Some techniques and results in deciding bisimilarity

Petr Jančar

Dept of Computer Science
Technical University Ostrava (FEI VŠB-TU)
Czech Republic
www.cs.vsb.cz/jancar

Talk at the Verification Seminar, ULB, Brussels, 16 December 2005

Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.

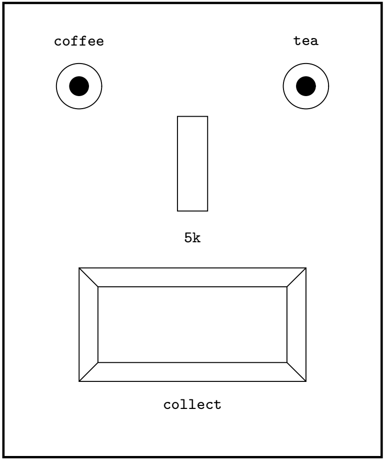
Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.

Vending machines

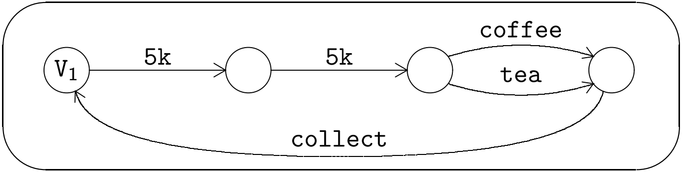
$$\mathtt{V}_1 \stackrel{\text{def}}{=} \mathtt{5k.5k.(\ coffee.collect.V}_1 \mathtt{\ +\ tea.collect.V}_1\mathtt{\ )}$$

$$\mathtt{V}_2 \stackrel{\text{def}}{=} \mathtt{5k.5k.coffee.collect.V}_2 \mathtt{\ +\ 5k.5k.tea.collect.V}_2$$

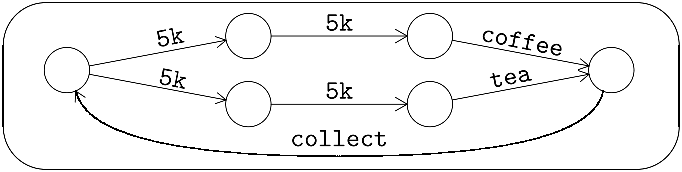


Vending machines - cont.

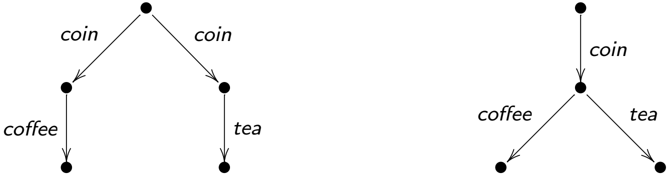
$V_1 \stackrel{\text{def}}{=}$ 5k.5k.(coffee.collect.V_1 + tea.collect.V_1)



$V_2 \stackrel{\text{def}}{=}$ 5k.5k.coffee.collect.V_2 + 5k.5k.tea.collect.V_2



Does a system implement another one ? Are they equivalent ?
(system = labelled transition system)
Language (trace) equivalences often too coarse



A binary relation R over STATES is a simulation if
whenever $(s,t) \in R$ then for every action a

$$\text{if } s \xrightarrow{a} s' \text{ then } t \xrightarrow{a} t' \text{ for some } t' \text{ such that } (s',t') \in R.$$

s is simulated by t if there is a simulation $R \ni (s,t)$.
The union of (all) simulations is (the maximal) simulation

Bisimulation equivalence

Milner, Park (1980s)
A binary relation R over STATES is a bisimulation if (it is a symmetric simulation, i.e.)
whenever $(s,t) \in R$ then for every action a

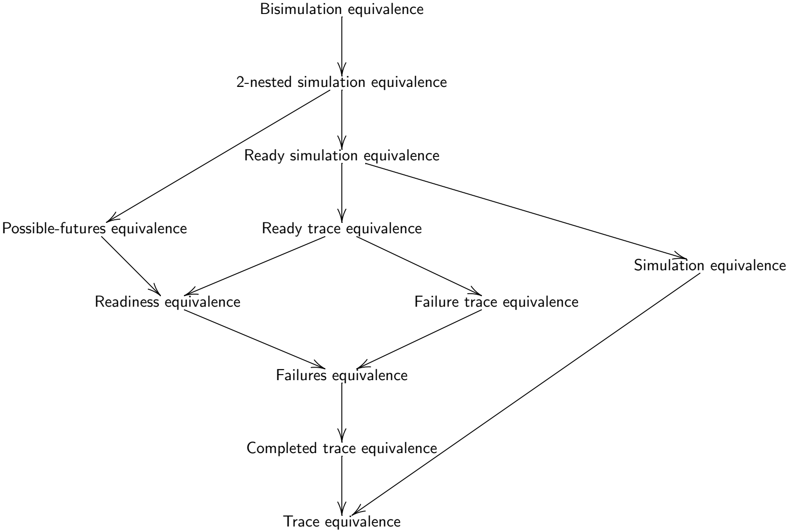
$$\text{if } s \xrightarrow{a} s' \text{ then } t \xrightarrow{a} t' \text{ for some } t' \text{ such that } (s',t') \in R.$$

whenever $(s,t) \in R$ then for every action a

$$\text{if } t \xrightarrow{a} t' \text{ then } s \xrightarrow{a} s' \text{ for some } s' \text{ such that } (s',t') \in R.$$

s is bisimilar with t if there is a bisimulation $R \ni (s,t)$.
The union of (all) bisimulations is (the maximal) bisimulation.

Linear Time / Branching Time Spectrum



Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.

Minsky counter machines

A Minsky counter machine C is given by

- a fixed number of (nonnegative integer) counters $c_1, c_2, \ldots, c_m$
- a program (in fact, a set of labelled instructions)

$1 : \mathrm{COM}_1;\ 2 : \mathrm{COM}_2;\ \ldots\ldots;\ n : \mathrm{COM}_n$, where

  - COM_n is instruction $HALT$
  - COM_i $(i = 1, 2, ..., n-1)$ are commands of two types:

$c_j := c_j + 1;\ goto\ k$

$\textbf{if}\ c_j = 0\ \textbf{then}\ goto\ k_1\ \textbf{else}\ (c_j := c_j - 1;\ goto\ k_2)$

Undecidability of behavioural equivalences for Petri nets

Fact

It is undecidable if a 2-counter machine C halts on the zero input (i.e., when starting with $c_1 = c_2 = 0$).

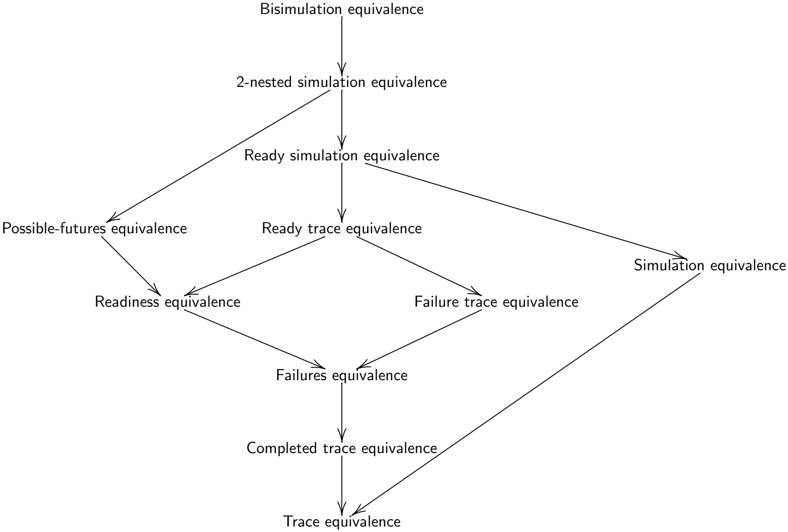
We show an algorithm A:

$$C \longrightarrow \boxed{A} \longrightarrow N_1^C, N_2^C$$

so that

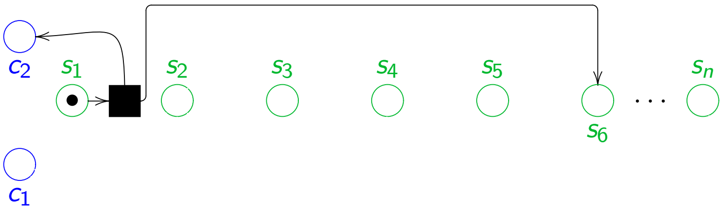
- if C halts (on input zero) then the behaviours of N_1^C, N_2^C differ 'drastically' (one can perform a trace which the other cannot)
- if C does not halt then the behaviours of N_1^C, N_2^C are the same in a strict sense (the nets are bisimilar)

Linear Time / Branching Time Spectrum

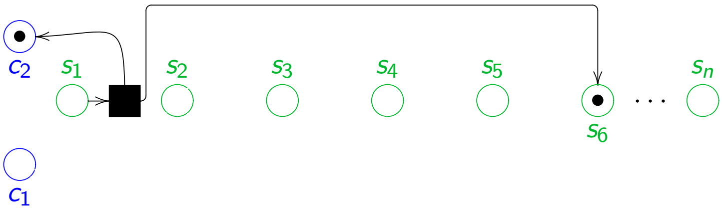


Reduction of Halting Problem to Petri net equivalences

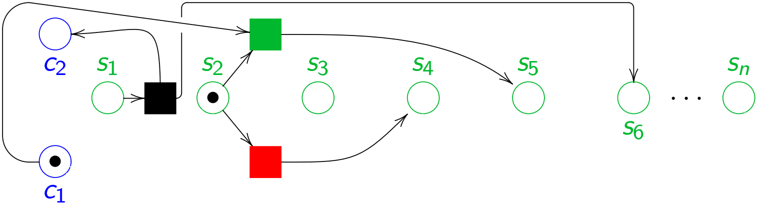
Reduction of Halting Problem to Petri net equivalences



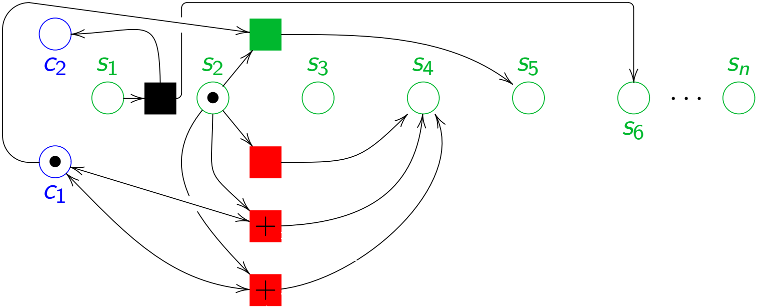
Reduction of Halting Problem to Petri net equivalences



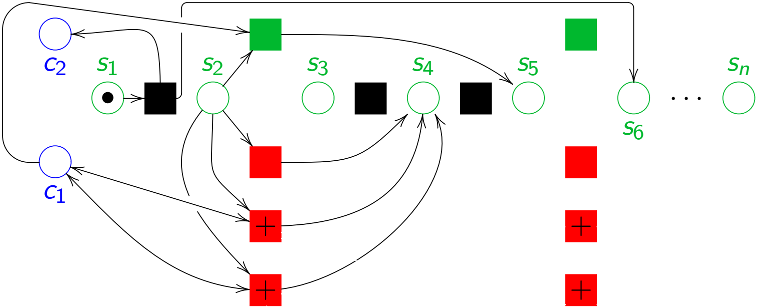
Reduction of Halting Problem to Petri net equivalences



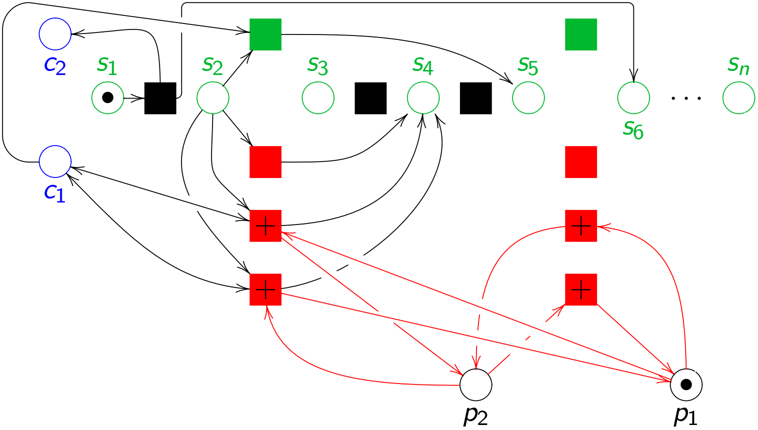
Reduction of Halting Problem to Petri net equivalences



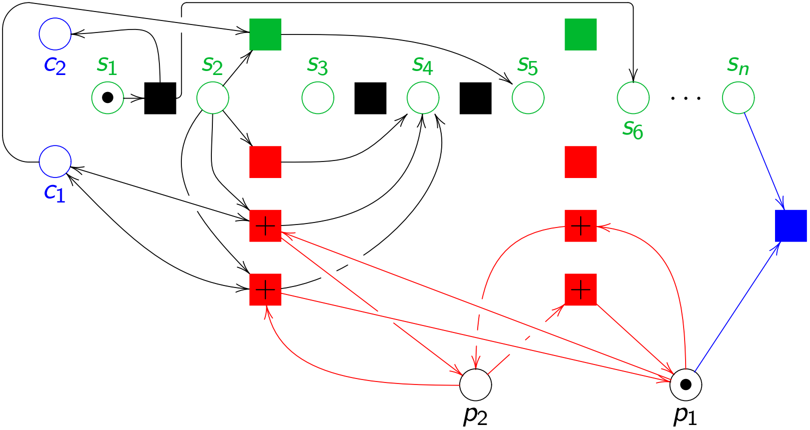
Reduction of Halting Problem to Petri net equivalences



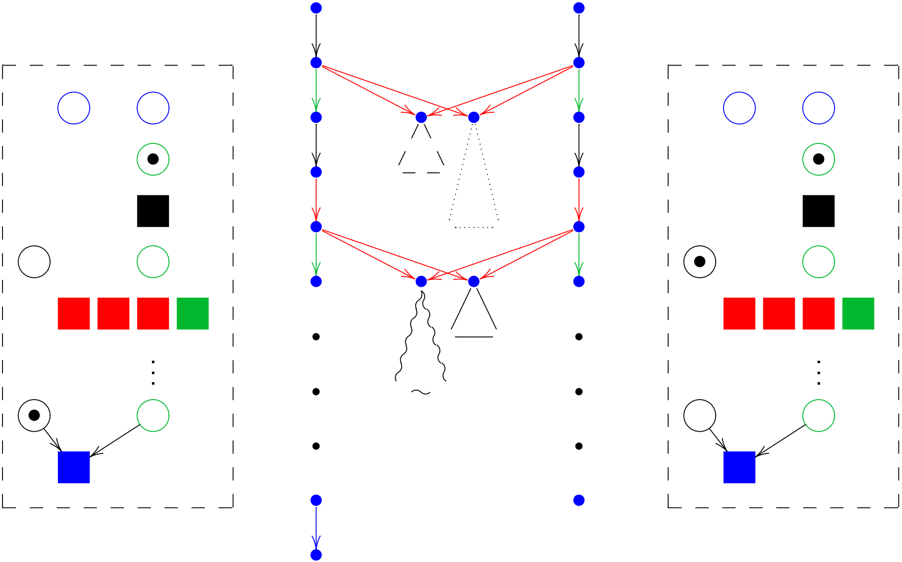
Reduction of Halting Problem to Petri net equivalences



Reduction of Halting Problem to Petri net equivalences



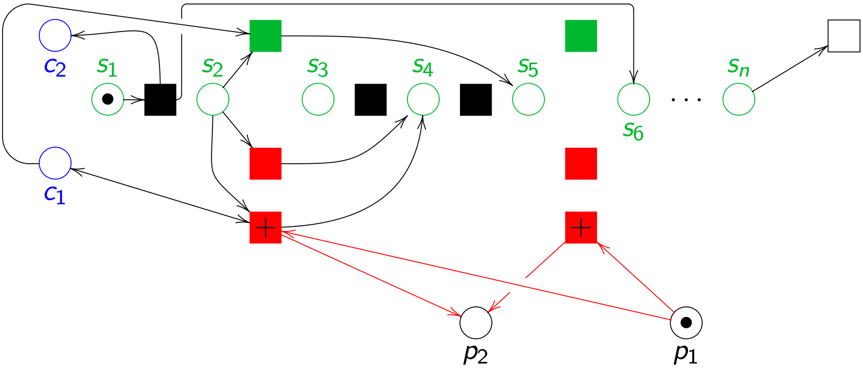
Reduction - cont.



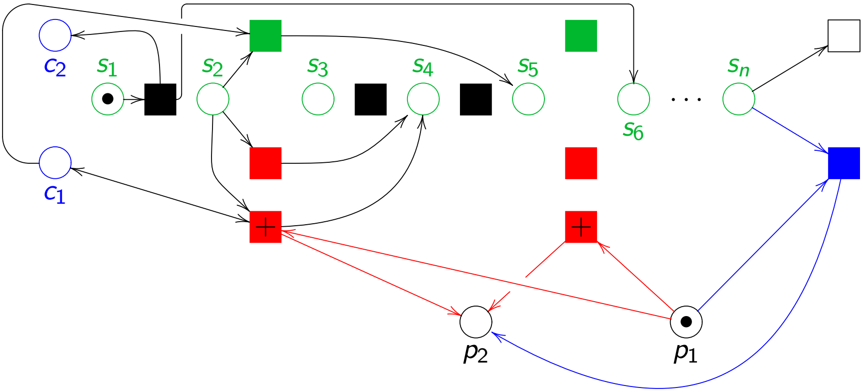
Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.

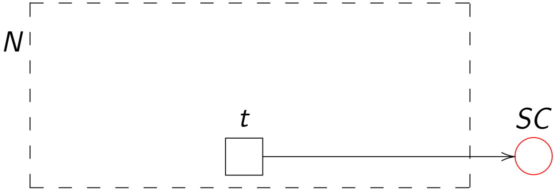
Reduction for the reachability set equality



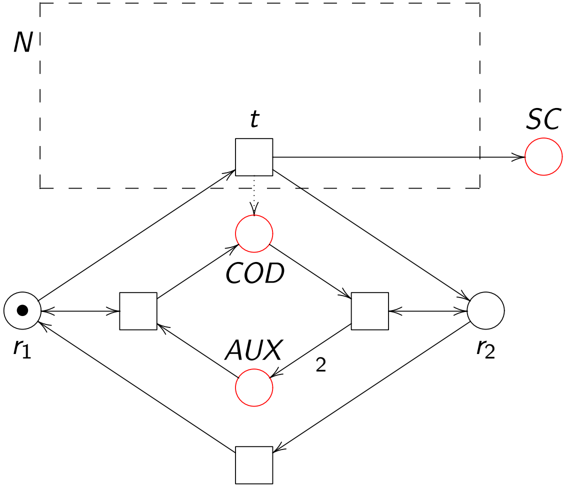
Reduction for the reachability set equality



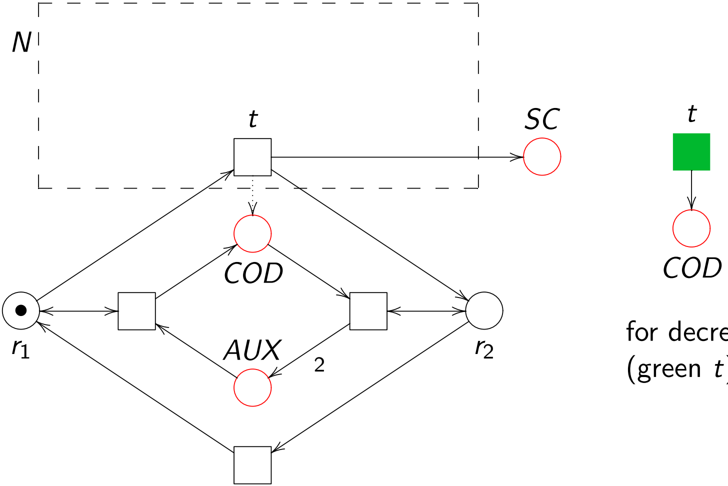
$R(N) \subseteq R(N_{blue})$; when $R(N_{blue}) \subseteq R(N)$?



$R(N) \subseteq R(N_{blue})$; when $R(N_{blue}) \subseteq R(N)$?



$R(N) \subseteq R(N_{blue})$; when $R(N_{blue}) \subseteq R(N)$?



for decreasing
(green t)

Reduction for the reachability set equality – cont.

If C halts, after m steps,
then N_{blue} has a possibility to simulate C correctly, and thus reach a dead marking where

- $SC = m$, and
- COD is in binary 001000010001100100, with length $m+1$, where 1s correspond to the decreasing (green) transitions

Such a marking is unreachable in N; therefore $R(N_{blue}) \not\subseteq R(N)$.

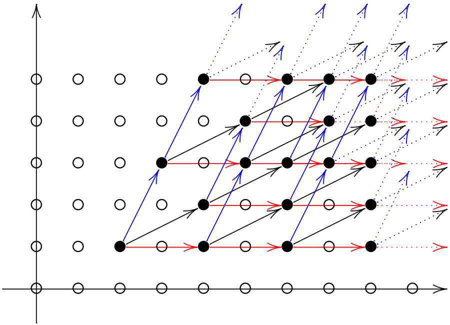
If C does not halt
then $R(N_{blue}) \subseteq R(N)$ (in fact, $R(N_{blue}) = R(N)$).

Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.

Presburger definable sets = semilinear sets

A set $L \subseteq \mathbb{N}^k$ is linear if there are:
a basis $b \in \mathbb{N}^k$ and periods $p_1, p_2, \ldots, p_n \in \mathbb{N}^k$ so that
$L = \{\, b + c_1 p_1 + c_2 p_2 + \cdots + c_n p_n \mid c_1, c_2, \ldots, c_n \in \mathbb{N} \,\}$



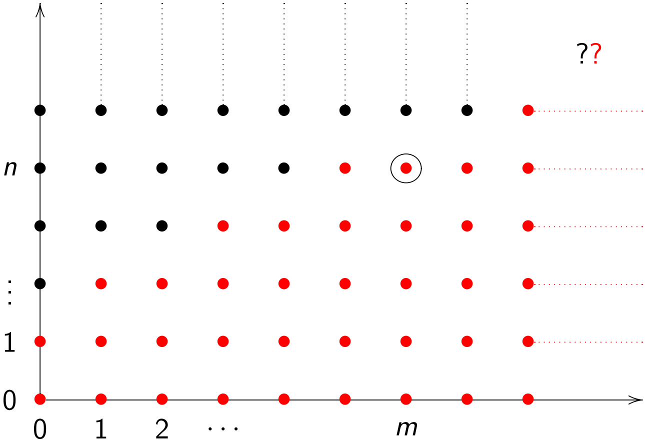
A set $S \subseteq \mathbb{N}^k$ is semilinear iff it is a finite union of linear sets.

Ginsburg, Spanier 1966:
Presburger-definable subsets of $\mathbb{N}^k$ are precisely the semilinear sets.

Reachability set equality – revisited

- For 2 unbounded places, $R(N)$ is (effectively) semilinear; hence $R(N_1) \subseteq R(N_2)$ (and $R(N_1) = R(N_2)$) decidable.
- For 5 unbounded places, $R(N_1) = R(N_2)$ (and $R(N_1) \subseteq R(N_2)$) undecidable.
- What about 3 and 4 unbounded places ??

$p(m)$ is not simulated by $q(n)$ ... red
$p(m)$ is simulated by $q(n)$ ... black

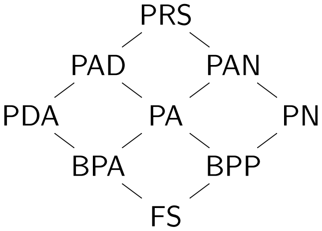


Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.

Process rewrite systems

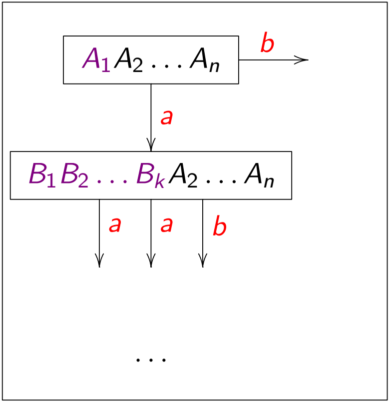
$$E_1 \stackrel{a_1}{\longrightarrow} F_1$$
$$E_2 \stackrel{a_2}{\longrightarrow} F_2$$
$$\dots$$
$$E_k \stackrel{a_k}{\longrightarrow} F_k$$



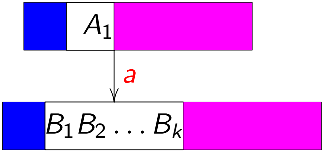
Bisimilarity over 'context-free' processes decidable

Context-free grammar (in Greibach NF)
finitely many rules of the type $A_1 \longrightarrow aB_1B_2 \ldots B_k$

Processes BPA (sequential)



Processes BPP (parallel)



Christensen, Hirshfeld, Moller (1993) decidable (nonprimitive recursive upper bound)

Srba (2002) PSPACE-hard

Jančar (2003) PSPACE-complete

DPDA language equivalence, PDA bisimilarity

G. Sénizergues (1997)
The language equivalence problem for deterministic pushdown automata is decidable

C. Stirling (2002)
Simplified the proof substantially, and showed that it is primitive recursive

They also showed that bisimilarity for nondeterministic pushdown automata decidable.

Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.

Distance-to-disabling functions (dd-functions)

$$dist(s,t) = \min \{ \, length(w) \mid s \xrightarrow{w} t \, \}$$

$dd_a : S \rightarrow \mathbb{N}_\omega$

$$dd_a(s) = \min \{ \, dist(s,t) \mid t \text{ has no } a\text{-successor} \, \}$$

Given a tuple $\mathcal{F} = (d_1, \ldots, d_k)$, each transition $s \xrightarrow{a} t$ determines a change

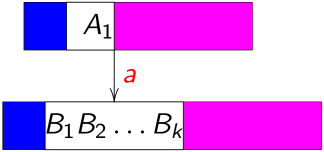
$$\mathcal{F}(t) - \mathcal{F}(s) = \big(d_1(t) - d_1(s), \ldots, d_k(t) - d_k(s) \big).$$

$$dd_{(a,\mathcal{F},\delta)}(s) = \min \{ \, dist(s,t) \mid \forall r: \text{ if } t \xrightarrow{a} r \text{ then } \mathcal{F}(r) - \mathcal{F}(t) \neq \delta \, \}.$$

Observation. $s \sim t \implies d(s) = d(t)$ for all dd-functions d .
The direction $\Longleftarrow$ holds for image finite systems.

Basic Parallel Processes

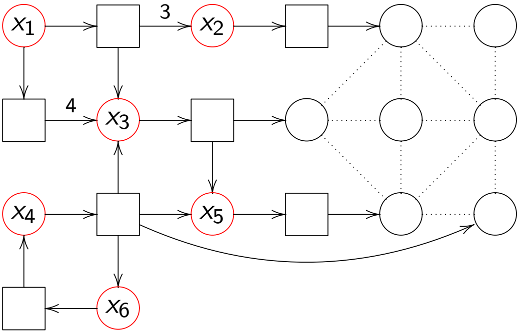
Finitely many rules of the type $A_1 \xrightarrow{a} B_1 B_2 \ldots B_k$



BPP-nets: variables = places

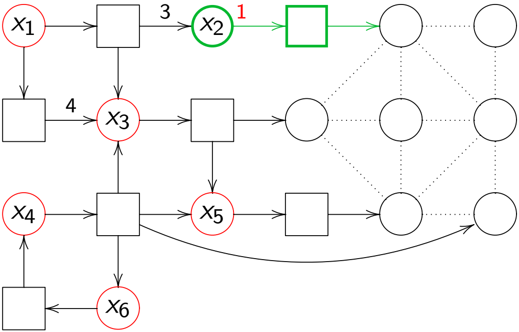
DD-functions on BPP-nets

$$\text{NORM}_Q(M) = \min \{ \mathit{dist}(M, M') \mid M' \text{ has no tokens on } Q \}$$



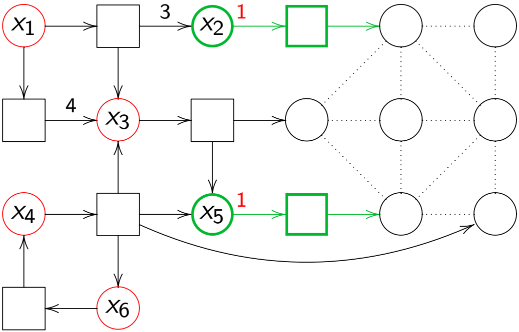
DD-functions on BPP-nets

$$\text{NORM}_Q(M) = \min \left\{ \mathit{dist}(M, M') \mid M' \text{ has no tokens on } Q \right\}$$



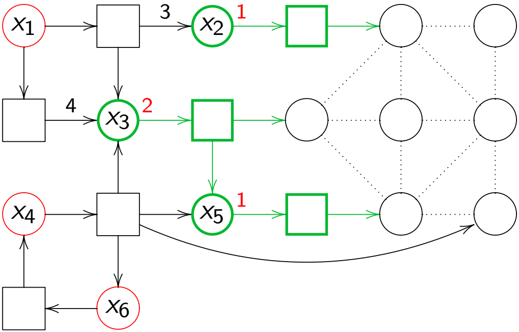
DD-functions on BPP-nets

$$\textsc{norm}_Q(M) = \min \left\{ \mathit{dist}(M, M') \;\middle|\; M' \text{ has no tokens on } Q \right\}$$



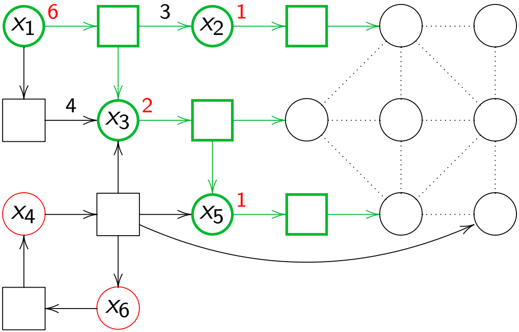
DD-functions on BPP-nets

$$\text{NORM}_Q(M) = \min \{ \, dist(M, M') \mid M' \text{ has no tokens on } Q \, \}$$



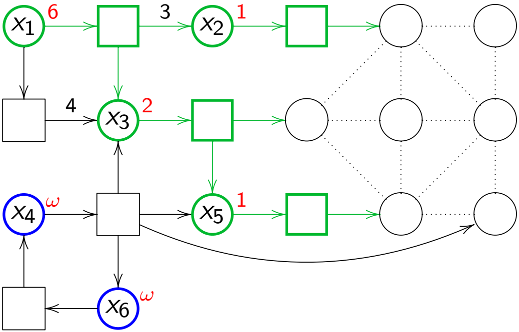
DD-functions on BPP-nets

$$\text{NORM}_Q(M) = \min \{ \mathit{dist}(M, M') \mid M' \text{ has no tokens on } Q \}$$



DD-functions on BPP-nets

$$\text{NORM}_Q(M) = \min \{ \, dist(M, M') \mid M' \text{ has no tokens on } Q \, \}$$



$$\text{NORM}_Q(x_1, x_2, x_3, x_4, x_5, x_6) = 6x_1 + x_2 + 2x_3 + \omega\, x_4 + x_5 + \omega\, x_6$$

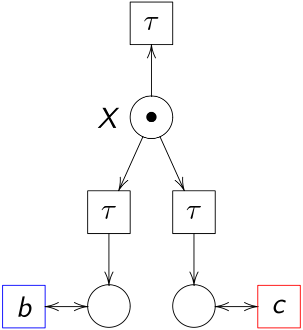
Bisimilarity on Basic Parallel Processes in PSPACE

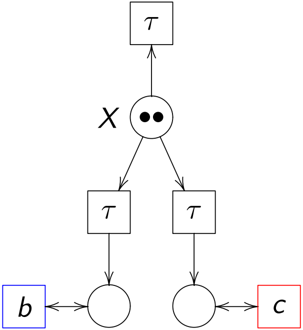
$$M_1 \sim M_2 \iff \forall \text{ important } Q : \mathrm{NORM}_Q(M_1) = \mathrm{NORM}_Q(M_2)$$

The following puzzling question is thus answered positively:

$$X^2 \sim X^3 \Longrightarrow X \sim X^2 \; ?$$

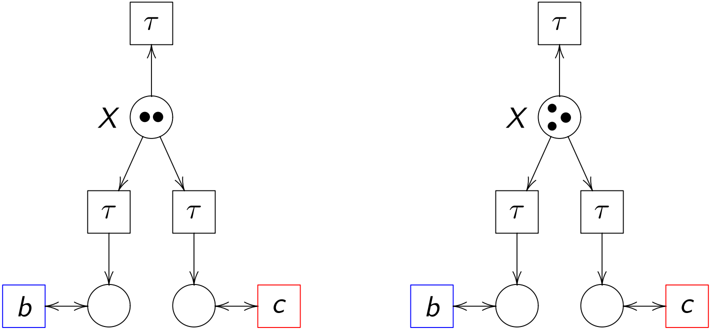
$X \not\approx X^2$





Weak bisimilarity on Basic Parallel Processes

$$X^2 \approx X^3$$



Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.

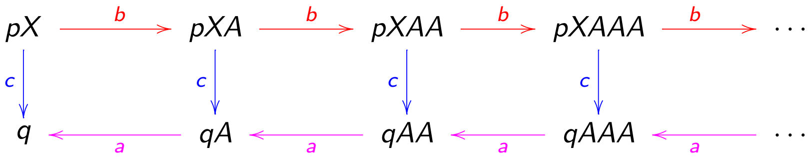
Bisimilarity on prefix rewrite systems

	Normed Processes	Unnormed Processes
Type -2	Σ^1_1-complete	Σ^1_1-complete
Type -1b	Π^0_1-complete	Σ^1_1-complete
Type -1a	Π^0_1-complete	Π^0_1-complete
Type 0, and Type $1\frac{1}{2}$	decidable EXPTIME-hard	decidable EXPTIME-hard
Type 2	$\in$ P P-hard	$\in$ 2-EXPTIME PSPACE-hard
Type 3	P-complete	P-complete

The main new result: Bisimilarity is undecidable on Type -1a systems.

Pushdown graphs; generated by Type 0 systems

$$pX \xrightarrow{b} pXA$$
$$pX \xrightarrow{c} q\varepsilon$$
$$qA \xrightarrow{a} q\varepsilon$$



Type 0 system: finite sets of rules $w_1 \xrightarrow{a} w_2$
(the same class of generated graphs)

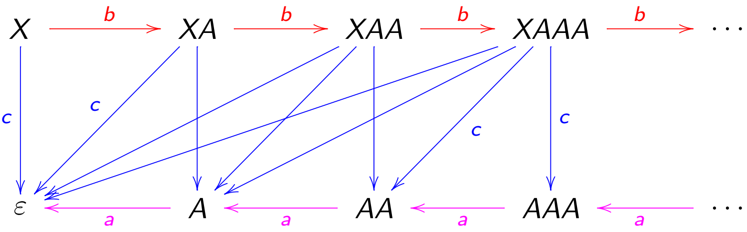
An involved result: Bisimilarity is decidable
(Sénizergues (1998,2005), then Stirling)

Type -1 systems (or Type -1a systems); rules $R \xrightarrow{a} w$

$X \xrightarrow{b} XA$

$XA^* \xrightarrow{c} \varepsilon$

$A \xrightarrow{a} \varepsilon$



Stirling, Sénizergues: is bisimilarity decidable ?

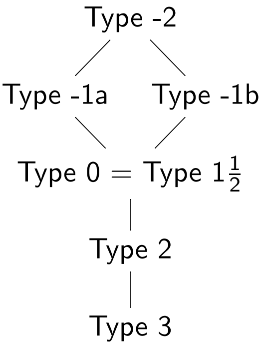
Sénizergues' decidability result for equational graphs of finite out-degree (equivalent to the case $R \xrightarrow{a} w$ with prefix-free R)

Maybe in the normed case ?

(u is normed if each path from u can be prolonged to reach ε)

Hierarchy of prefix rewriting

Type	**Form of Rewrite Rules**
Type -2	$R_1 \xrightarrow{a} R_2$
Type -1a/-1b	$R \xrightarrow{a} w \; / \; w \xrightarrow{a} R$
Type 0	$w \xrightarrow{a} w'$
Type $1\frac{1}{2}$	$pX \xrightarrow{a} qw$
Type 2	$X \xrightarrow{a} w$
Type 3	$X \xrightarrow{a} Y, \; X \xrightarrow{a} \varepsilon$



Bisimilarity is undecidable on Type -1a systems

	Normed Processes	**Unnormed Processes**
Type -2	Σ^1_1-complete	Σ^1_1-complete
Type -1b	Π^0_1-complete	Σ^1_1-complete
Type -1a	Π^0_1-complete	Π^0_1-complete
Type 0, and Type $1\frac{1}{2}$	decidable EXPTIME-hard	decidable EXPTIME-hard
Type 2	$\in$ P P-hard	$\in$ 2-EXPTIME PSPACE-hard
Type 3	P-complete	P-complete

Inf-PCP (a version of Post Correspondence Problem)

A PCP-instance :

u_1	u_2	...	u_n
v_1	v_2	...	v_n

u_i, v_i : nonempty words in an alphabet

An infinite initial solution: a sequence $i_1, i_2, i_3, \ldots$ from $\{1, 2, \ldots, n\}$ such that $i_1{=}1$ and $u_{i_1} u_{i_2} u_{i_3} u_{i_4} \cdots = v_{i_1} v_{i_2} v_{i_3} v_{i_4} \cdots$

Given a Turing machine M, with instructions $(q_0, a) \rightarrow (q_1, b, +1)$, ..., and an input word, say $w = aabab$, we can construct PCP-instance so that: M does not halt on w $\Longleftrightarrow$ there is an infinite initial solution.

$\#$	$q_0 a$	...	a	b	...
$\# q_0 aabab \#$	$b q_1$	...	a	b	...

$\# q_0 aa\ b \ldots$

$\# q_0 aabab \# b q_1\ ab \ldots$

So neg-HP is reducible to inf-PCP; inf-PCP is Π^0_1-complete.

Note: we can even require $|u_i| \leq |v_i|$

Inf-PCP is reducible to bisimilarity on (normed) Type -1a

u_1	u_2	$\ldots$	u_n
v_1	v_2	$\ldots$	v_n

$u_i, v_i \in \{A, B\}^+$, $|u_i| \leq |v_i|$

Observation: The following conditions are equivalent

- $u_{i_1} u_{i_2} u_{i_3} \cdots = v_{i_1} v_{i_2} v_{i_3} \cdots$
- $\forall m$: $u_{i_1} u_{i_2} \cdots u_{i_m}$ is a prefix of $v_{i_1} v_{i_2} \cdots v_{i_m}$
- $\forall m$: $(u_{i_1} u_{i_2} \cdots u_{i_m})^R$ is a suffix of $(v_{i_1} v_{i_2} \cdots v_{i_m})^R$
- $\forall m$: $(u_{i_m})^R (u_{i_{m-1}})^R \cdots (u_{i_1})^R$ is a suffix of $(v_{i_m})^R (v_{i_{m-1}})^R \cdots (v_{i_1})^R$

A game: Defender stepwise generates a sequence

$$\ldots I_{i_m} \ldots I_{i_3} I_{i_2} I_{i_1} \qquad (\text{with } i_1 = 1)$$

Attacker has a possibility to stop this process and win whenever $(u_{i_m})^R (u_{i_{m-1}})^R \cdots (u_{i_1})^R$ is not a suffix of $(v_{i_m})^R (v_{i_{m-1}})^R \cdots (v_{i_1})^R$

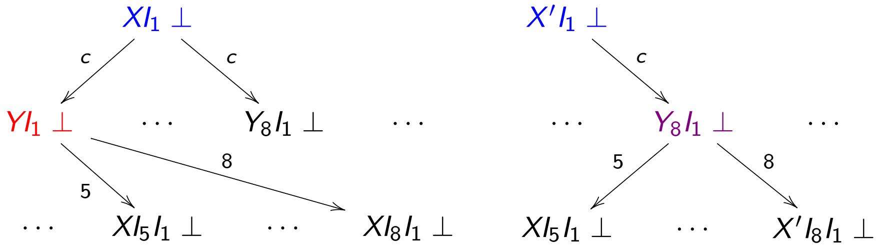
Implementation of the game; generating rules

$X \xrightarrow{c} Y$		
$X \xrightarrow{c} Y_i$	$X' \xrightarrow{c} Y_i$	for all $i \in \{1, 2, \ldots, n\}$
$Y \xrightarrow{i} XI_i$	$Y_i \xrightarrow{i} X'I_i$	for all $i \in \{1, 2, \ldots, n\}$
	$Y_i \xrightarrow{j} XI_j$	for all $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$

Implementation of the game; generating rules

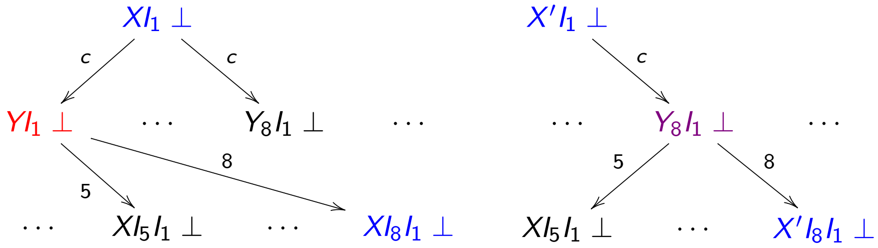
$$X \xrightarrow{c} Y$$

$X \xrightarrow{c} Y_i$	$X' \xrightarrow{c} Y_i$	for all $i \in \{1, 2, \ldots, n\}$
$Y \xrightarrow{i} XI_i$	$Y_i \xrightarrow{i} X'I_i$	for all $i \in \{1, 2, \ldots, n\}$
	$Y_i \xrightarrow{j} XI_j$	for all $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$



Implementation of the game; generating rules

$X \xrightarrow{c} Y$		
$X \xrightarrow{c} Y_i$	$X' \xrightarrow{c} Y_i$	for all $i \in \{1, 2, \ldots, n\}$
$Y \xrightarrow{i} XI_i$	$Y_i \xrightarrow{i} X'I_i$	for all $i \in \{1, 2, \ldots, n\}$
	$Y_i \xrightarrow{j} XI_j$	for all $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$



Switch-to-checking rules

$$X I_7 I_{15} I_3 I_8 I_1 \perp \qquad X' I_7 I_{15} I_3 I_8 I_1 \perp$$

$$(u_{i_7})^R (u_{i_{15}})^R (u_{i_3})^R (u_{i_8})^R (u_{i_1})^R \quad (v_{i_7})^R (v_{i_{15}})^R (v_{i_3})^R (v_{i_8})^R (v_{i_1})^R$$

$X \xrightarrow{d} C$

$X(I^*)I_i \xrightarrow{d} C'w \qquad X'(I^*)I_i \xrightarrow{d} C'w \qquad$ for all $i \in \{1, 2, \ldots, n\}$ and all suffices w of v_i^R

Notation: I^* stands for $(I_1 + I_2 + \cdots + I_n)^*$;

$$X I_7 I_{15} I_3 I_8 I_1 \perp \xrightarrow{d} C I_7 I_{15} I_3 I_8 I_1 \perp$$

$$X' I_7 I_{15} I_3 I_8 I_1 \perp \xrightarrow{d} C' w I_8 I_1 \perp$$

Checking rules

$C\, I_7 I_{15} I_3 I_8 I_1 \perp$ $\qquad$ $C'\, w I_8 I_1 \perp$

$$CA \xrightarrow{a} C \qquad C'A \xrightarrow{a} C'$$
$$CB \xrightarrow{b} C \qquad C'B \xrightarrow{b} C'$$
$$C\perp \xrightarrow{e} \varepsilon \qquad C'\perp \xrightarrow{e} \varepsilon$$
$$CI_i \xrightarrow{h(u_i^R)} C\ tail(u_i^R) \qquad C'I_i \xrightarrow{h(v_i^R)} C'\ tail(v_i^R)$$
$$\text{for all } i \in \{1, 2, \ldots, n\}$$

Notation. $h(w) = a$ when $head(w) = A$, $h(w) = b$ when $head(w) = B$

Observation. $C\, I_7 I_{15} I_3 I_8 I_1 \perp$ is bisimilar to $C'\, w I_8 I_1 \perp$

iff

$$(u_{i_7})^R (u_{i_{15}})^R (u_{i_3})^R (u_{i_8})^R (u_{i_1})^R = w (v_{i_8})^R (v_{i_1})^R$$

Summary of the reduction: inf-PCP $\longrightarrow$ bisim-Type-1a

u_1	u_2	$\ldots$	u_n
v_1	v_2	$\ldots$	v_n

$\longrightarrow$

generating rules
switch-to-checking rules
checking rules

There is an infinite initial solution $\iff$ $XI_1 \perp$ is bisimilar with $X'I_1 \perp$

(Moreover: $XI_1 \perp$, $X'I_1 \perp$ are normed processes.)

Π^0_1-completeness of bisim-normed-Type-1b

The only difference wrt Type -1a: switch-to-checking rules

$X I_7 I_{15} I_3 I_8 I_1 \perp$ $\qquad$ $X' I_7 I_{15} I_3 I_8 I_1 \perp$

$(u_{i_7})^R (u_{i_{15}})^R (u_{i_3})^R (u_{i_8})^R (u_{i_1})^R \quad (v_{i_7})^R (v_{i_{15}})^R (v_{i_3})^R (v_{i_8})^R (v_{i_1})^R$

$$X' \xrightarrow{d} C'$$
$$X \xrightarrow{d} C(A+B)^* \qquad X' \xrightarrow{d} C(A+B)^*$$

$$X I_7 I_{15} I_3 I_8 I_1 \perp \xrightarrow{d} C w I_7 I_{15} I_3 I_8 I_1 \perp$$

$$X' I_7 I_{15} I_3 I_8 I_1 \perp \xrightarrow{d} C' I_7 I_{15} I_3 I_8 I_1 \perp$$

High undecidability; Σ^1_1-completeness; rec-PCP

A predicate $\mathcal{P}(x)$ (over natural numbers) is in Σ^1_1 iff

$$\mathcal{P}(x) \Leftrightarrow \exists M \mathcal{F}(M, x)$$

for a set variable M and a first-order arithmetical formula $\mathcal{F}$.

A well known Σ^1_1-complete problem:
is there an infinite computation of a given nondeterministic Turing machine which visits q_0 infinitely often ?

A PCP-instance :

u_1	u_2	...	u_n
v_1	v_2	...	v_n

u_i, v_i : nonempty words in an alphabet

Recurrent solution: a sequence $i_1, i_2, i_3, \ldots$ from $\{1, 2, \ldots, n\}$ such that

- $i_1 = 1$ and $u_{i_1} u_{i_2} u_{i_3} u_{i_4} \cdots = v_{i_1} v_{i_2} v_{i_3} v_{i_4} \cdots$
- 1 appears infinitely often in the sequence $i_1, i_2, i_3, \ldots$

Fact. Problem rec-PCP is Σ^1_1-complete.

Σ_1^1-completeness for (unrestricted) Type -1b

$X \xrightarrow{c} Y$	
$X \xrightarrow{c} Y' I_1 I^*$	$X' \xrightarrow{c} Y' I_1 I^*$
	$Y' \xrightarrow{c} X'$
$Y \xrightarrow{c} X I^* \bot$	$Y' \xrightarrow{c} X I^* \bot$

	$X' \xrightarrow{d} C'$
$X \xrightarrow{d} C(A+B)^*$	$X' \xrightarrow{d} C(A+B)^*$
$X \xrightarrow{f} Z$	$X' \xrightarrow{f} Z'$

$CA \xrightarrow{a} C$	$C'A \xrightarrow{a} C'$	
$\dots$		
$ZI_i \xrightarrow{i} Z$	$Z'I_i \xrightarrow{i} Z'$	for all $i \in \{1, 2, \dots, n\}$
$Z\bot \xrightarrow{e} \varepsilon$	$Z'\bot \xrightarrow{e} \varepsilon$	

Σ_1^1-completeness for normed Type -2

Like for the previous Type -1b, the only difference is the generating rules:

$$
\begin{array}{ll}
X \xrightarrow{c} Y & \\
X \xrightarrow{c} Y' I_1 I^* & X' \xrightarrow{c} Y' I_1 I^* \\
 & Y' \xrightarrow{c} X' \\
Y I^* \perp \xrightarrow{c} X I^* \perp & Y' I^* \perp \xrightarrow{c} X I^* \perp
\end{array}
$$

Bisimilarity on prefix rewrite systems

	Normed Processes	Unnormed Processes
Type -2	Σ^1_1-complete	Σ^1_1-complete
Type -1b	Π^0_1-complete	Σ^1_1-complete
Type -1a	Π^0_1-complete	Π^0_1-complete
Type 0, and Type $1\frac{1}{2}$	decidable EXPTIME-hard	decidable EXPTIME-hard
Type 2	$\in$ P P-hard	$\in$ 2-EXPTIME PSPACE-hard
Type 3	P-complete	P-complete

Outline

- Behavioural equivalences; bisimulation equivalence.
- Undecidability of equivalences on (labelled place/transition) Petri nets (STACS 1994); imperfect Minsky machine simulation.
  Undecidability of the reachability set equality.
- Semilinear witnesses; decidable cases.
- Bisimilarity on process rewrite systems, and prefix rewrite systems.
- PSPACE-completeness of bisimilarity on Basic Parallel Processes (LiCS 2003); distance-to-disabling functions (dd-functions).
- Undecidability of bisimilarity on Type -1 systems (Jančar and Srba, FoSSaCS 2006); Defender's choice technique.
- THE END