Decidable Problems for Counter Systems

Day 3
Vector Addition Systems

Stéphane Demri
demri@lsv.ens-cachan.fr

LSV, ENS Cachan, CNRS, INRIA

ESSLLI 2010, Copenhagen, August 2010
Plan of the talk

- Previous lectures:
  - Classes of counter systems, Presburger arithmetic.
  - LTL-like dialects for counter systems.

- VASS and FO2 over data words

- Relationships between VASS, VAS and Petri nets

- (Coverability graphs in a nutshell)

- Covering problem in $\text{ESPACEx}$
Recapitulation about VASS

- VASS is a counter system with transitions of the form \( q \xrightarrow{\vec{b}} q' \) with \( \vec{b} \in \mathbb{Z}^n \), which is a shortcut for
\[
\bigwedge_{i \in [1,n]} x'_i = x_i + \vec{b}(i)
\]

- VAS = VASS with a unique control state.
VASS and a Logic on Data Words
Data words
[Bouyer, IPL 02]

- Timed word

\[
\begin{array}{cccccc}
  a & b & c & a & a & b \\
  0 & 0.3 & 1 & 2.3 & 3.5 & 3.51 \\
\end{array}
\]

- Runs from counter systems

\[
\begin{array}{ccccccc}
  q_0 & q_2 & q_3 & q_2 & q_3 & q_2 \\
  0 & 0 & 1 & 2 & 3 & 4 \\
\end{array}
\]

- Integer arrays

\[
\begin{array}{cccccc}
\end{array}
\]

- Extension to data trees (XML documents with values).
Specifying classes of data words

- Register automata
  - Register automata
  - Data automata
  - See the survey
  - Class automata
    [Kaminski & Francez, TCS 94]
    [Bouyer & Petit & Thérien, IC 03]
    [Segoufin, CSL 06]
    [Bojańczyk & Lasota, LICS’10]

- First-order languages
  [Bojańczyk et al., LICS 06]

- Temporal logics
  - Real-time logic TPTL
  - LTL with registers
    [Alur & Henzinger, JACM 94]
    [Demri & Lazić & Nowak, TIME 05]

- Many other formalisms
  - Rewriting systems with data
  - Hybrid logics
    [Bouajjani et al., FCT 07]
    [Schwentick & Weber, STACS 07]
  ...
First-order logic on data words

- Data word: nonempty finite sequence of pairs from $\Sigma \times \mathbb{N}$.

- Variable valuation $v$ for a model $\sigma$: map from $\text{VAR}'$ to the positions of $\sigma$.

- Variables are interpreted as positions.

- Formulae of the logic $\text{FO}^\Sigma(\sim, <, +1)$ ($\Sigma$ is a finite alphabet)

\[
\varphi ::= a(x) \mid x \sim y \mid x < y \mid x = y + 1 \mid \neg \varphi \mid \varphi \land \varphi \mid \exists x \varphi
\]

- Last position is labelled by the letter $a \in \Sigma$:

\[
\exists x (\neg \exists y x < y) \land a(x)
\]
Data words as first-order structures

- Alphabet $\Sigma = \{a_1, \ldots, a_N\}$ and infinite domain $\mathbb{N}$.

- Data word $\sigma = (a_{i_1}, d_1) \cdots (a_{i_K}, d_K)$ is equivalent to 
  $$(\{1, \ldots, K\}, <, \sim, +1, P_1, \ldots, P_N)$$

- For $j, j' \in \{1, \ldots, K\}$, $j \sim j'$ iff $d_j = d_{j'}$.

- For $l \in \{1, \ldots, N\}$, $P_l \overset{\text{def}}{=} \{j \in \{1, \ldots, K\} : a_j = a_l\}$.

- First-order logic can be naturally interpreted over such structures.
Semantics

\[ \sigma \models_v a(x) \iff \Sigma(x) = a \]

\[ \sigma \models_v x \sim y \iff \mathbb{N}(x) = \mathbb{N}(y) \]

\[ \sigma \models_v x < y \iff v(x) < v(y) \]

\[ \sigma \models_v x = y + 1 \iff v(x) = v(y) + 1 \]

\[ \sigma \models_v \exists x \varphi \iff \text{there is position } i \text{ s.t. } \sigma \models_v[x \mapsto i] \varphi. \]

- Satisfiability for \( \text{FO}(\sim, <, +1) \) restricted to three individual variables is undecidable [Bojańczyk et al., LICS 06].
Main decidability result

- **Theorem:** Satisfiability problem for $\text{FO}_2(\sim, <, +1)$ is decidable. [Bojańczyk et al., LICS 06].

- **Proof in two steps:**
  - Satisfiability is first reduced to nonemptiness for data automata (not defined in this course).
  - Nonemptiness for data automata is then reduced to the reachability problem for VASS.

- **Theorem:** There is a polynomial-space reduction from the reachability problem for VASS to finitary satisfiability for $\text{FO}_2(\sim, <, +1)$. 
Simplifying the instance

- Transitions can be restricted to increments or decrements of a single counter.

- Translation \( \left( \begin{array}{c} 2 \\ -3 \end{array} \right) \) can be encoded by 2 increments of the first counter followed by 3 decrements of the second counter.

- Initial and final configurations have all the counters equal to zero.

- \((q_0, \begin{array}{c} 2 \\ 1 \end{array})\) and \((q_f, \begin{array}{c} 1 \\ 0 \end{array})\) are reduced to \((q'_0, \begin{array}{c} 0 \\ 0 \end{array})\) and \((q'_f, \begin{array}{c} 0 \\ 0 \end{array})\) by adding the transitions

\[
\begin{array}{cccccc}
q'_0 & \xrightarrow{\text{inc}(1)} & q_0 & \xrightarrow{\text{inc}(1)} & q_0 & \xrightarrow{\text{inc}(2)} & q_0 \\
q_f & \xrightarrow{\text{dec}(1)} & q_f & \xrightarrow{\text{dec}(1)} & q'_f & \xrightarrow{\text{dec}(2)} & q'_f
\end{array}
\]

- All these reductions require only polynomial space.
Fixing a few more things (proof)

- Instance: $S = (Q, n, \delta), (q_i, \vec{0}), (q_f, \vec{0})$.

- $\Sigma = Q \uplus \{\text{inc}(i), \text{dec}(i) : i \in [1, n]\}$.
  (below $a \in \{\text{inc}(i), \text{dec}(i) : i \in [1, n]\}$)

- The run $(q_0, \vec{x}_0) \xrightarrow{a_0} (q_1, \vec{x}_1) \xrightarrow{a_1} \cdots \xrightarrow{a_{K-1}} (q_K, \vec{x}_K)$ encoded by a data word with projection $q_0 a_0 q_1 a_1 \cdots a_{K-1} q_K$.

- Run

  $q_0 \quad q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6$

  $\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 2 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

  corresponds to data word

  $q_0 \quad \text{inc}(1) \quad q_1 \quad \text{inc}(1) \quad q_2 \quad \text{inc}(2) \quad q_3 \quad \text{dec}(1) \quad q_4 \quad \text{dec}(1) \quad q_5 \quad \text{dec}(2) \quad q_6$

  $\ast \quad k_1 \quad \ast \quad k_2 \quad \ast \quad k_3 \quad \ast \quad k_1 \quad \ast \quad k_2 \quad \ast \quad k_3 \quad \ast$
Enforcing the projection on finite $\Sigma$

- $\varphi_{proj}$: conjunction of the formulae below.
  - The first letter is $q_i$:
    \[
    \exists x (\neg \exists y \; y < x) \land q_i(x)
    \]
  - The last letter is $q_f$:
    \[
    \exists x (\neg \exists y \; x < y) \land q_f(x)
    \]
  - Sequence of locations/actions respects the control graph:
    \[
    \forall x \left( \bigvee_{q \in Q} q(x) \right) \Rightarrow \left( (\neg \exists y \; x < y) \lor \bigvee_{q \in Q} q(x) \land (\exists y \; y = x + 1 \land a(y)) \land \right.
    \]
    \[
    \bigvee_{q \xrightarrow{a} q' \in \delta} \left( \exists y \; y = x + 1 \land (\exists x \; x = y + 1 \land q'(x)) \right)
    \]
  - Observe the nice (and standard) recycling of variables.
Constraints on data values

- To encode counter values, each increment or decrement is attached to a datum.

- A desirable data word:

  \[
  q_0 \quad \text{inc}(1) \quad q_1 \quad \text{inc}(1) \quad q_2 \quad \text{inc}(2) \quad q_3 \quad \text{dec}(1) \quad q_4 \quad \text{dec}(1) \quad q_5 \quad \text{dec}(2) \quad q_6
  \]

  \[
  \star \quad k_1 \quad \star \quad k_2 \quad \star \quad k_3 \quad \star \quad k_1 \quad \star \quad k_2 \quad \star \quad k_3 \quad \star
  \]

- \( \varphi \): conjunction of \( \varphi_{proj} \) and formulae below.

  - For \( i, j \in [1, n] \), there are no two positions labelled by \( \text{inc}(i) \) and \( \text{inc}(j) \) having the same datum:

    \[
    \forall x \ y \ (x < y \land \text{inc}(i)(x) \land \text{inc}(j)(y)) \Rightarrow \neg(x \sim y).
    \]

    (recall that \( \text{inc}(i) \) and \( \text{dec}(i) \) are letters in \( \Sigma \))

  - Idem for \( \text{dec}(i) \) and \( \text{dec}(j) \):

    \[
    \forall x \ y \ (x < y \land \text{dec}(i)(x) \land \text{dec}(j)(y)) \Rightarrow \neg(x \sim y).
    \]
Constraints on data values (II)

• For $i \in [1, n]$, for every position labelled by $\text{dec}(i)$, there is a past position labelled by $\text{inc}(i)$ with the same data value:

$$\forall x \ \text{dec}(i)(x) \Rightarrow (\exists y \ y < x \land x \sim y \land \text{inc}(i)(y))$$

• In the final configuration, any counter value is zero. So, for $i \in [1, n]$, for every position labelled by $\text{inc}(i)$, there is a future position labelled by $\text{dec}(i)$ with same data value:

$$\forall x \ \text{inc}(i)(x) \Rightarrow (\exists y \ x < y \land x \sim y \land \text{dec}(i)(y))$$

• One can show $(q_f, 0)$ is reachable from $(q_i, 0)$ iff $\varphi$ is satisfiable.
Petri nets and VASS
A few definitions on Petri nets

Petri net $N = (S, T, W, m_I)$

- finite set of places $S$,
- finite set of transitions $T$,
- weight function $W : (S \times T) \cup (T \times S) \rightarrow \mathbb{N}$,
- initial marking $m_I : S \rightarrow \mathbb{N}$.

(marking $m : S \rightarrow \mathbb{N}$, specifying the nb. of tokens by place)

$S = \{p_A, p_B, p_C, p_1\}$

$W(p_A, t_1) = 1$

$W(t_1, p_A) = 0$
Reachable markings

- Transition $t \in T$ is $m$-enabled, whenever for all places $p \in S$, $m(p) \geq W(p, t)$.

- An $m$-enabled transition $t$ may fire and produce the marking $m'$, written $m \xrightarrow{t} m'$, with for all places $p \in S$,
  \[ m'(p) = m(p) - W(p, t) + W(t, p) \]

- Marking $m'$ is reachable from $m$ whenever there is a sequence of the form
  \[ m_0 \xrightarrow{t_0} m_1 \xrightarrow{t_1} \cdots \xrightarrow{t_{k-1}} m_k \]
  with $m_0 = m$ and $m_k = m'$ (also written $m \xrightarrow{t_0 \cdots t_{k-1}} m'$).
Problems on Petri nets

• Reachability problem for Petri nets:
  
  **Input:** a Petri net \((S, T, W, m_I)\) and a marking \(m\).
  
  **Question:** is \(m\) reachable from \(m_I\)?

• Covering problem for Petri nets:
  
  **Input:** a Petri net \((S, T, W, m_I)\) and a marking \(m\).
  
  **Question:** is there a marking \(m'\) reachable from \(m_I\) such that for all \(p \in S\), we have \(m'(p) \geq m(p)\)?

• Boundedness problem for Petri nets:
  
  **Input:** a Petri net \((S, T, W, m_I)\).
  
  **Question:** is the set of markings reachable from \(m_I\) infinite?
Questions

- Is \((0, 0, 1, 1000)\) reachable from \((1, 0, 0, 0)\) (with implicit ordering of the places \(p_A, p_B, p_C, p_1\))?
- Is \((1, 0, 1, 2)\) reachable from \((1, 0, 0, 0)\)?
- Is the Petri net with initial marking \((1, 0, 0, 0)\) bounded?
- Is there some marking \(m\) reachable from \((1, 0, 0, 0)\) such that \((1, 0, 0, 1000) \preceq m\)?
From VASS to Petri nets
Systematic construction of Petri nets

- VASS $\mathcal{V} = (Q, n, \delta) + \text{configuration } (q_I, \vec{x}_I)$.

- The corresponding Petri net $N_{\mathcal{V}}$:
  - For $q \in Q$, we introduce a place $p_q$.
  - For $i \in [1, n]$, we introduce a place $p_i$.
  - For $q \xrightarrow{\vec{b}} q' \in \delta$, we consider a transition $t$ that consumes a token in $p_q$, produces a token in $p_{q'}$ and produces [resp. consumes] $\vec{b}(i)$ tokens in the place $p_i$ when $\vec{b}(i) \geq 0$ [resp. when $\vec{b}(i) < 0$].
  - Initial marking $m_I$ contains one token in the place $p_{q_I}$ and for $i \in [1, n]$, $m_I(p_i) = \vec{x}_I(i)$. 
For all \((q, \vec{x})\), we have the equivalence

- \((q, \vec{x})\) is reachable from \((q_I, \vec{x}_I)\),
- \(m\) is reachable from \(m_I\) where for \(q' \in Q \setminus \{q\}\), \(m(p_{q'}) = 0\), \(m(p_q) = 1\) and for \(i \in [1, n]\), \(m(p_i) = \vec{x}(i)\).

Boundedness, covering and reachability problems for VASS can be reduced to the analogous problems for Petri nets.
From Petri nets to VASS
Systematic construction of VASS

- Petri net \((S, T, W, m_I)\) with (arbitrary) bijection \(f : \{1, \ldots, \text{card}(S)\} \rightarrow S\).

- The corresponding VASS \(\nu = (Q, n, \delta)\):
  - \(Q = \{1\} \cup T\),
  - \(n = \text{card}(S)\),
  - for \(t \in T\), we consider two transitions in \(\delta\):
    \[
    1 \xrightarrow{b^-} t \quad t \xrightarrow{b^+} 1
    \]
    such that for \(i \in [1, n]\),
    \[
    b^-(i) = -W(f(i), t) \quad b^+(i) = W(t, f(i))
    \]
  - Initial configuration \((q_I, \vec{x}_I)\) with \(q_I = 1\) and for \(i \in [1, n]\),
    \[
    \vec{x}_I(i) = m_I(f(i)).
    \]
Reductions (bis)

• For all markings $m$, we have the equivalence:
  
  • $m$ is reachable from $m_I$,
  
  • there is a run from $(q_I, \vec{x}_I)$ to $(q, \vec{x})$ where
    • $q = 1$,
    • for $i \in [1, n]$, $\vec{x}(i) = m(f(i))$.

• Boundedness, covering and reachability problems for Petri nets can be reduced to the analogous problems for VASS.

Petri nets and VASS are equivalent models for many decision problems.
From VASS to VAS
(other direction is obvious)

$$\begin{align*}
A & \quad (x_1, A, B, C) \\
+1 & \quad (1, -1, 1, 0) \\
0 & \quad (0, 1, -1, 0) \\
B & \quad (-1, 0, -1, 1) \\
-1 & \quad (0, 0, 1, -1) \\
C & \\
0 & \quad (A, 4) \approx (4, 1, 0, 0) \text{ and } (C, 2) \approx (2, 0, 0, 1) \\
\end{align*}$$

Reduction is correct from VASS without self-loops
Reduction

• W.l.o.g., $\mathcal{V}$ has no transition of the form $q \xrightarrow{\vec{b}} q$. Otherwise, replace $q \xrightarrow{\vec{b}} q$ by $q \xrightarrow{\vec{0}} q_{\text{new}}$ and $q_{\text{new}} \xrightarrow{\vec{b}} q$.

• Boundedness, covering and reachability problems for VASS are equivalent to the analogous problems for VASS without self-loops.

• The VAS $\mathcal{T}$ built from the VASS $\mathcal{V} = (Q, n, \delta)$ has dimension $n + \text{card}(Q)$. Control states are encoded in the $\text{card}(Q)$ last components.

• Control state $q_k$ is encoded by the unit tuple $e_k$ on the $\text{card}(Q)$ last components.
Reduction (II)

- Each transition $q \xrightarrow{\vec{b}} q' \in \delta$ provides $t' \in T$:
  - $(t')_{[1,n]} = \vec{b}$; for $q'' \in Q \setminus \{q, q'\}$, $t'(h(q'')) = 0$,
  - $t'(h(q)) = -1$ and $t'(h(q')) = 1$.

  where $h : Q \rightarrow [n + 1, n + \text{card}(Q)]$ is an arbitrary bijection.

- Boundedness, covering and reachability problems for VASS can be reduced to the analogous problems for VAS.

- Alternative reduction from VASS of dimension $n$ to VAS of dimension $n + 3$ (instead of $n + \text{card}(Q)$).

  [Hopcroft & Pansiot, TCS 79]
Petri nets, VASS and VAS are equivalent models for many decision problems.
Solving the covering problem for VAS
About the covering problem for VAS

- **COVERING PROBLEM:**
  
  **Input:** a VAS $\mathcal{T}$ and two configurations $\vec{x}, \vec{x}' \in \mathbb{N}^n$,
  
  **Question:** is there some configuration $\vec{x}''$ reachable from $\vec{x}$ such that $\vec{x}' \preceq \vec{x}''$?

- The control state reachability problem for VASS can be reduced to the covering problem for VAS: require that one component has at least value 1.
  (reaching the control state $A$ in VASS is equivalent to cover $(0, 1, 0, 0)$ in corresponding VAS)

- The covering problem for VAS is $\text{EXPSPACE}$-complete:
  - Decidability with nonprimitive recursive complexity.
  - $\text{EXPSPACE}$ lower bound from [Lipton, TR 76].
  - $\text{EXPSPACE}$ upper bound from [Rackoff, TCS 78].
Part I: Coverability Graph
Coverability graphs in a nutshell

- Finite graph whose set of nodes is a finite subset of \( (\mathbb{N} \cup \{\infty\})^n \) that can be effectively computed.

- It approximates the set of reachable configurations.

- Simple properties on it allow to solve various problems: boundedness, covering, termination, etc.

- ...but in the worst-case, its size can be nonprimitive recursive.

- First, we need to define relations and operations on \( (\mathbb{N} \cup \{\infty\})^n \).
A digression on a variant of Ackermann function

- $A_0(m) = 2m + 1$, $A_{n+1}(0) = 1$.

- $A_{n+1}(m + 1) = A_n(A_{n+1}(m))$.

- $A(n) = A_n(2)$.

- The function $A(n)$ majorizes the primitive recursive functions.

- The size of the coverability graph can be in $O(A(n))$. ($n$: size of $T$ and $x_0$)
How to calculate with $\infty$?

- For $k, k' \in \mathbb{N} \cup \{\infty\}$,
  \[
  k \leq k' \overset{\text{def}}{\iff} \text{either } k, k' \in \mathbb{N} \text{ and } k \leq k' \text{ or } k' = \infty.
  \]

- $k < k'$ whenever $k \leq k'$ and $k \neq k'$.

- $(\mathbb{N} \cup \{\infty\}, <)$ is isomorphic to the ordinal $\omega + 1$.

- $\leq$ and $<$ are extended component-wise to $(\mathbb{N} \cup \{\infty\})^n$.

- \[
  \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} < \begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix} \rightarrow acc\left(\begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix}\right) \overset{\text{def}}{=} \begin{pmatrix} 2 \\ \infty \\ 1 \end{pmatrix}.
  \]

- For $\vec{x} < \vec{x}'$, let us define $acc(\vec{x}, \vec{x}') \in (\mathbb{N} \cup \{\infty\})^n$:
  - $acc(\vec{x}, \vec{x}')(i) \overset{\text{def}}{=} \vec{x}'(i)$ when $\vec{x}(i) = \vec{x}'(i)$,
  - $acc(\vec{x}, \vec{x}')(i) \overset{\text{def}}{=} \infty$ when $\vec{x}(i) < \vec{x}'(i)$.

  “The $i$th component can be as large as we wish.”
How to calculate with $\infty$? (II)

- Given $\vec{x} \in (\mathbb{N} \cup \{\infty\})^n$ and $t \in \mathbb{Z}^n$, let us define $\vec{x} + t \in (\mathbb{Z} \cup \{\infty\})^n$:
  - $(\vec{x} + t)(i) \stackrel{\text{def}}{=} \vec{x}(i) + t(i)$ if $\vec{x}(i) \in \mathbb{N}$,
  - $(\vec{x} + t)(i) \equiv \infty$ otherwise.

$(i \in [1, n])$

- $\left(\begin{array}{c} 2 \\ \infty \end{array}\right) + \left(\begin{array}{c} -3 \\ -6 \end{array}\right) = \left(\begin{array}{c} -1 \\ \infty \end{array}\right)$.

- The construction of the coverability graph $CG(T, \vec{x}_0)$ uses these operations on $\mathbb{N} \cup \{\infty\}$.
Example

\[(t_1) + 1 \rightarrow 0 (t_2) \]

\[(t_3) - 1 \rightarrow 0 (t_4) \]
Properties of $CG(\mathcal{T}, \vec{x}_0)$

- Given $\mathcal{T}$ and $\vec{x}_0$, coverability graph $CG(\mathcal{T}, \vec{x}_0)$ is a structure $(V, E)$ with $V \subseteq (\mathbb{N} \cup \{\infty\})^n$ and $E \subseteq V \times \mathcal{T} \times V$.

(a) $CG(\mathcal{T}, \vec{x}_0)$ is a finite structure with “root” $\vec{x}_0$.

(b) Every configuration reachable from $\vec{x}_0$ can be covered in $CG(\mathcal{T}, \vec{x}_0)$, i.e.
   - for $\vec{y}$ reachable from $\vec{x}_0$, there is $\vec{y}'$ in $CG(\mathcal{T}, \vec{x}_0)$ such that $\vec{y} \preceq \vec{y}'$.

(c) For every $\vec{y}$ in $CG(\mathcal{T}, \vec{x}_0)$ and bound $B \in \mathbb{N}$, there is a configuration $\vec{y}'$ reachable from $\vec{x}_0$ s.t.
   - $\vec{y}(i) = \infty$ implies $\vec{y}'(i) \geq B$,
   - $\vec{y}(i) \neq \infty$ implies $\vec{y}'(i) = \vec{y}(i)$.
   ($i \in [1, n]$)
A quick presentation of the construction

\[ E := \emptyset; \ V := \emptyset; \ ToBeTreated := \{\vec{x}_0\}; \]
while \( ToBeTreated \neq \emptyset \) do
  • Select an element \( \vec{x} \) from \( ToBeTreated \);
  • \( ToBeTreated := ToBeTreated \setminus \{\vec{x}\} \);
  • for \( t \in T \) such that \( \vec{x} + t \in (\mathbb{N} \cup \{\infty\})^n \) do
    • \( \vec{x}^' := \vec{x} + t \);
    • if there is \( \vec{y} \in V \) s.t. \( \vec{y} \xrightarrow{*} \vec{x} \) in \( (V, E) \) and \( \vec{y} < \vec{x}^' \) then
      • Let \( \vec{y}_0 \) be the extended configuration the closest to \( \vec{x} \) in \( (V, E) \) such that \( \vec{y}_0 < \vec{x}^' \);
      • \( \vec{x}^' := acc(\vec{y}_0, \vec{x}^') \);
    • if \( \vec{x}^' \notin V \) then
      • \( V := V \cup \{\vec{x}^'\} \);
      • \( ToBeTreated := ToBeTreated \cup \{\vec{x}^'\} \);
  • \( E := E \cup \{\vec{x} \xrightarrow{t} \vec{x}^'\} \);
Characterizations with coverability graph

• There is $x''$ reachable from $x_0$ s.t. $x' \preceq x''$ iff there is $\bar{y}$ in $CG(T, x_0)$ s.t. $x' \preceq \bar{y}$.

• The set of configurations reachable from $x_0$ is infinite iff $\infty$ appears in $CG(T, x_0)$.

• Every run from $x_0$ terminates iff there is no cycle in $CG(T, x_0)$.
There is $\vec{x''}$ reachable from $\vec{x}_0$ s.t. $\vec{x'} \preceq \vec{x''}$ iff there is $\vec{y}$ in $CG(\mathcal{T}, \vec{x}_0)$ s.t. $\vec{x'} \preceq \vec{y}$.

- Suppose that $\vec{x''}$ reachable from $\vec{x}_0$ and $\vec{x'} \preceq \vec{x''}$.
  - By (b), there is $\vec{y}$ in $CG(\mathcal{T}, \vec{x}_0)$ s.t. $\vec{x''} \preceq \vec{y}$.
  - Since $\preceq$ is transitive on $(\mathbb{N} \cup \{\infty\})^n$, $\vec{x'} \preceq \vec{y}$.

- Suppose that there is $\vec{y}$ in $CG(\mathcal{T}, \vec{x}_0)$ such that $\vec{x'} \preceq \vec{y}$.
  - $B$: maximal value occurring in $\vec{x'}$.
  - By (c), there is $\vec{y'}$ reachable from $\vec{x}_0$ such that for $i \in [1, n]$, if $\vec{y}(i) = \infty$ then $\vec{y'}(i) \geq B$ otherwise $\vec{y'}(i) = \vec{y}(i)$.
  - Hence, $\vec{x'} \preceq \vec{y'}$. 
The set of configurations reachable from $\vec{x}_0$ is infinite iff $\infty$ appears in $CG(\mathcal{T}, \vec{x}_0)$.

- Suppose the set of configurations reachable from $\vec{x}_0$ is infinite.
  - *Ad absurdum*, assume that $\infty$ does not occur in $CG(\mathcal{T}, \vec{x}_0)$.
  - By (b), there is $\vec{y}$ in $CG(\mathcal{T}, \vec{x}_0)$ s.t. for an infinite amount of configurations $\vec{x}$ reachable from $\vec{x}_0$, we have $\vec{x} \preceq \vec{y}$.
  - There are at most $(1 + \max(\vec{y}))^n$ distinct configurations smaller than $\vec{y} \in \mathbb{N}^n$, contradiction.

- Suppose $\infty$ occurs in $CG(\mathcal{T}, \vec{x}_0)$.
  - By (c) the set of configurations reachable from $\vec{x}_0$ is infinite.
  - For instance, consider bounds $B$ greater and greater when applying (c).
Part II: Exponential-space decision procedure
Small covering property

- VAS $T$ with configurations $\vec{x}$, $\vec{x}'$. Equivalence between
  - there is a run from $\vec{x}$ leading to $\vec{y}$ such that $\vec{x}' \preceq \vec{y}$,
  - there is a run from $\vec{x}$ leading to $\vec{y}'$ such that $\vec{x}' \preceq \vec{y}'$ and its length is at most double-exponential in the size of the instance $T$, $\vec{x}$ and $\vec{x}'$ (numbers in binary).

- A run of double-exponential length requires double-exponential space to be fully encoded.

- In the worst-case, there is a triple-exponential amount of such runs.

- Solution: guess nondeterministically the small run and invoke Savitch’s theorem.
Example of small covering

How to cover \((A, (1, K))\) from \((A, (0, 0))\)?

Long covering: \((A, (0, 0)) \xrightarrow{(t_1 t_2)^{2^K}} (B, (0, 2^K + 1)) \xrightarrow{t_3 t_4 t_2} (A, (1, 2^K)) \succeq (A, (1, K))\)

Short covering: \((A, (0, 0)) \xrightarrow{(t_1 t_2)^K t_1} (B, (0, K + 1)) \xrightarrow{t_3 t_4 t_2} (A, (1, K)) \succeq (A, (1, K))\)

\((t_1 t_2)^K t_1 t_3 t_4 t_2\) subword of \((t_1 t_2)^{2^K} t_1 t_3 t_4 t_2\)
How to be clever enough to guarantee a “short” covering?
Nondeterministic algorithm

- Algorithm for $\mathcal{T}, \vec{x}, \vec{x}'$ and $L$:
  1. $i := 0; \vec{x}_c := \vec{x}$ (current configuration);
  2. While $\vec{x}' \not\leq \vec{x}_c$ and $i < L$ do
     1. Guess a transition $t \in \mathcal{T}$; (nondeterministic step !)
     2. If $\vec{x}_c + t \not\in \mathbb{N}^n$ then abort;
     3. $i := i + 1; \vec{x}_c := \vec{x}_c + t.$
  3. If $\vec{x}' \leq \vec{x}_c$ then accept else abort ($i = L$).

- If the maximal absolute value in $\mathcal{T}, \vec{x}, \vec{x}'$ is $2^N$ and $L = 2^{2N^3}$, then the maximal absolute value appearing in the algorithm is $2^N + 2^N \times 2^{2N^3}$ (can be encoded with exponential space in $N$).

- Determinism can be regained with recursive calls to a function $F(\mathcal{T}, \vec{x}, \vec{x}', L)$ since the number of transitions is finite.
Design a decision procedure that nondeterministically guesses the small run and only requires exponential space:

- A counter with an exponential amount of bits can count until a double-exponential value.
- Only two configurations need to be store thanks to nondeterminism.
- $2^{2^N} \times 2^N$ is still of double-exponential magnitude.
- Comparing or adding two natural numbers requires logarithmic space only.
- [Savitch, JCSS 70]: a nondeterministic procedure for a given problem using space $f(N) \geq \log(N)$ can be turned into a deterministic procedure using $f(N) \times f(N)$ space.
- Exponential functions are closed under multiplication.
Further readings

- The proof for $\text{EXPSPACE}$ upper bound for boundedness is a bit more complex [Rackoff, TCS 78].

- The proof for $\text{EXPSPACE}$-hardness of covering, boundedness and reachability problems is nicely explained in [Esparza, 98] (based on [Lipton, TR 76]).


- Nice hints about the proof can be found in [Haddad, 01].
Definitions about sizes

- For $\vec{x} \in \mathbb{Z}^n$, 
  
  $\text{maxneg}(\vec{x}) \overset{\text{def}}{=} \max(\{\max(0, -\vec{x}(i)) : i \in [1, n]\})$: maximal absolute negative value.

- For example, $\text{maxneg}(\begin{pmatrix} -1 \\ -2 \\ -8 \\ 7 \end{pmatrix}) = \max(0, -(−8)) = 8$.

- $\text{max}(\vec{x}) \overset{\text{def}}{=} \max(\{\vec{x}(i) : i \in [1, n]\})$: maximal value.

- For instance, $\text{max}(\begin{pmatrix} -1 \\ -2 \\ -8 \\ 7 \end{pmatrix}) = 7$.

- $\text{scale}(T) \overset{\text{def}}{=} \max(\{|t(i)| : t \in T, i \in [1, n]\})$: maximal absolute value.

- For instance, $\text{scale}(\begin{pmatrix} -1 \\ -2 \\ -8 \\ 7 \end{pmatrix}) = |-8| = 8$. 


Definitions about sizes (II)
(or reasonably succinct encodings)

• $2 + \lceil \log_2(1 + K) \rceil$ is a sufficient number of bits to encode integers in $[-K, K]$ for $K > 0$.

• Size $|\mathcal{T}| \overset{\text{def}}{=} n \times \text{card}(\mathcal{T}) \times (2 + \lceil \log_2(1 + \text{scale}(\mathcal{T})) \rceil)$.

• Given $N = |\mathcal{T}| + |\{\vec{x}\}| + |\{\vec{x}'\}|$, we have $\maxneg(\mathcal{T}), \text{card}(\mathcal{T}), \max(\vec{x}') \leq 2^N$. 
Paths and pseudo-runs

- **Path** $\pi$: finite sequence of transitions (below in blue).

$$
\begin{pmatrix}
1 & 0 \\
0 & 1 \\
0 & 0 \\
\end{pmatrix} \xrightarrow{egin{pmatrix}
1 & -1 \\
0 & 1 \\
1 & 0 \\
\end{pmatrix}} \begin{pmatrix}
0 & 1 \\
1 & 0 \\
0 & 1 \\
\end{pmatrix} \xrightarrow{egin{pmatrix}
1 & 0 \\
0 & 1 \\
1 & 0 \\
\end{pmatrix}} \begin{pmatrix}
0 & 1 \\
1 & 0 \\
2 & 0 \\
\end{pmatrix} \xrightarrow{egin{pmatrix}
1 & 0 \\
0 & 1 \\
0 & 1 \\
\end{pmatrix}} \begin{pmatrix}
1 & 0 \\
0 & 1 \\
2 & 0 \\
\end{pmatrix}
$$

- $\pi'$ is a subpath of $\pi = t_1 \ldots t_k \overset{\text{def}}{\iff}$ there are $1 \leq j_1 < j_2 \ldots < j_{k'} \leq k$ s.t. $\pi' = t_{j_1} \ldots t_{j_{k'}}$.

- **Pseudo-configuration** $\vec{x} \in \mathbb{Z}^n$.

- **Given** $\pi = t_1 \ldots t_k$ and $\vec{x} \in \mathbb{Z}^n$, pseudo-run $(\pi, \vec{x}) = \vec{x}_0 \cdots \vec{x}_k$ s.t. $\vec{x}_0 = \vec{x}$ and for $i \in [1, k]$, $\vec{x}_i = \vec{x}_{i-1} + t_i$.

- The length of $(\pi, \vec{x})$ [resp. $\pi$] is $k + 1$ [resp. $k$].

- A pseudo-run $\vec{x}_0 \cdots \vec{x}_k$ is a covering of $\vec{x}'$ when $\vec{x}' \preceq \vec{x}_k$. 
Defining length of shortest coverings

- Assume that \((\pi, \vec{x})\) is a run covering \(\vec{x}'\),
  
  \[ m(\mathcal{T}, \vec{x}, \vec{x}', \pi) \stackrel{\text{def}}{=} \text{the length of the shortest subpath } \pi' \text{ of } \pi \]
  
  s.t. \((\pi', \vec{x})\) is a run covering \(\vec{x}'\).

- \(M_B(n) (B, n \geq 1)\): be the supremum of the set below

  \[ \left\{ m(\mathcal{T}, \vec{y}, \vec{y}', \pi) : (\pi, \vec{y}) \text{ is a run covering } \vec{y}' \right. \]
  
  \[ \mathcal{T} \text{ is a VAS of dimension } n \text{ and } \maxneg(\mathcal{T}) + \max(\vec{y}') \leq B \}

- We shall establish these inequalities for \(n, B \geq 1\):

  \[ M_B(n) \leq \begin{cases} 
    B & \text{if } n = 1, \\
    (B \cdot M_B(n - 1))^n + M_B(n - 1) & \text{if } n \geq 2. 
  \end{cases} \]
Towards the rough bound $2^{2N^3}$

- For $n \geq 1$ and $B \geq 2$, $M_B(n) \leq B^{3n!}$ ($M_B(1) \leq B$):

  $$M_B(n) \leq (B \cdot M_B(n-1))^n + M_B(n-1) \leq (B \cdot M_B(n-1))^{n+1} \leq \ldots \leq (B^{1+(3(n-1))!})^{n+1} \leq B^{(3n)!}$$

- Existence of a covering of $\vec{x}'$ from $\vec{x}$ in $\mathcal{T}$ is equivalent to the existence of a covering of length at most

  $$\alpha = (\max_{\text{neg}}(\mathcal{T}) + \max(\vec{x}')) + 2)^{(3n)!}$$

- With $N = |\mathcal{T}| + |\{\vec{x}\}| + |\{\vec{x}'\}|$,

  $$\alpha \leq (2^N + 2^N + 2)^{2N \log_2(N)} \leq (2^N+2)^{2N^2} \leq 2^{2N^3}.$$  

- The covering problem for VAS, VASS and Petri nets can be solved in exponential space.
Back to inequalities (main proof)

- We show for $n, B \geq 1$:

$$M_B(n) \leq \begin{cases} B & \text{if } n = 1, \\ (B \cdot M_B(n - 1))^n + M_B(n - 1) & \text{if } n \geq 2. \end{cases}$$

- Base case $n = 1$.

  - $\mathcal{T}$ of dim. 1, $\vec{x}, \vec{x}' \in \mathbb{N}$ and $\maxneg(\mathcal{T}) + \max(\vec{x}') \leq B$.

  - $\vec{x}' \preceq \vec{x}$ implies empty path produces a run covering $\vec{x}'$.

  - Otherwise, no need to use negative values from $\mathcal{T}$ and $m(\mathcal{T}, \vec{x}, \vec{x}', \pi)$ is bounded by $\max(\vec{x}')$.

  - $M_B(1) \leq B$ since $\max(\vec{x}') \leq B$. 

Induction step

• Suppose the property holds true for $n - 1 \geq 1$.

• It is sufficient to show:

$$m(T, \vec{x}, \vec{x}', \pi) \leq (B \cdot M_B(n - 1))^n + M_B(n - 1)$$

whenever $\text{maxneg}(T) + \max(\vec{x}') \leq B$ and $T$ of dimension $n$.

• After $M_B(n - 1)$ steps, a component greater than $M_B(n - 1)\text{maxneg}(T) + \max(\vec{x}')$, has value at least $\max(\vec{x}')$.

• $B' = M_B(n - 1)\text{maxneg}(T) + \max(\vec{x}') \leq B M_B(n - 1)$.

• Pseudo-run $\vec{x}_0 \cdots \vec{x}_k$ is $r$-bounded ($r > 0$) $\iff$ for $i \in [0, k]$, we have $\vec{x}_i \in [0, r - 1]^n$. 
Two cases are distinguished

- \((\pi, \vec{x})\): run covering \(\vec{x}'\) for the VAS \(\mathcal{T}\).

- \(\pi = t_1 \cdots t_k\) and \((\pi, \vec{x}) = \vec{x}_0 \cdots \vec{x}_k\).

- We distinguish the case when \((\pi, \vec{x})\) is \(B'\)-bounded or not.
  - If \((\pi, \vec{x})\) is \(B'\)-bounded, then subpaths between identical configurations can be removed (pigeonhole principle).
  - Otherwise, the path is divided in two:
    - the first part is \(B'\)-bounded and can be shortened,
    - shortening the second part can be done by using induction hypothesis.
Case 1: \((\pi, \vec{x})\) is \(B'\)-bounded

- If \(\vec{x}_i = \vec{x}_j\) with \(0 \leq i < j \leq k\) then
  - \((\pi', \vec{x})\) is also a run covering of \(\vec{x}'\),
  - \(\pi' = t_1 \cdots t_i t_{i+1} \cdots t_k\),
  - \(\pi'\) is a strict subpath of \(\pi\).

- This situation occurs as soon as \(k \geq (B')^n\).

- This transformation can be repeated until \(k < (B')^n\). (pigeonhole principle)

- Conclusion: there is a subpath \(\pi'\) s.t. \((\pi', \vec{x})\) is also a run covering \(\vec{x}'\) of length bounded by \((B')^n \leq (BM_B(n-1))^n\).
Case 2: \((\pi, \vec{x})\) is not \(B'\)-bounded

- Unique decomposition \(\pi = \pi_1 \pi_2\) s.t.
  - \(\pi_1\) and \(\pi_2\) of respective length \(k_1\) and \(k_2\),
  - all values in \(\vec{x}_0 \cdots \vec{x}_{k_1-1}\) are < \(B'\).
- \((\pi_1, \vec{x})\) is not \(B'\)-bounded ("faulty" last configuration).

- There is \(\pi'_1\), subpath of \(\pi_1\), such that
  - \(\pi'_1\) is a subpath of \(\pi_1\),
  - its length is bounded by \((BM_B(n-1))^n + 1\),
    (again pigeonhole principle!)
  - \((\pi_1, \vec{x})\) and \((\pi'_1, \vec{x})\) have the same final configuration \(\vec{y}\).
- \((\pi'_1 \pi_2, \vec{x})\) and \((\pi_2, \vec{y})\) are both runs covering \(\vec{x}'\).
Case 2: \((\pi, \vec{x})\) is not \(B'\)-bounded (III)

- Let \(i \in [1, n]\) s.t. \(\vec{y}(i) \geq B'\).

- \(\mathcal{T}^-, \pi_2^-, \vec{y}^-, \vec{x}'^-\): restrictions of \(\mathcal{T}, \pi_2, \vec{y}, \vec{x}'\) to the components in \([1, n] \setminus \{i\}\).
  (dimension is reduced by 1)

- \((\pi_2^-, \vec{y}^-)\) is a run covering \(\vec{x}'^-\) in \(\mathcal{T}^-\).

- \(\max\text{neg}(\mathcal{T}^-) + \max(\vec{x}'^-) \leq B\).

- **By induction hypothesis**, there is \(\pi'_2\), subpath of \(\pi^-_2\) s.t.
  - \((\pi'_2, \vec{y}^-)\) is a run covering \(\vec{x}'^-\),
  - its length is bounded by \(M_B(n - 1)\).

- \(\pi''_2\): path obtained from \(\pi'_2\) by adding the \(i\)th missing component.
Case 2: \((\pi, \vec{x})\) is not \(B'-\)bounded (II)

- \((\pi''_2, \vec{y})\) is a pseudo-run with final pseudo-configuration \(\vec{z}\) such that for \(j \in (\{1, n\} \setminus \{i\})\), \(\vec{z}(j) \geq \vec{x}'(j)\).

- Since \(\vec{y}(i) \geq B'\) and after \(M_B(n - 1)\) steps, the \(i\)th component is greater or equal to \(\max(\vec{x}')\).

- Conclusion: \((\pi''_2, \vec{y})\) is a run covering \(\vec{x}'\).

- Length of \(\pi'_1 \pi''_2\) is at most \((B \times M_B(n - 1))^n + M_B(n - 1)\).

- \(\pi'_1 \pi''_2\) is a subpath of \(\pi\).
Conclusion

• Today’s lecture:
  • Relationships between VAS, VASS, Petri nets (and FO2 over data words).
  • (Coverability graphs for VASS).
  • Covering problem for VAS in \( \text{EXPSPACE} \) by induction on the dimension (Rackoff’s proof).
  • Control state reachability for VASS in \( \text{EXPSPACE} \).

• Tomorrow’s lecture:
  • Definitions on reversal-bounded counter automata.
  • Proof for semilinearity.
  • Variants.
  • Repeated reachability and related logical problems.