text{def}}{=} Q_0 \times \{1\}$; $F^b \stackrel{\text{def}}{=} F_1 \times \{1\}$.
- $\delta^b((q, i), a)$ is the union of the following sets:
  1. $\{(q', i) : q \xrightarrow{a} q' \in \delta,\ q \notin F_i\}$ (stay in the same copy if no final state in F_i is reached),
  2. $\{(q', (i \ mod \ k) + 1) : q \xrightarrow{a} q' \in \delta,\ q \in F_i\}$ (go to the next copy if a final state in F_i is reached).
- $\mathcal{A}$ and $\mathcal{A}^b$ accepts the same language.

Properties about Büchi automata

- The class of languages accepted by Büchi automata corresponds to the class of ω-regular languages.

- The family of ω-regular languages is closed by intersection, union and complementation.

- Nonemptiness problem for Büchi automata:

  Input: a Büchi automaton $\mathcal{A}$,
  Question: is $\mathrm{L}(\mathcal{A}) \neq \emptyset$?

- The nonemptiness problem for Büchi automata is NLogSpace-complete.
  (and can therefore be solved in polynomial time.)

- By contrast, the universality problem for Büchi automata is PSpace-complete [Sistla & Vardi & Wolper, TCS 87].

Recapitulation

- LTL formulae:

$$\varphi, \psi ::= \mathrm{p} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \psi$$

- Models: ω-sequences over $\mathcal{P}(\mathrm{PROP})$.

- Models(φ): the set of sequences ρ in $\mathcal{P}(\mathrm{PROP})^\omega$ such that $\rho, 0 \models \varphi$.

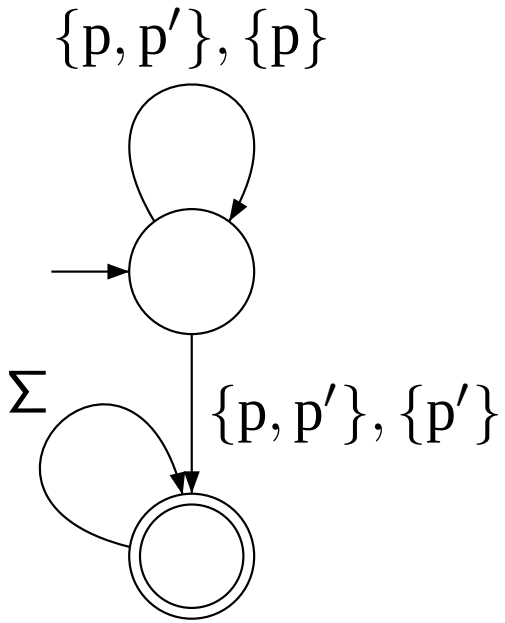
- Nonemptiness for GBA can be reduced (in logarithmic space) to nonemptiness for BA.

- The nonemptiness problem for Büchi automata can be solved in polynomial-time.

From LTL Formulae to Automata

LTL captured by Büchi automata

- $\mathrm{L}(\mathcal{A}) =$ Models(p U p').



- There is a Büchi automaton $\mathcal{A}_\varphi$ s.t.
  1. $\mathrm{L}(\mathcal{A}_\varphi) =$ Models(φ); $|\mathcal{A}_\varphi|$ is in $2^{\mathcal{O}(|\varphi|)}$,
  2. $\mathcal{A}_\varphi$ can be effectively computed in polynomial space in $|\varphi|$.

  [Vardi & Wolper, IC 94]

- Below, we briefly recall how $\mathcal{A}_\varphi$ is defined from φ.

Closure set

- Closure set contains all the formulae we need to consider to check satisfiability.
- Closure $\mathrm{cl}(\varphi)$ of φ: smallest set
  - containing the subformulae of φ,
  - closed under negation (we identify $\neg\neg\psi$ with ψ),
  - if $\chi_1 \mathsf{U} \chi_2 \in \mathrm{cl}(\varphi)$, then $\mathsf{X}(\chi_1 \mathsf{U} \chi_2) \in \mathrm{cl}(\varphi)$.
- Cardinal of $\mathrm{cl}(\varphi)$ is linear in the size of φ.

Atom

- An atom is nothing but a maximally consistent subset of $\mathrm{cl}(\varphi)$.
- Atom X: subset of $\mathrm{cl}(\varphi)$ s.t.
  1. for all formulae ψ in $\mathrm{cl}(\varphi)$, $\psi \in X$ iff $\neg\psi \notin X$,
  2. $\psi_1 \wedge \psi_2 \in X$ iff $\psi_1, \psi_2 \in X$,
  3. $\psi_1 \vee \psi_2 \in X$ iff $\psi_1 \in X$ or $\psi_2 \in X$,

One-step consistent pairs

- (Y, Y') is one-step consistent:
  - if $\psi_1 \mathsf{U} \psi_2 \in Y$, then $\psi_2 \in Y$ or ($\psi_1 \in Y$ and $\psi_1 \mathsf{U} \psi_2 \in Y'$). ($\psi_1 \mathsf{U} \psi_2$ is equivalent to $\psi_2 \vee (\psi_1 \wedge \mathsf{X}\psi_1 \mathsf{U} \psi_2)$.)
  - for $\mathsf{X}\psi \in \mathrm{cl}(\varphi)$, $\mathsf{X}\psi \in Y$ iff $\psi \in Y'$.
- The states of $\mathcal{A}_\varphi$ are atoms and the transition relation contains one-step consistent pairs.
- Each state X of $\mathcal{A}_\varphi$ is a set of formulae that are intended to be satisfied from the current position.
- Either this satisfaction can be checked locally or the transition relation of $\mathcal{A}_\varphi$ allows us to propagate the constraints.

Defining $\mathcal{A}_\varphi$

- GBA $\mathcal{A}_\varphi = (\Sigma, Q, Q_0, \delta, F_1, \ldots, F_\alpha)$:
  - $\Sigma = \mathcal{P}(\{\mathrm{p}_1, \ldots, \mathrm{p}_N\})$,
  - Q is the set of atoms,
  - Q_0 is the subset of atoms containing φ,
  - $X \xrightarrow{a} Y \in \delta$ iff $a = \{\mathrm{p}_1, \ldots, \mathrm{p}_N\} \cap X$ and (X, Y) is one-step consistent,
  - for $\psi_1 \mathsf{U} \psi_2 \in \mathrm{cl}(\varphi)$, there is exactly one set F_i such that $F_i = \{X \in Q : \text{either } \psi_1 \mathsf{U} \psi_2 \notin X \text{ or } \psi_2 \in X\}$.
- Accepting conditions $F_1, \ldots, F_\alpha$ guarantee that the search for witnesses is not delayed forever.

Main properties for $\mathcal{A}_\varphi$

- $\mathrm{L}(\mathcal{A}_\varphi) = \text{Models}(\varphi)$.

- Given a model $\rho : \mathbb{N} \to \mathcal{P}(\{\mathrm{p}_1, \ldots, \mathrm{p}_N\}) \in \text{Models}(\varphi)$, there is a unique accepting run $X_0 \xrightarrow{\rho(0)} X_1 \xrightarrow{\rho(1)} X_2 \xrightarrow{\rho(2)} \cdots$ in $\mathcal{A}_\varphi$ s.t. ($\star$) for $i \geq 0$ and $\psi \in \mathrm{cl}(\varphi)$, we have
$$\psi \in X_i \text{ iff } \rho, i \models \psi.$$

- Conversely, for each accepting run $X_0 \xrightarrow{a_0} X_1 \xrightarrow{a_1} X_2 \xrightarrow{a_2} \cdots$ in $\mathcal{A}_\varphi$, the model ρ defined by $\rho(i) = a_i$ for $i \geq 0$ satisfies ($\star$).

- In the sequel, we write $\mathcal{A}_\varphi$ to denote the Büchi automaton recognizing Models(φ).

Full Presburger LTL for Counter Systems

New ingredients

- Models of the form $(q_0, \vec{x_0}), (q_1, \vec{x_1}), \ldots$.
- Control states q as atomic formulae.
- Arithmetical constraints about counter values ($\mathrm{x}_1 > \mathrm{x}_2$).
- Comparing values at successive positions, e.g. $\mathrm{x}_1 > \mathbf{X}\mathrm{x}_2$.
- First-order quantification over counter values, e.g.

  $\exists\, \mathrm{y}\ \mathtt{G}(\mathrm{x}_1 \leq \mathrm{y}). \approx$ "Along the run, counter 1 is bounded."

$\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ syntax

- Queen logic $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ (fragments are defined from it).
- Giving up the standard abstraction: propositional variables understood as properties about the current configuration.
- $\mathrm{VAR}^{\mathrm{p}} = \{\mathsf{y}_1, \mathsf{y}_2, \ldots\}$: set of integer variables.
- $\mathrm{VAR} = \{\mathsf{x}_1, \mathsf{x}_2, \ldots\}$: set of counter variables.
- $\mathsf{Q} = \{q_1, q_2, \ldots\}$: set of control state symbols.
- $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ formulae:

$$\varphi ::= \psi \mid q \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \exists\, \mathsf{y}\, \varphi$$

  - ψ is a Presburger formula with free variables included in $\mathrm{VAR}^{\mathrm{p}} \cup \mathrm{VAR}$,
  - $q \in \mathsf{Q}$.

Satisfaction relation

- Model ρ of dimension n: element of $(Q \times \mathbb{N}^n)^\omega$ with finite $Q \subseteq \mathsf{Q}$.

- Environment $\mathcal{E}$: partial map $\mathrm{VAR}^{\mathrm{p}} \to \mathbb{N}$.

- $\rho, i \models_{\mathcal{E}} q \overset{\text{def}}{\Leftrightarrow} q = q_i$.

- $\rho, i \models_{\mathcal{E}} \psi \overset{\text{def}}{\Leftrightarrow} \mathbf{v}_i \models \psi$ in PrA
  with $\mathbf{v}_i$ extends $\mathcal{E}$ s.t. $\mathbf{v}_i(\mathsf{x}_j) = \vec{x_i}(j)$ ($j \in [1, n]$),
  assuming ψ is a Presburger formula with free variables in $\mathrm{VAR}^{\mathrm{p}} \cup \{\mathsf{x}_1, \ldots, \mathsf{x}_n\}$.

- $\rho, i \models_{\mathcal{E}} \mathsf{X}\varphi \overset{\text{def}}{\Leftrightarrow} \rho, i+1 \models_{\mathcal{E}} \varphi$.

- $\rho, i \models_{\mathcal{E}} \exists\, \mathsf{y}\, \varphi$ iff there is $k \in \mathbb{N}$ such that $\rho, i \models_{\mathcal{E}[\mathsf{y} \mapsto k]} \varphi$.

Decision problems for $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$

- Semi-closed formula: no variable from $\mathrm{VAR}^{\mathrm{p}}$ is free. $\mathtt{F}(\mathsf{x}_1 = \mathsf{y})$ is not semi-closed unlike $\mathtt{G}(\mathsf{x}_1 > \mathsf{x}_2)$ and $\exists\, \mathsf{y}\ \mathtt{G}(\mathsf{x}_1 \leq \mathsf{y})$.

- SATISFIABILITY PROBLEM

  Input: An $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ semi-closed formula φ with free counter variables $\mathsf{x}_1, \ldots, \mathsf{x}_n$.

  Question: Is there a model $\rho \in (\mathtt{Q} \times \mathbb{N}^n)$ s.t. $\rho, 0 \models_\emptyset \varphi$?

- EXISTENTIAL MODEL-CHECKING PROBLEM

  Input: CS $\mathcal{S} = (Q, n, \delta)$, $(q_0, \vec{x}_0)$ and semi-closed formula φ with free variables in $\{\mathsf{x}_1, \ldots, \mathsf{x}_n\}$.

  Question: Is there an infinite run ρ starting at $(q_0, \vec{x}_0)$ such that $\rho, 0 \models_\emptyset \varphi$?

(Infinite runs of CS are $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ models)

A simple reduction

- The control state repeated reachability problem can be reduced to the model-checking problem for $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$.

- Let $\mathcal{S}$, $(q, \vec{x})$ and q_f be an instance.

- Equivalence:
  - there is an infinite run from $(q, \vec{x})$ such that q_f is repeated infinitely often,
  - there is an infinite run ρ from $(q, \vec{x})$ such that $\rho, 0 \models \mathtt{G}\ \mathtt{F}\ q_f$.

- This can be extended to the sequence $F_1, \ldots, F_N$ understood conjunctively and each F_i disjunctively.

$$\bigwedge_{i=1}^{N} \Big(\bigvee_{q' \in F_i} \mathtt{G}\ \mathtt{F}\ q' \Big)$$

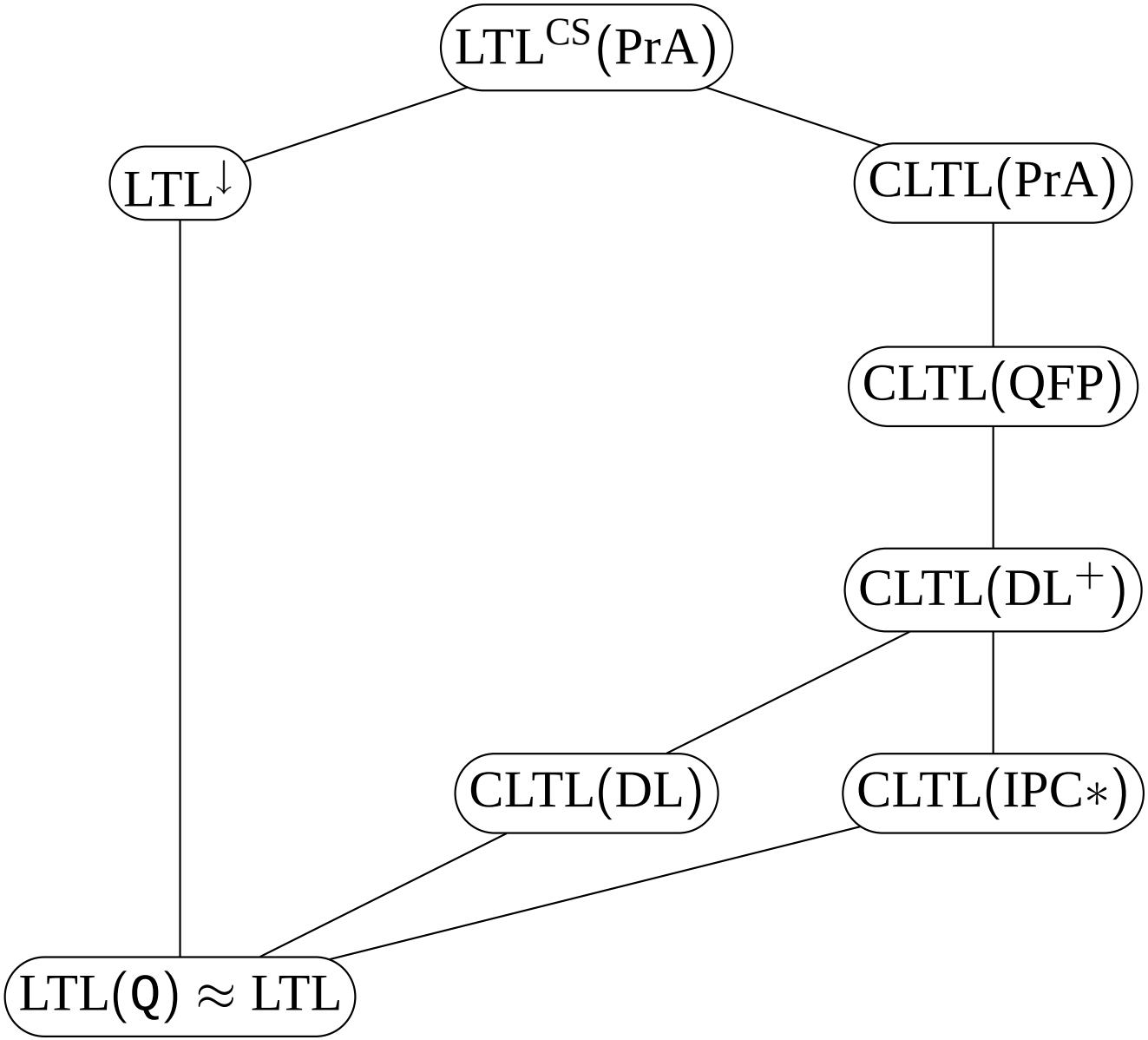
Temporal logics with Presburger constraints

- Constraints on the number of event occurrences.
  [Bouajjani et al., LICS'95; Laroussinie et al., TIME'10]
- Constraints on XML documents.
  [Dal Zilio & Lugiez, RTA'03; Seidl et al, ICALP'04]
- LTL with first-order variables for log auditing.
  [Roger & Goubault-Larrecq, CSFW'01]
- Fairness logic for VASS [Jančar, TCS 90].
- and many many others . . .

$\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ is sometimes too expressive !

- Model-checking restricted to $\mathrm{LTL}(\mathtt{Q})$ is already undecidable ...
- ... but $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ restricted to formulae in which temporal operators are not in the scope of first-order quantification has a decidable satisfiability problem.
- Let us restrict first-order quantification on $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ to regain decidability for satisfiability.
- Two types of restrictions:
  - First-order quantification only to speak about successive counter values.
  - First-order quantification only to store a counter value and possibly perform equality tests later.

Fragments


$\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$
$\mathrm{LTL}^{\downarrow}$
$\mathrm{CLTL}(\mathrm{PrA})$
$\mathrm{CLTL}(\mathrm{QFP})$
$\mathrm{CLTL}(\mathrm{DL}^{+})$
$\mathrm{CLTL}(\mathrm{DL})$
$\mathrm{CLTL}(\mathrm{IPC}*)$
$\mathrm{LTL}(\mathtt{Q}) \approx \mathrm{LTL}$

Presburger LTL $\mathrm{CLTL}(\mathrm{PrA})$

A macro for comparing successive counter values

- From PrA formula $\psi(z_1, \ldots, z_k)$,

$$\psi(\mathbf{X}^{i_1} x_{j_1}, \ldots, \mathbf{X}^{i_k} x_{j_k}) \stackrel{\text{def}}{=}$$

$$(\exists\, y_1, \ldots, y_k\ \mathsf{X}^{i_1}(y_1 = x_{j_1}) \wedge \cdots \wedge \mathsf{X}^{i_k}(y_k = x_{j_k}) \wedge \psi(y_1, \ldots, y_k)$$

- For instance, $x_1 = \mathbf{X} x_2$ is obtained from $z_1 = z_2$.
- $\mathsf{G}(x_1 = \mathbf{X} x_1)$ states that first counter has constant value.
- $\mathrm{CLTL}(\mathrm{PrA})$: fragment of $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ by allowing quantification only in $\psi(\mathbf{X}^{i_1} x_{j_1}, \ldots, \mathbf{X}^{i_k} x_{j_k})$.
- $\mathbf{X}$ and X are of distinct nature but both refer to the next position.

The fragment $\mathrm{CLTL(PrA)}$

- Terms and formulae:

$$t ::= 0 \mid 1 \mid \mathbf{X}^i \mathsf{x} \mid t + t$$

$$\varphi ::= t \equiv_k t \mid t < t \mid q \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi.$$

  with $\mathsf{x} \in \mathrm{VAR}$, $k > 1$ and $q \in \mathtt{Q}$.

- NB: no integer variables, quantifier-free Presburger formulae.

- $\mathtt{G}(\mathsf{x}_1 < \mathbf{X}\mathsf{x}_1)$ is satisfiable; $\mathtt{G}(\mathsf{x}_1 > \mathbf{X}\mathsf{x}_1)$ is not.

- $\sigma, i \models \psi(\mathbf{X}^{l_1}\mathsf{x}_1, \ldots, \mathbf{X}^{l_n}\mathsf{x}_n)$ iff $(\sigma_1(i + l_1)(\mathsf{x}_1), \ldots, \sigma_1(i + l_n)(\mathsf{x}_n)) \in \mathrm{REL}(\psi)$ with
  - $\sigma = (\sigma_1, \sigma_2)$,
  - $\sigma_1 : \mathbb{N} \to (\mathrm{VAR} \to \mathbb{N})$,
  - $\sigma_2 : \mathbb{N} \to Q$ for some finite subset $Q \subseteq \mathtt{Q}$.

- Model-checking and satisfiability problems as subproblems for $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$.

X-length

- **X**-length of φ: maximal l such that $\mathbf{X}^l \mathrm{x}$ occurs in φ.
- $\mathrm{CLTL}_n^l(\mathrm{PrA})$: restriction of $\mathrm{CLTL}(\mathrm{PrA})$ to formulae with at most n variables and **X**-length less or equal to l.
- There is a logspace reduction from the satisfiability problem for $\mathrm{CLTL}(\mathrm{PrA})$ to its restriction with **X**-length 1.
  - Expressions $\mathrm{x}_1, \ldots, \mathbf{X}^3\mathrm{x}_1$ are encoded by

$$\mathtt{G}(\mathrm{x}'' = \mathbf{X}\mathrm{x}' \wedge \mathrm{x}' = \mathbf{X}\mathrm{x} \wedge \mathrm{x} = \mathbf{X}\mathrm{x}_1)$$

  - $\mathbf{X}\mathrm{x}_1$ [resp. $\mathbf{X}^2\mathrm{x}_1$, $\mathbf{X}^3\mathrm{x}_1$] is replaced by x [resp. x', x''].
  - Take the conjunction.

Two fragments of PrA

- Difference logic DL:

$$\varphi ::= \mathsf{x} \sim \mathsf{y} + d \;\mid\; \mathsf{x} \sim d \;\mid\; \varphi \wedge \varphi \;\mid\; \neg\varphi$$

  with $d \in \mathbb{Z}$, $\sim \in \{<, >, =\}$.

- Quantifier-free Presburger arithmetic QFP:

$$\varphi ::= \sum_{i \in I} a_i \mathsf{x}_i \sim d \;\mid\; \sum_{i \in I} a_i \mathsf{x}_i \equiv_k c \;\mid\; \varphi \wedge \varphi \;\mid\; \neg\varphi$$

  ($c \in \mathbb{N}$, $k \geq 1$, a_i's in $\mathbb{Z}$).

- Satisfiability problem for QFP is NP-complete.

- $\mathrm{CLTL}(L)$: fragment of $\mathrm{CLTL}(\mathrm{PrA})$ to atomic formulae of the form $\psi(\mathbf{X}^{i_1}\mathsf{x}_{j_1}, \ldots, \mathbf{X}^{i_k}\mathsf{x}_{j_k})$ with $\psi \in L$.

A bunch of complexity results

- DL-counter system: counter system such that the Presburger formulae labelling the transitions are in DL.

	MC (DL)	SAT	MC (CA)
$\mathrm{CLTL}_3^1(\mathrm{DL})$	U	U	U
$\mathrm{CLTL}_2^\omega(\mathrm{DL})$	U	U	U
$\mathrm{CLTL}_1^2(\mathrm{DL})$	U	U	PSPACE-c
$\mathrm{CLTL}_2^1(\mathrm{DL})$	U	U	U
$\mathrm{CLTL}_1^1(\mathrm{DL}\text{ or }\mathrm{DL}^+)$	PSPACE-c.	PSPACE-c.	PSPACE-c
$\mathrm{CLTL}_1^1(\mathrm{QFP})$	U	U	PSPACE-c
$\mathrm{CLTL}_1^\omega(\mathrm{QFP})$	U	U	PSPACE-c.

Satisfiability problem for $\mathrm{CLTL}_3^1(\mathrm{DL})$ is undecidable

- Simple reduction from halting problem for Minsky machines.
- Formulae in $\mathrm{CLTL}_3^1(\mathrm{DL})$ can easily internalize the instructions of Minsky machines.
- $\mathcal{S}$ has n instructions, the nth one halts.
- x_1 and x_2 encode counter values, x_3 encodes the instruction ordinal.
- $\mathcal{S}$ does not halt iff formula below is satisfiable:

$$\Big(\bigwedge_{i \in [1,n]} \psi_i\Big) \wedge x_1 = 0 \wedge x_2 = 0 \wedge x_3 = 1$$

Internalizing instructions from Minsky machines

- For "i: increment counter j and goto i'"

$$\psi_i \stackrel{\text{def}}{=} \mathtt{G}(\mathrm{x}_3 = i \Rightarrow (\mathbf{X}\mathrm{x}_3 = i') \wedge (\mathrm{x}_j + 1 = \mathbf{X}\mathrm{x}_j) \wedge (\mathrm{x}_{3-j} = \mathbf{X}\mathrm{x}_{3-j})).$$

- For "i: if counter j equals zero then goto i' else (decrement counter j; goto i'')":

$$\mathtt{G}(\mathrm{x}_3 = i \Rightarrow ((\mathrm{x}_j = 0 \Rightarrow (\mathbf{X}\mathrm{x}_3 = i') \wedge (\mathrm{x}_j = \mathbf{X}\mathrm{x}_j) \wedge (\mathrm{x}_{3-j} = \mathbf{X}\mathrm{x}_{3-j})) \wedge$$

$$(\mathrm{x}_j \neq 0 \Rightarrow (\mathbf{X}\mathrm{x}_3 = i'') \wedge (\mathrm{x}_j - 1 = \mathbf{X}\mathrm{x}_j) \wedge (\mathrm{x}_{3-j} = \mathbf{X}\mathrm{x}_{3-j}))))$$

- Satisfiability problem for $\mathrm{CLTL}_2^1(\mathrm{DL})$ is undecidable too.

LTL with Registers (a.k.a. Freeze LTL)

Registers

- A register can store a counter value and equality tests are performed between registers and counters.

- $\downarrow_r^j \varphi \stackrel{\text{def}}{=} \exists\, y_r\ (y_r = x_j \wedge \varphi)$; $\uparrow_r^j \stackrel{\text{def}}{=} y_r = x_j$.

- $\mathtt{G}(\downarrow_1^1 \mathtt{XG}\neg \uparrow_1^1)$: first counter has distinct values at distinct positions.

- Generalized disjunction

$$\bigvee_{i \geq 1} x_1 = \mathbf{X}^i x_2$$

can be encoded by

$$\downarrow_1^1 \mathrm{XF} \uparrow_1^2$$

Freeze operator

- Freeze quantifier in hybrid logics.
  [Goranko 94; Blackburn & Seligman, JOLLI 95]
  $\downarrow_{\mathsf{x}} \varphi$: φ holds true in the variant model where x is true only at the current state/world.

- Freeze quantifier in real-time logics.
  [Alur & Henzinger, JACM 94]
  $\mathsf{x} \cdot \varphi(\mathsf{x})$ binds the variable x to the current time t.

- Predicate λ-abstraction [Fitting, JLC 02].
  $\langle \mathsf{x} \cdot \mathsf{F}p(\mathsf{x}) \rangle(c)$: current value of constant c satisfies the predicate p.

In LTL with registers, FO quantification is restricted to $\downarrow$ and $\uparrow$.

LTL with registers

- For $n \geq 1$, $\mathrm{LTL}^{\downarrow}[n]$ formulae:

$$\varphi \quad ::= \quad q \mid \uparrow_r^j \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \mathsf{U} \varphi \mid \mathsf{X}\varphi \mid \downarrow_r^j \varphi$$

  where $q \in \mathsf{Q}$, $j \in \{1, \ldots, n\}$ and $r \in \mathbb{N}^+$.

- Register valuation f: finite partial map from $\mathbb{N}^+$ to $\mathbb{N}$.

- Models: element of $(Q \times \mathbb{N}^n)^\omega$ with finite $Q \subseteq \mathsf{Q}$.

- Satisfiability relation:

$$\begin{aligned} \rho, i \models_f \uparrow_r^j \quad &\overset{\text{def}}{\Leftrightarrow} \quad r \in \mathrm{dom}(f) \text{ and } f(r) = \vec{x_i}(j) \\ \rho, i \models_f \downarrow_r^j \varphi \quad &\overset{\text{def}}{\Leftrightarrow} \quad \rho, i \models_{f[r \mapsto \vec{x_i}(j)]} \varphi \end{aligned}$$

Helpful properties

- Nonce property:

$$\mathtt{G}(\downarrow_1^1 \mathtt{XG}\neg \uparrow_1^1)$$

- There is a value of counter 1 such that infinitely often counter 2 takes that value iff infinitely often counter 3 takes that value:

$$\mathtt{F} \downarrow_1^1 (\mathtt{GF} \uparrow_1^2 \Leftrightarrow \mathtt{GF} \uparrow_1^3)$$

- Value of counter 1 is repeated on counter 2 but not on counter 3:

$$\downarrow_1^1 (\mathtt{F} \uparrow_1^2 \wedge \mathtt{G}\neg \uparrow_1^3)$$

Complexity of satisfiability problems

- Infinitary satisfiability problem for $\mathrm{LTL}^{\downarrow}[1]$ restricted to X and F and to a single register is undecidable.
- Finitary satisfiability problem for $\mathrm{LTL}^{\downarrow}[1]$ restricted to a single register is decidable but nonprimitive recursive.

  [Demri & Lazić, TOCL 09].

- Finitary satisfiability problem for $\mathrm{LTL}^{\downarrow}[1]$ restricted to a single register and to F is nonprimitive recursive too.
- Finitary satisfiability problem for $\mathrm{LTL}^{\downarrow}[1]$ restricted to F and to two registers is undecidable.

  [Figueira & Segoufin, MFCS'09]

- Nonprimitive recursiveness uses [Schnoebelen, IPL 02].

Gainy counter automata

- Gainy counter automata: standard counter automaton (Q, n, δ) s.t. for $q \in Q$ and $i \in [1, n]$, $q \xrightarrow{\mathrm{inc}(i)} q \in \delta$
- Alternative one-step relation: $(q, \vec{x}) \xrightarrow{t}_g (q', \vec{x'})$ iff there are $\vec{y}, \vec{y'}$ in $\mathbb{N}^n$ s.t.

$$\vec{x} \preceq \vec{y} \quad (q, \vec{y}) \xrightarrow{t} (q', \vec{y'}) \text{ (exact step)} \quad \vec{y'} \preceq \vec{x'}$$

- Gains can occur in a lazy way: decrement on zero has no effect.

Benefits from Gainy CA

- Control state reachability problem is decidable.
- Control state repeated reachability problem is undecidable (see Day 5).
- Reduction of the latter problem to satisfiability problem for $\mathrm{LTL}^{\downarrow}[1]$ restricted to X and F and to a single register.

Simulating Gainy CA

- Gainy CA $\mathcal{S}$ with initial configuration $(q_0, \vec{0})$.
- For $t \in \delta$, $\Sigma(t)$ denotes the instruction labelling it in $\Sigma = \{\mathrm{inc}(i), \mathrm{dec}(i), \mathrm{zero}(i) : i \in [1, n]\}$.
- Let us build φ in $\mathrm{LTL}^{\downarrow}[1]$ s.t. φ is satisfiable iff $(\mathcal{S}, (q_0, \vec{0}))$ has an infinite run with q_f occurring infinitely often.
- φ is satisfiable only in models in which each position is labelled by a transition and by a value in $\mathbb{N}$.
- Each increment or decrement is associated to a unique value.
- Infinite models of φ are of the form

$$\begin{array}{cccc} t_0 & t_1 & t_2 & t_3 \\ d_0 & d_1 & d_2 & d_3 \end{array} \quad \ldots$$

with $t_i \in \delta$ (finite alphabet) and $d_i \in \mathbb{N}$ (infinite domain).

Simulating Gainy CA (II)

- Let us explain how the run from $(q_0, \vec{0})$ below is encoded.

$$(q_0, \vec{x_0}) \xrightarrow{a_0} (q_1, \vec{x_1}) \xrightarrow{a_1} \cdots \xrightarrow{a_{K-1}} (q_K, \vec{x_K}) \cdots$$

- Projection of the model over δ is

$$t_0 t_1 t_2 \cdots = q_0 \xrightarrow{a_0} q_1, q_1 \xrightarrow{a_1} q_2, \cdots$$

  and q_f is repeated inifinitely often.

- Initial state is q_0:

$$\bigvee_{t = q_0 \xrightarrow{a} q} t$$

- The sequence of transitions respects $\mathcal{S}$:

$$\mathtt{G}\Big(\bigwedge_{t = q \xrightarrow{a} q' \in \delta} \Big(t \Rightarrow \mathtt{X} \bigvee_{t' = q' \xrightarrow{a} q''} t' \Big) \Big)$$

Simulating Gainy CA (III)

- Control state q_f is visited infinitely often:

$$\mathtt{GF} \bigvee_{t = q \xrightarrow{a} q_f} t$$

- Below, $\downarrow$ [resp. $\uparrow$] denotes $\downarrow_1^1$ [resp. $\uparrow_1^1$].
- For $a \in \Sigma$, a is also used as a shortcut for

$$\bigvee_{t = q \xrightarrow{b} q' \in \delta,\ a = b} t$$

- For $i, j \in [1, n]$, there are no two positions for increments having the same value:

$$\mathtt{G}(\mathrm{inc}(i) \Rightarrow \neg(\downarrow \mathtt{XF}(\uparrow \wedge \mathrm{inc}(j))))$$

- For $i, j \in [1, n]$, there are no two positions for decrements having the same value:

$$\mathtt{G}(\mathrm{dec}(i) \Rightarrow \neg(\downarrow \mathtt{XF}(\uparrow \wedge \mathrm{dec}(j))))$$

Simulating Gainy CA (IV)

- For each zero-test on i at position K, each increment on i before position K has a corresponding decrement on i before K.

- The two next conditions are formulated in such a way to avoid using the until operator U.

- $I \sim K$: data value at position I is equal to data value at position K.

- For $i \in [1, n]$ and $J > I$, if $\Sigma(t_I) = \mathrm{inc}(i)$ and $\Sigma(t_J) = \mathrm{zero}(i)$, then there is no $K > J$ such that $\Sigma(t_K) = \mathrm{dec}(i)$ and $I \sim K$:

$$\mathsf{G}(\mathrm{inc}(i) \Rightarrow \downarrow \neg(\mathsf{F}(\mathrm{zero}(i) \wedge (\mathsf{F}(\uparrow \wedge \mathrm{dec}(i))))))$$

Simulating Gainy CA (V)

- For $i \in [1, n]$ and $J > I$, if $\Sigma(t_I) = \mathrm{inc}(i)$ and $\Sigma(t_J) = \mathrm{zero}(i)$, then there is no $K > J$ such that $\Sigma(t_K) = \mathrm{dec}(i)$ and $I \sim K$:

$$\mathsf{G}(\mathrm{inc}(i) \Rightarrow\downarrow \neg(\mathsf{F}(\mathrm{zero}(i) \wedge (\mathsf{F}(\uparrow \wedge \mathrm{dec}(i))))))$$

- For $i \in [1, n]$, if there are $J > I$ such that $\Sigma(t_I) = \mathrm{inc}(i)$ and $\Sigma(t_J) = \mathrm{zero}(i)$, then there is $K > I$ such that $\Sigma(t_K) = \mathrm{dec}(i)$ and $I \sim K$.

$$\mathsf{G}((\mathrm{inc}(i) \wedge \mathsf{F}\ \mathrm{zero}(i)) \Rightarrow\downarrow (\mathsf{F}(\mathrm{dec}(i) \wedge \uparrow)))$$

- φ is satisfiable iff $(\mathcal{S}, (q_0, \vec{0}))$ has an infinite run such that q_f occurs infinitely often.

Related formalisms

- Register automata
  - Register automata [Kaminski & Francez, TCS 94]
  - Data automata [Bouyer & Petit & Thérien, IC 03]
  - See the survey [Segoufin, CSL 06]
- First-order languages [Bojańczyk et al., LICS 06]
- Temporal logics
  - Real-time logic TPTL [Alur & Henzinger, JACM 94]
  - LTL with freeze [D. & Lazić & Nowak, TIME 05]
- Many other formalisms
  - Rewriting systems with data [Bouajjani et al., FCT 07]
  - Hybrid logics [Schwentick & Weber, STACS 07]
  - ...

Conclusion

- Today's lecture:
  - Standard LTL
  - Logic $\mathrm{LTL}^{\mathrm{CS}}(\mathrm{PrA})$ for counter systems
  - Presburger LTL
  - LTL with registers
- Tomorrow's lecture:
  - VASS and FO2 over data words
  - Relationships between VASS, VAS and Petri nets
  - Coverability graphs in a nutshell
  - Covering problem in EXPSPACE