Verification of temporal logics on infinite-state systems

Lecture 3.2
Decidable and undecidable reachability problems for counter systems

Stéphane Demri and Valentin Goranko

ESSLLI 2007, Trinity College, August 2007
Overview

Semilinear sets
- Sets of tuples
- Presburger arithmetic

Presburger Counter Systems

Minsky machines
- Deterministic machines
- Nondeterministic machines
- One-counter Minsky machines

VASS
- Definition
- Problems

Counter automata with errors
- Counter automata
- Lossy counter automata with reset
- Gainy counter automata
- Recurrence problem

Reversal-bounded counter automata
- Reversals
- Parikh images
- Reachability relation is semilinear

Relational counter automata
- Definition
- Reachability relation is semilinear
Counter systems

- Model-checking of infinite-state systems needed for formal verification.

- Ubiquity of counter systems (CS)
  - Embedded systems/protocols, Petri nets, ...
  - Programs with pointer variables. [Bardin et al., AVIS 06; Bouajjani et al., CAV 06]
  - Broadcast protocols. [Leroux & Finkel, FSTTCS 02]
  - Logics for data words. [Bojańczyk et al, LICS 06]

- (High) undecidability
  - Checking safety properties for CS is undecidable.
  - Checking liveness properties for CS is $\Sigma^1_1$-hard.
Taming counter systems

- Classes with decidable reachability problems
  - Reversal-bounded CS. [Ibarra, JACM 78]
  - Flat relational CS. [Comon & Jurski, CAV 98]
  - Flat linear CS. [Boigelot, PhD 98; Finkel & Leroux, FSTTCS 02]
  - Petri nets. [Kosaraju, STOC 82]

- Decision procedures
  - Translation into Presburger arithmetic. [Ibarra, JACM 78, Comon & Jurski, CAV 98]
  - Well-structured transition systems. [Finkel & Schnoebelen, TCS 01]

- Tools: FAST, LASH, TReX, ...
Semilinear sets

Given \( P = \{c_1, \ldots, c_m\} \) a finite subset of \( \mathbb{N}^k \), \( P^* \) is defined by

\[
P^* = \{ \sum_{i=1}^{m} a_i c_i : a_i \in \mathbb{N}, 1 \leq i \leq m \}.
\]

Linear set: \( c + P^* \) with \( x \in \mathbb{N}^k \) and \( P = \{c_1, \ldots, c_m\} \):

- \( c \) is the basis.
- \( P \) is the set of periods.

Semilinear set: finite union of linear sets.
(encoded by a finite amount of pairs (basis, periods))

Semilinear sets are closed under Boolean operations.

[Ginsburg & Spanier, PJM 66]
Examples

- A linear set:
  \[
  \left\{ \left( \begin{array}{c} 3 \\ 4 \end{array} \right) + i \times \left( \begin{array}{c} 2 \\ 5 \end{array} \right) + j \times \left( \begin{array}{c} 4 \\ 7 \end{array} \right) : i, j \in \mathbb{N} \right\}
  \]

- Subsets of \( \mathbb{N} \) that are not semilinear:
  - \( \{2^i : i \in \mathbb{N}\} \).
  - \( \{i^2 : i \in \mathbb{N}\} \).

- \( X \subseteq \mathbb{N} \) is semilinear iff there are \( N, M \in \mathbb{N} \) such that for every \( n \geq N, n \in X \) iff \( n + M \in X \). (\( X \) is ultimately periodic)
Presburger arithmetic [Presburger, 29]

- “First-order theory of \((\mathbb{N}, +, =)\)” (no multiplication).

- Example: \((\exists z \ x = 2z) \Rightarrow (\exists z \ y = 3z)\).

- Atomic formulae: \(\sum_{i \in I} x_i = \sum_{j \in J} y_j\) with \(x_i, y_j\) variables interpreted in \(\mathbb{N}\).

- Formulae:

  \[
  \phi ::= \sum_{i \in I} x_i = \sum_{j \in J} y_j \mid \neg \phi \mid \phi \land \phi \mid \exists x \ \phi.
  \]

- Satisfaction relation \(v \models \phi\) with \(v : \text{VAR} \to \mathbb{N}\).
  
  e.g. \(v \models \sum_{i \in I} x_i = \sum_{j \in J} y_j\) iff \(\sum_{i \in I} v(x_i) = \sum_{j \in J} v(y_j)\).
Expressive power

Assuming a total ordering of variables, a Presburger formula \( \phi(x_1, \ldots, x_k) \) with \( k \) free variables defines the subset of \( \mathbb{N}^k \)
\[
\{(n_1, \ldots, n_k) \in \mathbb{N}^k : (n_1, \ldots, n_k) \models \phi(x_1, \ldots, x_k)\}.
\]

Semilinear sets coincide with sets definable by Presburger formulae [Ginsburg & Spanier, PJM 66].

Presburger formula for
\[
\left\{ \left( \begin{array}{c} 3 \\ 4 \end{array} \right) + i \times \left( \begin{array}{c} 2 \\ 5 \end{array} \right) + j \times \left( \begin{array}{c} 4 \\ 7 \end{array} \right) : i, j \in \mathbb{N} \right\}
\]
\[\exists i, j \ x_1 = 3 + 2i + 4j \land x_2 = 4 + 5i + 7j\]
Complexity

- Satisfiability problem for Presburger arithmetic is decidable. [Presburger, 29]

- Quantifiers can be eliminated assuming that atomic formulae $x \equiv_m m'$ are added.

- Satisfiability is in $3\text{ExpTime}$.

- Quantifier-free Presburger arithmetic is NP-complete.
Presburger constraints on graphs/trees

- Constraints in counter automata.
- Constraints on the number of event occurrences.  
  [Bouajjani & Echahed & Habermehl, LICS 95]
- Constraints on XML documents.  
  [Dal Zilio & Lugiez, RTA 03]
- LTL with counters.  
  [Comon & Cortier, CSL 00]
- Graded modal logics ($\Diamond \geq 3 \ p$).  
  [Fine, NDJFL 72]
- ...and many more examples (description logics, FO, MSO, etc.)
Presburger counter systems (PCS) \((\Sigma, Q, T)\)

- \(x_1' = x_1 + 1\)
- \(x_2' = x_2 + 1\)
- \(x_3' = x_3 + 1\)

Labels: Presburger formulae over

\(x_1' = x_2' = x_3' = 0\)
Presburger transition systems (PTS)

\[ \mathcal{C} = (\Sigma, Q, T) \quad \mapsto \quad S_C = (S, \rightarrow) \]

\[ S = Q \times \mathbb{N}^n. \]

\[ \langle q, a \rangle \rightarrow \langle q', a' \rangle \text{ iff } \exists q \xrightarrow{\psi(x, x')} q' \in T \text{ s.t. } a, a' \models \psi(x, x'). \]
Minsky machines
Deterministic Minsky machines [Minky, book 67]

- A counter stores a single natural number.

- A Minsky machine can be viewed as a finite-state machine with two counters.

- Operations on counters:
  - Check whether the counter is zero.
  - Increment the counter by one.
  - Decrement the counter by one if nonzero.
2-counter Minsky machines

- Set of $n$ instructions.

- The $l$th instruction has one of the forms below ($i \in \{0, 1\}$, $l' \in \{1, \ldots, n\}$):
  
  1. $l$: $C_i := C_i + 1; \text{ goto } l'$
  2. $l$: if $C_i = 0$ then goto $l'$ else $C_i := C_i - 1; \text{ goto } l''$.

- Configurations are elements of $\{1, \ldots, n\} \times \mathbb{N} \times \mathbb{N}$.

- Initial configuration: $(1, 0, 0)$. 
Computation

- A computation is a sequence of configurations starting from the initial configuration and such that two successive configurations respect the instructions.

- \((1, 0, 0) \rightarrow (2, 1, 0) \rightarrow (1, 1, 1) \rightarrow (2, 2, 1) \rightarrow (1, 2, 2) \rightarrow (2, 3, 2) \ldots\)

  1: \text{ } C_0 := C_0 + 1; \text{ goto 2}

  2: \text{ } C_1 := C_1 + 1; \text{ goto 1}
Halting problem

- **Halting problem:**
  - **input:** a 2-counter Minsky machine $M$;
  - **question:** is there a finite computation that ends with location counter equal to $n$?

- The halting problem is $\Sigma^0_1$-complete.

- 2-counter Minsky machines are Turing-complete:
  1. A Turing machine can be simulated by two stacks.
     (the infinite tape is cut in half).
  2. A stack can be simulated by two counters.
     (one of the counters is the binary representation of the bits on the stack)
  3. Four counters can be simulated by two counters.
     (the factorization of one of the counters is $2^{a3b5c7d}$).
Non-deterministic Minsky machines

- Nondeterministic choice after incrementation and decrementation.

- Instructions are of the forms below:
  
  \[ l: \quad C_i := C_i + 1; \text{ goto } l' \text{ or goto } l'' \]
  
  \[ l: \quad \text{if } C_i = 0 \text{ then goto } l' \text{ else } C_i := C_i - 1; \text{ goto } l'' \text{ or goto } l_1'' \]

- Recurrence problem:
  
  \text{input: a 2-counter Minsky machine } M; \]

  \text{question: is there an infinite computation with location counter equal to 1 infinitely often?}

- The recurrence problem is $\Sigma_1^1$-complete.

[Alur & Henzinger, JACM 94]
One-counter deterministic Minsky machines

- Restriction to a single counter $C_0$.

- There is a unique infinite computation of the form 
  $(l_0, n_0) \rightarrow (l_1, n_1) \rightarrow (l_2, n_2) \ldots$.

- Given a 1-counter machine $M$, one can compute a Presburger formula $\phi(x, y, z)$ in polynomial-time in $|M|$ such that 
  $$a, b, i \models \phi(x, y, z) \iff (l_i, n_i) = (a, b).$$

- $\Rightarrow$ to compute the $i$th configuration, there is no need to compute the preceding configurations.
Encoding simple problems in Presburger arithmetic

- The infinite computation has a "saucepan skeleton".

- An instance of the halting problem can be solved by checking satisfiability for $\exists y, z \, \phi(x, y, z) \land x = n$.

- The recurrence problem can be solved with

$$\forall z \, \exists z' \, z < z' \land \exists x, y \, \phi(x, y, z') \land x = 1$$
Semilinear sets
Presburger Counter Systems
Minsky machines
VASS
Counter automata with errors
Reversal-bounded counter automata
Relational counter automata

VASS
What is a VASS?

- A Vector Addition System with States (VASS) is a finite-state automaton with transitions labelled by tuples of integers.

- A VASS of dimension $k \geq 1$ is a structure $\mathcal{A} = (Q, q_0, F, \delta)$ such that
  - $Q$ is a finite set of control states.
  - $q_0 \in Q$ and $F \subseteq Q$.
  - $\delta$ is a finite subset of $Q \times \mathbb{Z}^k \times Q$.

- A configuration $(q, c)$ is an element of $Q \times \mathbb{N}^k$.

- One-step relation between configurations:
  $$(q, c) \rightarrow (q', c')$$ iff there is $(q, d, q') \in \delta$ s.t. $c' = c + d$. 
Reachability problem

- Reachability problem
  
  **input:** Configurations \((q, c), (q', c')\) from a VASS \(A\);
  
  **question:** Does \((q, c) \xrightarrow{*} (q', c')\) in \(A\)?

- The reachability problem is decidable.
  
  [Mayr, STOC 81; Kosaraju, STOC 82; Reutenauer, book 90]

- Best known complexity lower bound: \(\text{ExpSpace}\)-hardness.
  
  [Lipton, TR 76]

- No known primitive recursive upper bound.

- Checking equality between accessibility sets of two configurations is undecidable.
Coverability/Covering problem

Definition

- **input:** Configurations \((q, c), (q', c')\) from a VASS \(A\);
- **question:** Is there a configuration \((q'', c'')\) such that
  \((q, c) \rightarrow^* (q', c'')\) and \(c' \leq c''\)?

(state to set reachability problem)

The coverability problem is
- decidable,
- \text{ExpSpace-hard},
- in \text{ExpSpace}.

The control-state reachability problem is a subproblem of the coverability problem with \(c' = 0\).

[Karp & Miller, JSCC 69]
[Lipton, TR 75]
[Rackoff, TCS 78]
Boundedness problem

- **Definition**

  **input:** a configuration \((q, c)\) from a VASS \(A\);
  
  **question:** Is \(\{(q', c') : (q, c) \rightarrow^* (q', c')\}\) finite?

- The boundedness problem is also \(\text{ExpSpace}\)-complete.

  [Lipton, TR 75; Rackoff, TCS 78]

- In case of finiteness, the transition system from \((q, c)\) can be equivalently represented by a finite-state automaton.

- Boundedness for VASS with resets is undecidable.

  [Dufourd & Finkel & Schnoebelen, ICALP 98]
Extended VASS

- Set of Instructions:
  - Adding $d \in \mathbb{Z}^k$ $\text{add}(d)$ (standard instruction in VASS).
  
  - Zero-test $test(i)$: $(q, c) \xrightarrow{test(i)} (q', c')$ iff $c(i) = 0$ and $c = c'$.

  - Reset $\text{reset}(i)$: $(q, c) \xrightarrow{\text{reset}(i)} (q', c')$ iff $c'(i) = 0$ and $c(j) = c'(j)$ for $j \neq i$.

  - Transfer $\text{transfer}(i \rightarrow j)$: $(q, c) \xrightarrow{\text{transfer}(i \rightarrow j)} (q', c')$ iff
    1. $c'(i) = 0$, $c'(j) = c(j) + c(i)$,
    2. for $k \neq i, j$, $c'(k) = c(k)$. 

Stéphane Demri and Valentin Goranko
Verification of temporal logics on infinite-state systems Lecture
A class of E-VASS

- Class of E-VASS $T_{1,2} R_{1,2} \text{Tr}_{12}$ [Finkel & Sutre, STACS 00]
  - $k = 2$,
  - set of instructions: $\text{add}(v)$ for $v \in \mathbb{Z}^2$ and, $\text{test}(1)$, $\text{reset}(1)$, $\text{reset}(2)$ and $\text{transfer}(1 \rightarrow 2)$.

- The reachability problem is decidable for the class $T_{1,2} R_{1,2} \text{Tr}_{12}$.
- For any E-VASS $\mathcal{A}$ in $T_{1,2} R_{1,2} \text{Tr}_{12}$ and semilinear set $X_0$ of configurations of $\mathcal{A}$,
  
  $\text{pre}^*(X_0)$ is semilinear and effectively computable.

- Consequently, the control-state reachability problem is also decidable.

- See other classes in [Finkel & Sutre, STACS 00].
Counter automata with errors
Counter automata

- Counter automaton $A = (Q, q_0, n, \delta, F)$
  - $Q$ is a finite set of locations and $q_0 \in Q$,
  - $n \in \mathbb{N}$ is the number of counters,
  - $\delta \subseteq Q \times L \times Q$ is the transition relation where $L = \{\text{inc}, \text{dec}, \text{ifzero}\} \times \{1, \ldots, n\}$,
  - $F \subseteq Q$.

- A counter valuation is a function $v : \{1, \ldots, n\} \rightarrow \mathbb{N}$.

- A configuration is a pair $(q, v) \in Q \times \mathbb{N}^n$.

- Accessibility relation $\rightarrow$ between configurations is defined in the usual way following the instructions.

- The reachability problem for counter automata is undecidable. (counter automata generalize Minsky machines).
A lossy counter automaton with reset is a structure $(Q, q_0, n, \delta, F)$ with $L = \{inc, dec, reset\} \times \{1, \ldots, n\}$.

Lossy accessibility relation:

$(q, v) \xrightarrow{lossy} (q', v')$ iff there are $v_\leq v$ and $v'_+ \geq v'$ such that $(q, v_\leq) \xrightarrow{perf} (q', v'_+)$. 

The control-state reachability problem for lossy counter automata with reset is decidable.

Consequence of the decidability of the coverability problem for reset Petri nets [Dufourd & Finkel & Schnoebelen, ICALP 98].

Lossy counter automata (with no reset) fromm a subclass of lossy channel systems. See a survey in [Schnoebelen, 05].
Gainy counter automata

- $A = (Q, q_0, n, \delta, F)$ can be viewed as a gainy counter automata:
  
  $(q, v) \xrightarrow{\text{gainy}} (q', v')$ iff there are $v_+ \geq v$ and $v'_- \leq v'$ such that $(q, v_+) \xrightarrow{\text{perf}} (q', v'_-)$. 

- From a gainy counter automaton $A$, one can effectively compute in logspace a lossy counter automaton with reset $A'$ such that $A$ has an accepting run iff $A'$ has an accepting run.

- Consequently, the control-state reachability problem for gainy counter automata is decidable.

- The control-state reachability problem for gainy counter automata is not primitive recursive, as a consequence of [Schnoebelen, IPL 02].
Recurrence problem is undecidable

- Recurrence problem for gainy counter automata:
  
  **input:** a gainy counter automaton $A = (Q, q_0, n, \delta, F)$;  
  
  **question:** is there an infinite computation  
  $$(q_0, 0) \rightarrow_{\text{gainy}} (q_1, v_1) \rightarrow_{\text{gainy}} (q_2, v_2) \ldots$$  
  such that infinitely often $q_i \in F$?

- The recurrence problem is undecidable [Demri & Lazić, TR 06].

- The proof is by adapting the proof for undecidability of the recurrence problem for Insertion Channel Machines with Emptiness-Testing (ICMET).
  
  [Ouaknine & Worrell, FOSSACS 06]
Proof sketch (undecidability)

- Non-reachability of a control state $q$ in Minsky machine $M$ can be reduced to recurrence problem in gainy counter automata.

- From $M$, one build a counter automaton $M'$ with 3 counters such that
  - $C_1$ and $C_2$ simulate the counters in $M$,
  - $C_3$ is incremented after each instruction of $M$.

- We build a gainy counter automaton $A$ with counters $C_1$, $C_2$, $C_3$, $D_1$, $D_2$ and $C$. 
After location (2), $D1$ is the maximal value of the counters in the simulation of $M'$. 

1. $\text{simulate } M'$
   - $C_{i+} \Rightarrow D1--$ if nonzero
   - $C_{i-} \Rightarrow D1++$
   - $q$ is reached $\Rightarrow$ go to dead-end state

2. $D1 := C$

3. $D1 = 0$

4. $D1 := D1 + C1 + C2 + C3$
   - $C1 := 0$; $C2 := 0$; $C3 := 0$
   - $D1 := D1 - 1$

(Add an arrow from 4 to 3 if $D1 \neq 0$)
Correctness

- Minsky machine \( M \) cannot reach \( q \) iff \( A \) visits infinitely often location (1).

- Minsky machine \( M \) cannot reach \( q \) iff counter automaton \( M' \) cannot reach \( q \).

- If \( M' \) cannot reach \( q \) then an error-free run of \( A \) visits infinitely often (1).

- For the converse direction we use the following facts:
  - In (4), the only way to decrement \( D_1 \) is to simulate exactly \( M' \).
  - In order to reach (1), in (3)-(4) \( D_1 \) is decremented regularly.
  - If \( A \) visits infinitely often (1) and \( M' \) can reach \((q, n_1, n_2, n_3)\) then at some point an error-free simulation of \( M' \) shall be done with \( D_1 \geq n_1 + n_2 + n_3 \), a contradiction.
Reversal-bounded counter automata
Reversals

- Reversal: Alternation form nonincreasing mode to nondecreasing mode and vice-versa.

- Sequence with 0 reversal:
  
  00111333555577777

- Sequence with 3 reversals:
  
  0011223334444错过了5557777

- Reversal-bounded counter automata: there is $r \geq 0$ such that for any computation, every counter makes no more than $r$ reversals.

- Reversal-bounded machines introduced in [Ibarra, JACM 78].
Parikh image

- $\Sigma = \{a_1, \ldots, a_k\}$.

- Parikh image of $w \in \Sigma^*$: \[
\begin{pmatrix}
i_1 \\
i_2 \\
\vdots \\
i_k
\end{pmatrix} \in \mathbb{N}^k
\] where each $i_j$ is the number of occurrences of $a_j$ in $w$.

- Parikh image of $a\ b\ a\ a\ b$ is \[
\begin{pmatrix}3 \\ 2\end{pmatrix}.
\]

- The Parikh image of any context-free language is semilinear. 
  
  [Parikh, JACM 66]

- Effective computation from pushdown automata.
Reversal-bounded counter automata with alphabet

- Transition relation of the form
  \[ Q \times L \times \Sigma \times Q \]

- \( L(\mathcal{A}) \subseteq \Sigma^* \): language accepted by \( \mathcal{A} \).

- The Parikh image of languages accepted by every reversal-bounded counter automata with alphabet \( \mathcal{A} \) is semilinear and it is effectively computable from \( \mathcal{A} \).
  
  
  [Ibarra, JACM 78]

- Let \( \mathcal{A} \) be a reversal-bounded counter automaton. One can effectively compute a formula \( \phi \) such that for \( c, c' \in \mathbb{N}^k \),

  \[
  (q, c) \xrightarrow{*} (q', c') \text{ iff } c, c' \models \phi.
  \]
Flat relational counter automata
Definition

- A relational counter automata is a structure \( \mathcal{A} = (Q, C, \delta) \) such that
  - \( Q \) is a finite set of control states,
  - \( C \) is a finite set of counters,
  - \( \delta \subseteq Q \times \text{guards}(C) \times Q \) is the transition relation.

- A guard in \( \text{guards}(C) \) is a conjunction of expressions of the form
  \[
  x \sim y + c, \quad x \sim c
  \]
  where \( x, y \in C \cup C', \ c \in \mathbb{Z} \) and \( \sim \in \{\geq, \leq, =, >, <\} \).

- A configuration of \( \mathcal{A} \) is an element \((q, c)\) in \( Q \times \mathbb{N}^{|C|} \).

- Relational counter automata include the class of Minsky machines.
Properties [Comon & Jurski, CAV 98]

- Given two transitions \((q, g, q')\) and \((q', g', q'')\), there is a guard \(g''\) such that for all \(c, c', c'' \in \mathbb{N}^{|C|}\),

\[
(q, c) \xrightarrow{g} (q', c') \xrightarrow{g'} (q'', c'') \iff (q, c) \xrightarrow{g''} (q'', c'').
\]

- \(x' \geq x + 1 \land y' \leq y, x' \leq y \land x' = y' \Rightarrow x' \leq y \land x' = y'\).

- Given a transition \((q, g, q)\), one can effectively compute a Presburger formula \(\phi\) with free variables among \(C \cup C'\) such that for all \(c, c' \in \mathbb{N}^{|C|}\),

\[
(q, c)(\xrightarrow{g})^*(q, c') \iff c, c' \models \phi.
\]

(the proof is difficult)
Flatness

A relational counter automata is flat if every control state belongs to at most one cycle with no repeated vertex.
Reachability relation is Presburger-definable

Let $A$ be a flat relational counter automata and $q, q' \in Q$. One can effectively compute a Presburger formula $\phi$ with free variables $C \cup C'$ such that for all $c, c' \in \mathbb{N}^k$,

$$(q, c) \xrightarrow{*} (q', c') \iff c, c' \models \phi.$$ 

The reachability problem for flat relational counter automata is decidable.
Sketch of the proof

- For each cycle $q_1 \xrightarrow{g_1} q_2 \xrightarrow{g_2} \ldots \xrightarrow{g_N} q_N$ ($q_1 = q_N$) compute the equivalent transition $(q_1, g, q_1)$.

- For $q, q'$, enumerate the path schemas between $q$ and $q'$

- Compute the formula for accessibility by composition.
Regaining decidability for the reachability problem

- Restrictions on the control graph.
  See e.g. the flat relational counter automata.

- Restrictions on the sets of instructions.
  See e.g. the vector addition systems with states (no zero-test).

- Errors in the computation.
  See e.g. gainy/lossy counter automata.

- Semantical conditions on the behaviors of counters.
  See e.g. the reversal-bounded counter automata (верхнее и нижнее соответственно).
Plan for future lectures

- Thursday 9h-10h30: Logics for counter systems
  - Temporal logics for counter systems
  - Presburger counter systems and acceleration

- Friday 9h-10h30: Timed Systems
  - Reachability problems for timed automata
  - Timed temporal logics
Incomplete bibliography

R. Alur and Th. Henzinger.
A really temporal logic.

H. Comon and Y. Jurski.
Multiple counters automata, safety analysis and Presburger analysis.

A. Finkel and Ph. Schnoebelen.
Well-structured transitions systems everywhere!

O. Ibarra.
Reversal-bounded multicounter machines and their decision problems.

J. Ouaknine and J. Worrell.
On Metric temporal logic and faulty Turing machines.

Ph. Schnoebelen.
Verifying lossy channel systems has nonprimitive recursive complexity.