What is in the course?

Analysis of Counter Systems
From Reachability to Temporal Logics

• Part I
  • Lecture 1: Introduction to counter systems
  • Lecture 2: Classes of semilinear reachability sets
  • Lecture 3: Vector addition systems with states

• Part II
  • Lecture 4: Linear-time temporal logics
  • Lecture 5: Model-checking counter systems
  • Lecture 6: Data logics and counter systems

but also Presburger arithmetic, undecidability, computational complexity, etc.
Plan of this talk

- Recalling a few things about counter systems.
- Standard LTL
- Logic $LTL^{CS}(PrA)$ for counter systems
- Presburger LTL
Course material

- Main material can be found in ESSLLI 2010 lecture notes:
  
  www.lsv.ens-cachan.fr/~demri/esslli10-course.html

- Slides shall be available on the dedicated web page:
  
  www.lsv.ens-cachan.fr/~demri/UBAUNC10-course.php
Reminder on counter systems
Formal verification

- Digital systems are everywhere. Desktops, embedded systems, cellular phones, etc.

- Needs for verifying functional/security properties:
  - Hardware components
  - Software (programs, communication protocols, web applications, ...)

*Formal verification is a process in which mathematical techniques are used to guarantee the correctness of a design with respect to some specified behavior.*

[Halpern et al., BSL 01]
Towards counter systems

- Counter system: finite-state automaton with counters interpreted by non-negative integers.

- Techniques for model-checking infinite-state systems are required for formal verification.

- Many applications:
  - Broadcast protocols, Petri nets, . . .
  - Programs with pointer variables. [Bouajjani et al., CAV’06]
  - Replicated finite-state programs. [Kaiser & Kroening & Wahl, CAV’10]
  - Relationships with data logics. [Bojańczyk et al., LICS’06]
  - . . .

- But, counter systems can simulate Turing machines.

- Checking safety or liveness properties for counter systems are undecidable problems.
Presburger arithmetic

• Terms: \( t ::= 0 \mid 1 \mid x \mid t + t. \)

• Presburger formulae (\( k \geq 2 \))

\[
\varphi ::= t \equiv_k t \mid t < t \mid \neg \varphi \mid \varphi \land \varphi \mid \exists x \varphi \mid \forall x \varphi
\]

• Valuation \( v : \text{VAR} \to \mathbb{N} \) + extension to all terms with

\[
v(0) = 0 \quad v(1) = 1 \quad v(t + t') = v(t) + v(t')
\]

• Formula \( \varphi(x_1, \ldots, x_n) \) with \( n \) free variables:

\[
\text{REL}(\varphi(x_1, \ldots, x_n)) \overset{\text{def}}{=} \{(v(x_1), \ldots, v(x_n)) \in \mathbb{N}^n : v \models \varphi\}.
\]

• Sets of tuples definable in Presburger arithmetic are precisely semilinear sets.
Counter systems

- Counter system \( S = (Q, n, \delta) \) of dimension \( n \):
  - \( Q \) is a nonempty finite set of control states.
  - \( n \geq 1 \) is the dimension.
  - \( \delta \) is the transition relation: finite set of transitions of the form
    \[ t = (q, \varphi, q') \]
    where \( q, q' \in Q \) and \( \varphi \) is a Presburger formula with free variables \( x_1, \ldots, x_n, x'_1, \ldots, x'_n \).

- Configuration \( (q, \vec{a}) \in Q \times \mathbb{N}^n \).
  - \( (q, \vec{a}) \xrightarrow{t} (q', \vec{a}') \overset{\text{def}}{=} v[\vec{x} \leftarrow \vec{a}, \vec{x}' \leftarrow \vec{a'}] \models \varphi \).

- Runs as nonempty (possibly infinite) sequences
  \[ \rho = (q_0, \vec{a}_0) \rightarrow (q_1, \vec{a}_1) \cdots (q_k, \vec{a}_k) \cdots \]
Decision problems

- **Reachability Problem:**
  - **Input:** counter system $S$, $(q, \vec{x})$ and $(q', \vec{x}')$.
  - **Question:** $(q, \vec{x}) \rightarrow^* (q', \vec{x}')$?

- **Control State Reachability Problem:**
  - **Input:** counter system $S$, $(q, \vec{x})$ and $q'$.
  - **Question:** $\exists \vec{x}' (q, \vec{x}) \rightarrow^* (q', \vec{x}')$?

- **Control State Repeated Reachability Problem:**
  - **Input:** counter system $S$, $(q, \vec{x})$ and $q_f$.
  - **Question:** is there an infinite run with initial configuration $(q, \vec{x})$ such that the control state $q_f$ is repeated infinitely often?

- **Boundedness Problem:**
  - **Input:** counter system $S$ and $(q, \vec{x})$.
  - **Question:** is $\{(q', \vec{x}') \in Q \times \mathbb{N}^n : (q, \vec{x}) \rightarrow^* (q', \vec{x}')\}$ finite?
Subclasses of counter systems

- Standard counter automaton \((Q, n, \delta)\): transitions are of the form either \(q \overset{\text{inc}(i)}{\longrightarrow} q'\) or \(q \overset{\text{dec}(i)}{\longrightarrow} q'\) or \(q \overset{\text{zero}(i)}{\longrightarrow} q'\).

- Succinct counter automaton \((Q, n, \delta)\): transitions of the form either \(q \overset{\text{inc}(\vec{b})}{\longrightarrow} q'\) with \(\vec{b} \in \mathbb{Z}^n\) or \(q \overset{\text{zero}(\vec{b}')}{\longrightarrow} q'\) with \(\vec{b}' \in \{0, 1\}^n\).

- Vector addition systems with states (VASS): succinct counter automata without zero-tests.
Specifying existence of runs in temporal logic

- Repeated reachability can be obviously expressed by \( G \ F \ q_f \).

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

- In temporal logic lingua:

\[
(\mathcal{V}, (q, \vec{z})) \models E \exists y_1, \ldots, y_n F (\bigwedge_{i=1}^{n} x_i = y_i \land F (\bigwedge_{i=1}^{n} x_i \geq y_i \land \bigvee_{i=1}^{n} x_i > y_i))
\]

- 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), \( \lor \) (disjunction), \( \land \) (conjunction) and \( \Rightarrow \) (material implication).
"next" and "sometimes"

- $X\varphi$ states that the next state satisfies $\varphi$.
  
  $Xp$: next-time $p$

  ![Diagram](image)

  For example, $\varphi \lor X\varphi$ states that $\varphi$ is satisfied now or in the next state.

- $Fp$ announces that a future state satisfies $\varphi$ without specifying which state, and $G\varphi$ that all the future states satisfy $\varphi$.

  $Fp$: sometimes $p$

  ![Diagram](image)
“always” and “until”

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

$$Gp: \text{always } p$$

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

$$p U q: p \text{ until } q$$

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

$$G(\text{alert } \Rightarrow (\text{alarm } U \text{ halt})).$$
• The $F$ combinator is a special case of $U$: $F\varphi$ and $true \cup \varphi$ are equivalent.

• Weak until operator $\overline{w}$.

• $\varphi_1 \overline{w} \varphi_2$ expresses “$\varphi_1 \cup \varphi_2$”, but without the inevitable occurrence of $\varphi_2$ and if $\varphi_2$ never occurs, then $\varphi_1$ remains true forever.

• $\varphi_1 \overline{w} \varphi_2$ is equivalent to $G \varphi_1 \lor (\varphi_1 \cup \varphi_2)$. 
Plain LTL
LTL syntax

• LTL formulae:

\[ \varphi, \psi ::= p \mid \neg \varphi \mid \varphi \land \psi \mid \varphi \lor \psi \mid X \varphi \mid \varphi U \psi \]

• Atomic formulae are propositional variables.

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

• LTL models \( \rho \) are \( \omega \)-sequences of propositional valuations of the form \( \rho : \mathbb{N} \rightarrow \mathcal{P}(\text{PROP}) \).
Satisfaction relation  
(formal semantics)

- \( \rho, i \models p \overset{\text{def}}{\iff} p \in \rho(i) \),

- \( \rho, i \models \neg \varphi \overset{\text{def}}{\iff} \rho, i \not\models \varphi \),

- \( \rho, i \models \varphi_1 \land \varphi_2 \overset{\text{def}}{\iff} \rho, i \models \varphi_1 \) and \( \rho, i \models \varphi_2 \),

- \( \rho, i \models \text{x}\varphi \overset{\text{def}}{\iff} \rho, i + 1 \models \varphi \),

- \( \rho, i \models \varphi_1 \cup \varphi_2 \overset{\text{def}}{\iff} \) there is \( j \geq i \) such that \( \rho, j \models \varphi_2 \) and \( \rho, k \models \varphi_1 \) for all \( i \leq k < j \).

\( F \varphi \overset{\text{def}}{=} \top \cup \varphi \), \( G \varphi \overset{\text{def}}{=} \neg F \neg \varphi \), \( \varphi \Rightarrow \psi \overset{\text{def}}{=} \neg \varphi \lor \psi \), etc.
About LTL

- **Models(φ):** set of models $\rho$ such that $\rho, 0 \models \varphi$.

- Models can be viewed as $\omega$-words over the alphabet $\mathcal{P}(\text{PROP})$.

- Models$(\varphi)$ can be effectively represented by a Büchi automaton $A_\varphi$.

- 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.

- Preferably, 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 $\omega$-words are accepted instead of finite words.

- The set of final states is used as an acceptance condition for $\omega$-words.

- Büchi automaton $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$:
  - $\Sigma$ is a finite alphabet,
  - $Q$ is a finite set of states,
  - $Q_0 \subseteq Q$ is the set of initial states,
  - the transition relation $\delta$ 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 \( \rho \) is successful: \( q_0 \in Q_0 \) and some state of \( F \) is repeated infinitely often in \( \rho \).

- Label of \( \rho \): \( \sigma = a_0a_1\ldots \in \Sigma^\omega \).

- \( \mathcal{A} \) accepts \( L(\mathcal{A}) \) made of \( \omega \)-words \( \sigma \in \Sigma^\omega \) such that there exists a successful run of \( \mathcal{A} \) with label \( \sigma \).
Generalized Büchi automata

- Standard generalization: conjunctions of classical Büchi conditions.

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

- Successful run \( \rho \) of \( A \): the first state is initial and there is a state in \( F_i \) that is repeated infinitely often in \( \rho \) for \( 1 \leq i \leq n \).

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

- \( Q^b \overset{\text{def}}{=} Q \times \{1, \ldots, k\}; \quad Q^b_0 \overset{\text{def}}{=} Q_0 \times \{1\}; \quad F^b \overset{\text{def}}{=} F_1 \times \{1\} \).

- \( \delta^b((q, i), a) \) 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 $\omega$-regular languages.

- The family of $\omega$-regular languages is closed by intersection, union and complementation.

- Nonemptiness problem for Büchi automata:
  - **Input:** a Büchi automaton $A$,
  - **Question:** is $L(A) \neq \emptyset$?

- The nonemptiness problem for Büchi automata is \text{NLOGSPACE}-complete.
  (and can therefore be solved in polynomial time.)

- By contrast, the universality problem for Büchi automata is \text{PSPACE}-complete [Sistla & Vardi & Wolper, TCS 87].
About determinism

- $A$ is said to be deterministic iff $\text{card}(Q_0) = 1$ and for all $q \in Q$ and $a \in \Sigma$, $\text{card}(\delta(Q, a)) \leq 1$.

- It is well-known that every regular language can be accepted by a deterministic finite-state automaton.

- By contrast, $(a + b)^* \cdot a^\omega$ is not accepted by any deterministic Büchi automata.

- Suppose there is a deterministic $A = (\{a, b\}, Q, \{q_0\}, \delta, F)$ such that $L(A) = (a + b)^* \cdot a^\omega$.

- Extension of $\delta$ to a partial function $\delta : Q \times \{a, b\}^* \rightarrow Q$.

- $\sigma_0 = a^\omega \in L(A)$ and there is prefix $u_0$ of $\sigma_0$ such that $\delta(q_0, u_0) \in F$. 
Sketch of the proof (II)

• \( \sigma_1 = u_0 \cdot b \cdot a^\omega \in L(\mathcal{A}) \) and there is prefix \( u_0 \cdot b \cdot u_1 \) of \( \sigma_1 \) such that \( \delta(q_0, u_0 \cdot b \cdot u_1) \in F \).

• By generalization, there are finite words \( u_i \) such that \( \delta(q_0, u_0 \cdot b \cdot u_1 \cdot b \ldots b \cdot u_i) \in F \).

• Finiteness of \( Q \) implies there are \( 0 \leq i < j \) such that \( \delta(s_0, u_0 \cdot b \cdot u_1 \cdot b \ldots b \cdot u_i) = \delta(s_0, u_0 \cdot b \cdot u_1 \cdot b \ldots b \cdot u_i \cdot b \ldots b \cdot u_j) \).

• Consequently,

\[
u_0 \cdot b \cdot u_1 \cdot b \ldots b \cdot u_i \cdot (b \ldots b \cdot u_j)^\omega\]

is accepted by \( \mathcal{A} \), which leads to a contradiction.
Recapitulation

- **LTL formulae:**

  \[ \varphi, \psi ::= p \mid \neg \varphi \mid \varphi \land \psi \mid \varphi \lor \psi \mid X\varphi \mid \varphi \cup \psi \]

- **Models:** \( \omega \)-sequences over \( \mathcal{P}(PROP) \).

- **Models(\( \varphi \)):** the set of sequences \( \rho \) in \( \mathcal{P}(PROP)^\omega \) 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

- \( L(\mathcal{A}) = \text{Models}(p \cup p') \).

\[
\begin{align*}
\{p, p'\}, \{p\} \\
\Sigma \\
\{p, p'\}, \{p'\}
\end{align*}
\]

- There is a Büchi automaton \( \mathcal{A}_\varphi \) s.t.
  1. \( L(\mathcal{A}_\varphi) = \text{Models}(\varphi) \); \( |\mathcal{A}_\varphi| \) is in \( 2^{O(|\varphi|)} \),
  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 \( \varphi \).
Closure set

• Closure set contains all the formulae we need to consider to check satisfiability.

• Closure $\text{cl}(\varphi)$ of $\varphi$: smallest set
  
  • containing the subformulae of $\varphi$,
  
  • closed under negation (we identify $\neg\neg\psi$ with $\psi$),
  
  • if $\chi_1 \cup \chi_2 \in \text{cl}(\varphi)$, then $X(\chi_1 \cup \chi_2) \in \text{cl}(\varphi)$.

• Cardinal of $\text{cl}(\varphi)$ is linear in the size of $\varphi$. 
Atom

- An atom is nothing but a maximally consistent subset of $\text{cl}(\varphi)$.

- Atom $X$: subset of $\text{cl}(\varphi)$ s.t.
  
  1. for all formulae $\psi$ in $\text{cl}(\varphi)$, $\psi \in X$ iff $\neg \psi \notin X$,
  
  2. $\psi_1 \land \psi_2 \in X$ iff $\psi_1, \psi_2 \in X$,
  
  3. $\psi_1 \lor \psi_2 \in X$ iff $\psi_1 \in X$ or $\psi_2 \in X$,
One-step consistent pairs

- \((Y, Y')\) is one-step consistent:
  
  - if \(\psi_1 \cup \psi_2 \in Y\), then \(\psi_2 \in Y\) or (\(\psi_1 \in Y\) and \(\psi_1 \cup \psi_2 \in Y'\)).
    
    (\(\psi_1 \cup \psi_2\) is equivalent to \(\psi_2 \lor (\psi_1 \land X \psi_1 \cup \psi_2\)).)

  - for \(X \psi \in \text{cl}(\varphi)\), \(X \psi \in Y\) iff \(\psi \in Y'\).

- The states of \(A_{\varphi}\) are atoms and the transition relation contains one-step consistent pairs.

- Each state \(X\) of \(A_{\varphi}\) 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 \(A_{\varphi}\) allows us to propagate the constraints.
Defining $A_\varphi$

- GBA $A_\varphi = (\Sigma, Q, Q_0, \delta, F_1, \ldots, F_\alpha)$:
  - $\Sigma = \mathcal{P}(\{p_1, \ldots, p_N\})$,
  - $Q$ is the set of atoms,
  - $Q_0$ is the subset of atoms containing $\varphi$,
  - $X \xrightarrow{a} Y \in \delta$ iff $a = \{p_1, \ldots, p_N\} \cap X$ and $(X, Y)$ is one-step consistent,
  - for $\psi_1 \cup \psi_2 \in \text{cl}(\varphi)$, there is exactly one set $F_i$ such that $F_i = \{X \in Q : \text{either } \psi_1 \cup \psi_2 \notin X \text{ or } \psi_2 \in X\}$.

- Accepting conditions $F_1, \ldots, F_\alpha$ guarantee that the search for witnesses is not delayed forever.
Main properties for $A_\varphi$

- $L(A_\varphi) = \text{Models}(\varphi)$.

- Given a model $\rho : \mathbb{N} \to \mathcal{P}(\{p_1, \ldots, p_N\}) \in \text{Models}(\varphi)$, there is a unique accepting run $X_0 \xrightarrow{\rho(0)} X_1 \xrightarrow{\rho(1)} X_2 \xrightarrow{\rho(2)} \cdots$ in $A_\varphi$ s.t. $(\star)$ for $i \geq 0$ and $\psi \in \text{cl}(\varphi)$, 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 $A_\varphi$, the model $\rho$ defined by $\rho(i) = a_i$ for $i \geq 0$ satisfies $(\star)$.

- In the sequel, we write $A_\varphi$ to denote the Büchi automaton recognizing $\text{Models}(\varphi)$.
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, e.g. \((x_1 > x_2)\).

- Comparing values at successive positions, e.g. \(x_1 > X x_2\).

- First-order quantification over counter values, e.g. 
  \[ \exists y \ G(x_1 \leq y). \approx \text{“Along the run, counter 1 is bounded.”} \]
**LTL$^{CS}(PrA)$ syntax**

- Queen logic LTL$^{CS}(PrA)$ (fragments are defined from it).
- Giving up the standard abstraction: propositional variables understood as properties about the current configuration.
- $\text{VAR}^{p} = \{y_1, y_2, \ldots\}$: set of integer variables.
- $\text{VAR} = \{x_1, x_2, \ldots\}$: set of counter variables.
- $Q = \{q_1, q_2, \ldots\}$: set of control state symbols.
- LTL$^{CS}(PrA)$ formulae:
  \[
  \varphi ::= \psi \mid q \mid \varphi \land \varphi \mid \neg \varphi \mid X\varphi \mid \varphi U \varphi \mid \exists y \varphi
  \]
  - $\psi$ is a Presburger formula with free variables included in $\text{VAR}^{p} \cup \text{VAR}$,
  - $q \in Q$. 
Satisfaction relation

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

- Hence, we disregard sequences with an infinite amount of control state symbols; infinite runs of counter systems are models.

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

- $\rho, i \models_E q \iff q = q_i$.

- $\rho, i \models_E \psi \iff \mathbf{v}_i \models \psi$ in PrA with $\mathbf{v}_i$ extends $E$ s.t. $\mathbf{v}_i(x_j) = \vec{x}_i(j) (j \in [1, n])$, assuming $\psi$ is a Presburger formula with free variables in $\text{VAR}^p \cup \{x_1, \ldots, x_n\}$.

- $\rho, i \models_E x \varphi \iff \rho, i + 1 \models_E \varphi$.

- $\rho, i \models_E \exists y \varphi$ iff there is $k \in \mathbb{N}$ such that $\rho, i \models_E [y \mapsto k] \varphi$. 
Decision problems for $\text{LTL}^\text{CS}(\text{PrA})$

- **Semi-closed formula**: no variable from $\text{VAR}^p$ is free. $F(x_1 = y)$ is not semi-closed unlike $G(x_1 > x_2)$ and $\exists y \ G(x_1 \leq y)$.

- **SATISFIABILITY PROBLEM**
  
  **Input**: An $\text{LTL}^\text{CS}(\text{PrA})$ semi-closed formula $\varphi$ with free counter variables $x_1, \ldots, x_n$.
  
  **Question**: Is there a model $\rho \in (Q \times \mathbb{N}^n)$ s.t. $\rho, 0 \models \emptyset \varphi$?

- **EXISTENTIAL MODEL-CHECKING PROBLEM**
  
  **Input**: CS $S = (Q, n, \delta), (q_0, \vec{x}_0)$ and semi-closed formula $\varphi$ with free variables in $\{x_1, \ldots, x_n\}$.
  
  **Question**: Is there an infinite run $\rho$ starting at $(q_0, \vec{x}_0)$ such that $\rho, 0 \models \emptyset \varphi$?

(Infinite runs of CS are $\text{LTL}^\text{CS}(\text{PrA})$ models)
A simple reduction

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

• Let $S$, $(q, \bar{x})$ and $q_f$ be an instance.

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

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

\[
\bigwedge_{i=1}^{N} \left( \bigvee_{q' \in F_i} G F q' \right)
\]
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]

- Temporal semantics of imperative programs.
  [Manna & Pnueli, 1992]

  Program variable $x$ never decreases below its initial value:

  $$
  \exists y \ (x = y) \land G(x \geq y)
  $$

- and many many others . . .
LTL$^\text{CS}(\text{PrA})$ is sometimes too expressive!

- Model-checking restricted to LTL($Q$) is already undecidable.
  
- ... but LTL$^\text{CS}(\text{PrA})$ 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 LTL$^\text{CS}(\text{PrA})$ to regain decidability for satisfiability.

- Two types of restrictions on first-order quantification:
  - Only used to speak about successive counter values.
  - Only used to store a counter value and possibly perform equality tests later.
Fragments

LTL^{CS}(PrA)

LTL(q) \approx LTL

CLTL(PrA)

CLTL(QFP)

CLTL(DL^+)

CLTL(DL)

CLTL(IPC*)
Presburger LTL $\mathsf{CLTL}^{(\mathsf{PrA})}$
A macro for comparing successive counter values

- From PrA formula $\psi(z_1, \ldots, z_k)$,
  
  $$\psi(X^{i_1}x_{j_1}, \ldots, X^{i_k}x_{j_k}) \overset{\text{def}}{=} (\exists y_1, \ldots, y_k \ X^{i_1}(y_1 = x_{j_1}) \land \cdots \land X^{i_k}(y_k = x_{j_k}) \land \psi(y_1, \ldots, y_k))$$

- For instance, $x_1 = Xx_2$ is obtained from $z_1 = z_2$.
  
  $$\exists y_1 \ y_2 \ (y_1 = x_1) \land X(y_2 = x_2) \land y_1 = y_2$$

- $G(x_1 = Xx_1)$ states that first counter has constant value.

- CLTL(PrA): fragment of LTL$^{CS}$(PrA) by allowing quantification only in $\psi(X^{i_1}x_{j_1}, \ldots, X^{i_k}x_{j_k})$.

- $X$ and $x$ are of distinct nature but both refer to the next position.
The fragment CLTL(PrA)

- Terms and formulae:
  \[ t ::= 0 \mid 1 \mid X^i x \mid t + t \]
  \[ \varphi ::= t \equiv_k t \mid t < t \mid q \mid \varphi \land \varphi \mid \neg \varphi \mid X \varphi \mid \varphi U \varphi. \]
  with \( x \in \text{VAR} \), \( k > 1 \) and \( q \in Q \).

- NB: no integer variables, quantifier-free Presburger formulae (hence, no need for environment).

- \( G(x_1 < X x_1) \) is satisfiable; \( G(x_1 > X x_1) \) is not.

- \( \sigma, i \models \psi(X^{l_1} x_1, \ldots, X^{l_n} x_n) \) iff
  \( (\sigma_1(i + l_1)(x_1), \ldots, \sigma_1(i + l_n)(x_n)) \in \text{REL}(\psi) \) with
  - \( \sigma = (\sigma_1, \sigma_2) \),
  - \( \sigma_1 : \mathbb{N} \rightarrow (\text{VAR} \rightarrow \mathbb{N}) \),
  - \( \sigma_2 : \mathbb{N} \rightarrow Q \) for some finite subset \( Q \subseteq Q \).

- Model-checking and satisfiability problems as subproblems for LTL^{CS}(PrA).
**X-length**

- **X-length of \( \varphi \):** maximal \( l \) such that \( \mathbf{X}^l x \) occurs in \( \varphi \).

- **CLTL\(_n\)(PrA):** restriction of CLTL(PrA) to formulae with at most \( n \) variables and \( \mathbf{X} \)-length less or equal to \( l \).

- There is a logspace reduction from the satisfiability problem for CLTL(PrA) to its restriction with \( \mathbf{X} \)-length 1.
  - Expressions \( x_1, \ldots, \mathbf{X}^3 x_1 \) are encoded by
    
    \[ G(x'' = \mathbf{X} x' \land x' = \mathbf{X} x \land x = \mathbf{X} x_1) \]

    - \( \mathbf{X} x_1 \) [resp. \( \mathbf{X}^2 x_1, \mathbf{X}^3 x_1 \)] is replaced by \( x \) [resp. \( x', x'' \)].
    - Take the conjunction.
Two fragments of PrA

- Difference logic DL:

\[ \varphi ::= x \sim y + d \mid x \sim d \mid \varphi \land \varphi \mid \neg \varphi \]

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

- Quantifier-free Presburger arithmetic QFP:

\[ \varphi ::= \sum_{i \in I} a_i x_i \sim d \mid \sum_{i \in I} a_i x_i \equiv_k c \mid \varphi \land \varphi \mid \neg \varphi \]

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

- Satisfiability problem for QFP is NP-complete.

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

<table>
<thead>
<tr>
<th></th>
<th>MC (DL)</th>
<th>SAT</th>
<th>MC (CA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{CLTL}_3^1(\text{DL})$</td>
<td>$U$</td>
<td>$U$</td>
<td>$U$</td>
</tr>
<tr>
<td>$\text{CLTL}_2^\omega(\text{DL})$</td>
<td>$U$</td>
<td>$U$</td>
<td>$U$</td>
</tr>
<tr>
<td>$\text{CLTL}_2^2(\text{DL})$</td>
<td>$U$</td>
<td>$U$</td>
<td>$\text{PSPACE-c}$</td>
</tr>
<tr>
<td>$\text{CLTL}_1^1(\text{DL})$</td>
<td>$U$</td>
<td>$U$</td>
<td>$\text{PSPACE-c}$</td>
</tr>
<tr>
<td>$\text{CLTL}_2^1(\text{DL} \text{ or } \text{DL}^+)$</td>
<td>$\text{PSPACE-c.}$</td>
<td>$\text{PSPACE-c.}$</td>
<td>$\text{PSPACE-c}$</td>
</tr>
<tr>
<td>$\text{CLTL}_1^1(\text{QFP})$</td>
<td>$U$</td>
<td>$U$</td>
<td>$\text{PSPACE-c}$</td>
</tr>
<tr>
<td>$\text{CLTL}_1^\omega(\text{QFP})$</td>
<td>$U$</td>
<td>$U$</td>
<td>$\text{PSPACE-c.}$</td>
</tr>
</tbody>
</table>

$\text{CLTL}_n^l(L)$: restriction of $\text{CLTL}(L)$ to formulae with at most $n$ variables and $X$-length less or equal to $l$. 
Satisfiability problem for $\text{CLTL}_{3}^{1}(\text{DL})$ is undecidable

- Simple reduction from halting problem for Minsky machines.

- Formulae in $\text{CLTL}_{3}^{1}(\text{DL})$ can easily internalize the instructions of Minsky machines.

- $S$ has $n$ instructions, the $n$th one halts.

- $x_1$ and $x_2$ encode counter values, $x_3$ encodes the instruction ordinal.

- $S$ does not halt iff formula below is satisfiable:

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

initial configuration
Internalizing instructions from Minsky machines

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

$$\psi_i \overset{\text{def}}{=} G(x_3 = i \Rightarrow (Xx_3 = i') \land (x_j + 1 = Xx_j) \land (x_{3-j} = Xx_{3-j})).$$

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

$$G(x_3 = i \Rightarrow ((x_j = 0 \Rightarrow (Xx_3 = i') \land (x_j = Xx_j) \land (x_{3-j} = Xx_{3-j}))) \land (x_j \neq 0 \Rightarrow (Xx_3 = i'') \land (x_j - 1 = Xx_j) \land (x_{3-j} = Xx_{3-j})))$$

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

• Today’s lecture:
  • Standard LTL
  • Logic $\text{LTL}^\text{CS}(\text{PrA})$ for counter systems
  • Presburger LTL

• Tomorrow’s lecture: Model-checking counter systems