"Defender's Forcing" – a method for showing hardness of behavioural equivalences

Petr Jančar

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

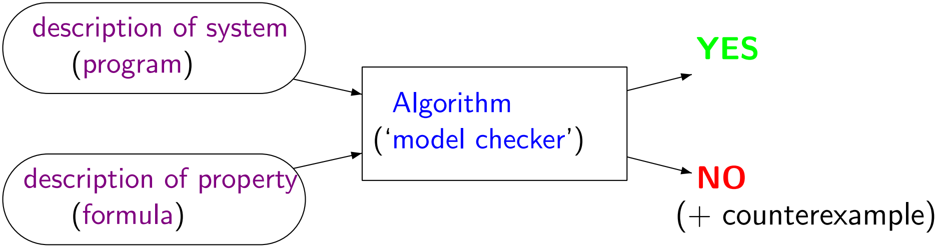
CAK, ES-Colloquium, Prague, 5 February 2008

Overview

- Recalling some areas of automated verification
  - model checking
  - behavioural equivalence (or 'implementation preorder') checking
- "Defender's Forcing" (in bisimulation games)
  - An (old) application for Petri net equivalences
  - A (new) undecidability result for a generalization of pushdown graphs
- Publication in Journal of the ACM

Model checking

Question: Does the given system have the tested property?

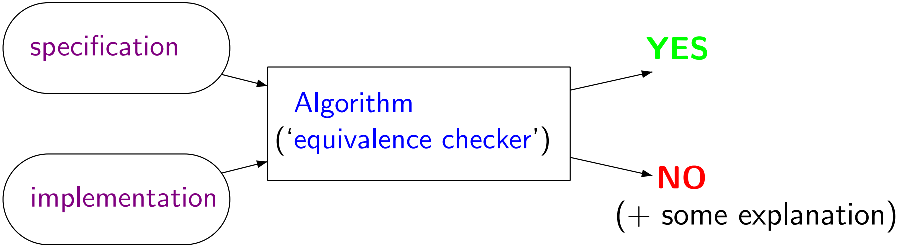


SPIN
http://spinroot.com/
Bell Labs

SMV (Symbolic model checking)
http://www.cs.cmu.edu/~modelcheck
Carnegie Mellon Univ.

Equivalence checking

Question: Do the two given systems have the same behaviour?

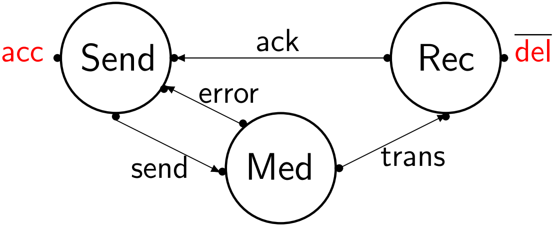


Remark. A comprehensive survey of verification tools
(for model checking, equivalence checking, ...)
at Faculty of Informatics, Masaryk University Brno, Czech Rep.
http://anna.fi.muni.cz/yahoda/

Simple communication protocol

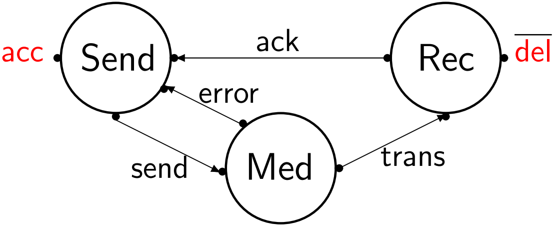
interface: acc $\boxed{\mathsf{P}}$ $\overline{\mathsf{del}}$ ..... behaviour: Spec $\stackrel{\text{def}}{=}$ acc.$\overline{\mathsf{del}}$.Spec

Simple communication protocol



interface: acc $\boxed{\text{P}}$ $\overline{\text{del}}$ ..... behaviour: Spec $\stackrel{\text{def}}{=}$ acc.$\overline{\text{del}}$.Spec

Simple communication protocol



interface: acc $\boxed{\text{P}}$ $\overline{\text{del}}$ ..... behaviour: $\text{Spec} \stackrel{\text{def}}{=} \text{acc}.\overline{\text{del}}.\text{Spec}$

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
\text{Send} & \stackrel{\text{def}}{=} & \text{acc.Sending} & \text{Rec} & \stackrel{\text{def}}{=} & \text{trans.Del} \\
\text{Sending} & \stackrel{\text{def}}{=} & \overline{\text{send}}.\text{Wait} & \text{Del} & \stackrel{\text{def}}{=} & \overline{\text{del}}.\text{Ack} \\
\text{Wait} & \stackrel{\text{def}}{=} & \text{ack.Send} + \text{error.Sending} & \text{Ack} & \stackrel{\text{def}}{=} & \overline{\text{ack}}.\text{Rec}
\end{array}
$$

$$
\begin{array}{rcl}
\text{Med} & \stackrel{\text{def}}{=} & \text{send.Med}' \\
\text{Med}' & \stackrel{\text{def}}{=} & \tau.\text{Err} + \overline{\text{trans}}.\text{Med} \\
\text{Err} & \stackrel{\text{def}}{=} & \overline{\text{error}}.\text{Med}
\end{array}
$$

Question for Equivalence Checking

Here description of specification and implementation in CCS (Calculus of Communicating Systems, Milner)

$$\mathsf{Spec} \stackrel{\text{def}}{=} \mathsf{acc}.\overline{\mathsf{del}}.\mathsf{Spec}$$

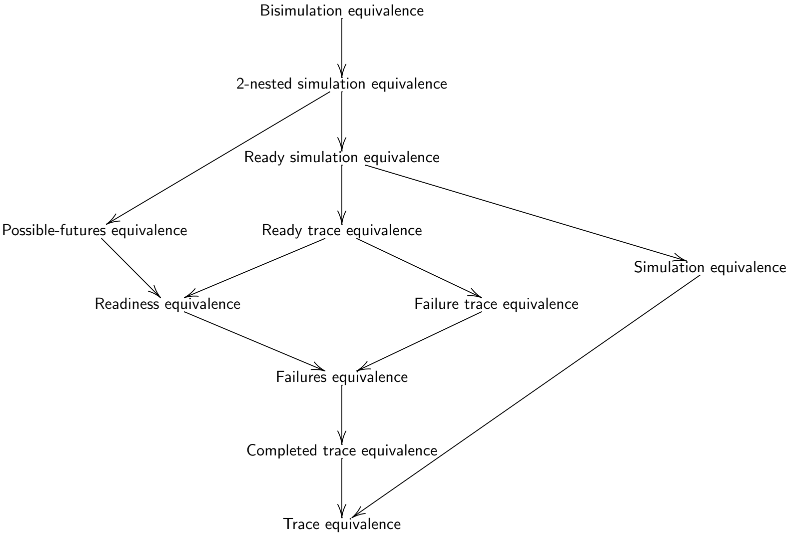
$$\mathsf{Impl} \stackrel{\text{def}}{=} (\mathsf{Send} \mid \mathsf{Med} \mid \mathsf{Rec}) \setminus \{\mathsf{send}, \mathsf{trans}, \mathsf{ack}, \mathsf{error}\}$$

Verification question: $\boxed{\mathsf{Impl} \stackrel{?}{\approx} \mathsf{Spec}}$

A good behavioural equivalence $\approx$ is here (weak) bisimulation equivalence.

CWB
(Concurrency Workbench)
www.lfcs.inf.ed.ac.uk/cwb/
Univ. Edinburgh, UK

Linear Time / Branching Time Spectrum



An (old) application of "Defender's Forcing"

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 $c_1 = c_2 = 0$).

Jančar (Journal of Theoretical Computer Science, 1995):
a (reducing) algorithm A

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

such that

- if C halts (on zero input) 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)

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)

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

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

$$\boxed{c_j := c_j + 1;\ goto\ k}$$

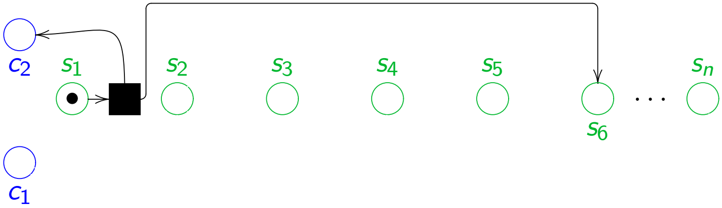
$$\boxed{\textbf{if } c_j = 0 \textbf{ then } goto\ k_1 \textbf{ else } (c_j := c_j - 1;\ goto\ k_2)}$$

Reduction of halting problem to Petri net equivalences

$s_1 : \; c_2 := c_2 + 1; \; goto \; s_6$

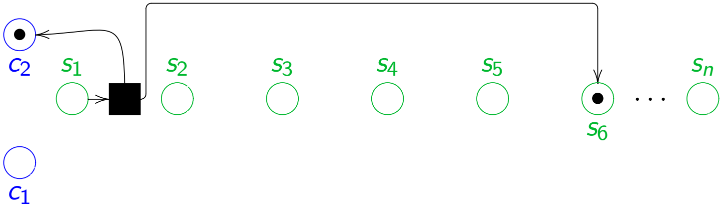
Reduction of halting problem to Petri net equivalences

$s_1 : \; c_2 := c_2 + 1; \; goto \; s_6$



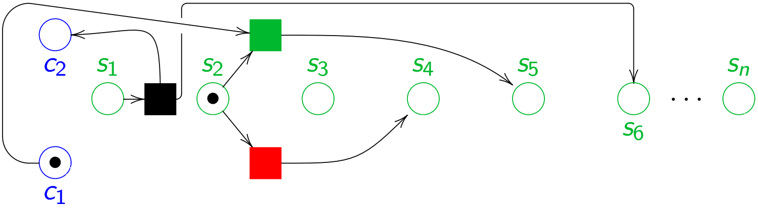
Reduction of halting problem to Petri net equivalences

s_2 : if $c_1 = 0$ then goto s_4 else ($c_1 := c_1 - 1$; goto s_5)

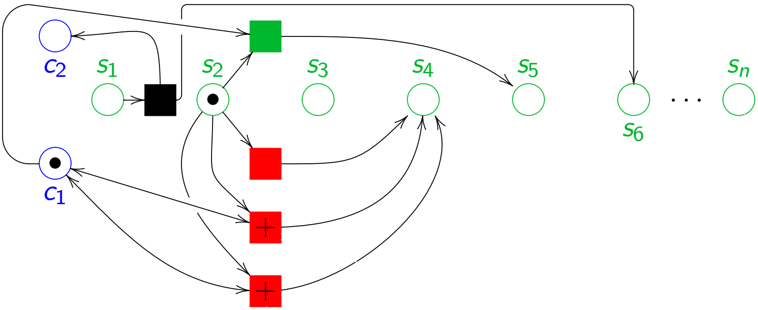


Reduction of halting problem to Petri net equivalences

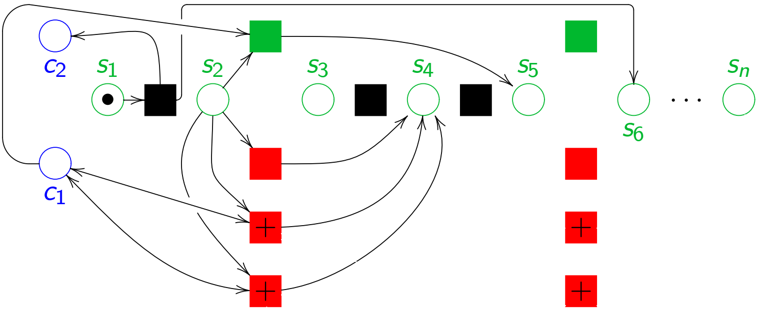
$s_2 : \quad \text{if} \;\; c_1 = 0 \; \text{then} \;\; \text{goto} \; s_4 \; \text{else} \; (c_1 := c_1 - 1; \; \text{goto} \; s_5)$



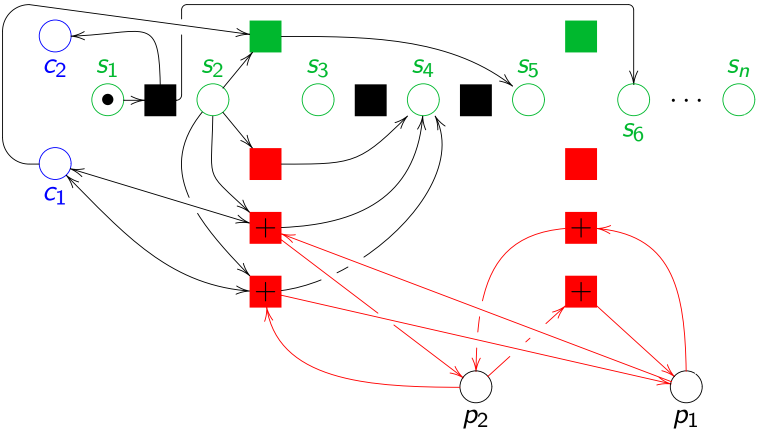
Reduction of halting problem to Petri net equivalences



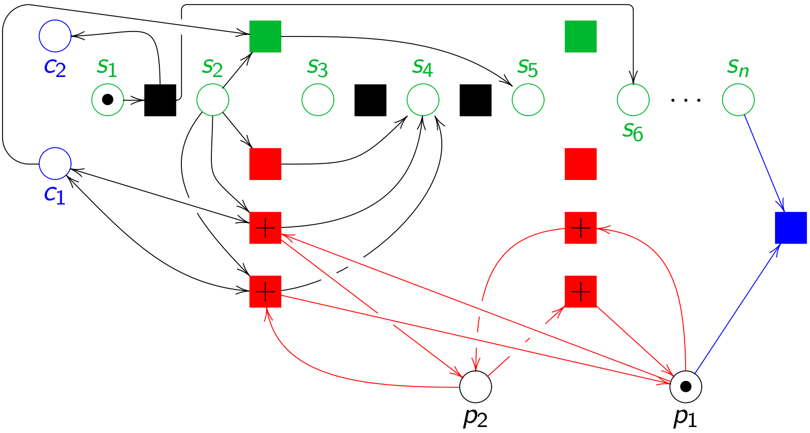
Reduction of halting problem to Petri net equivalences



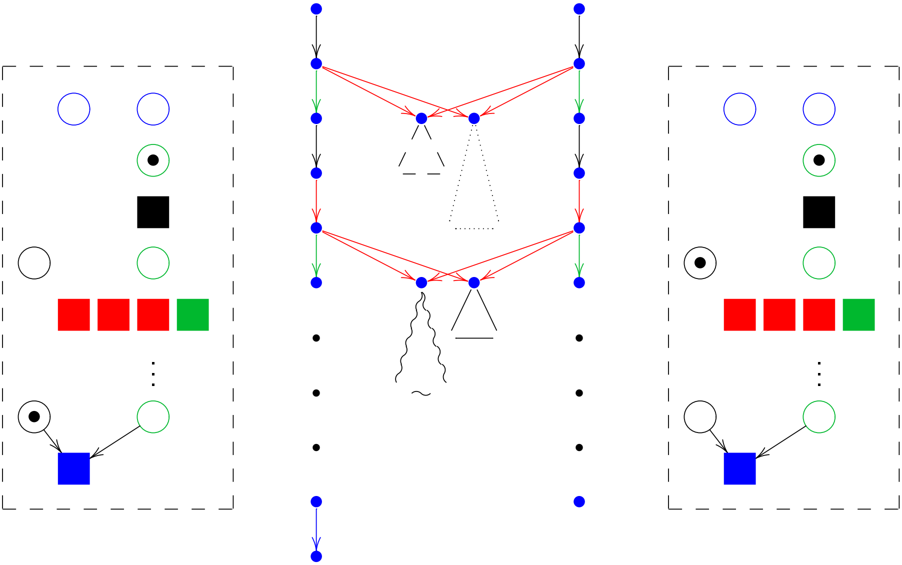
Reduction of halting problem to Petri net equivalences



Reduction of halting problem to Petri net equivalences

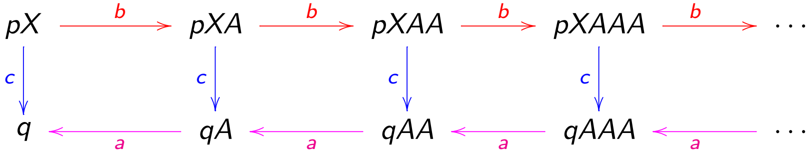


Reduction of HP to Petri nets - cont.



Pushdown graphs; generated by Type 0 systems

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

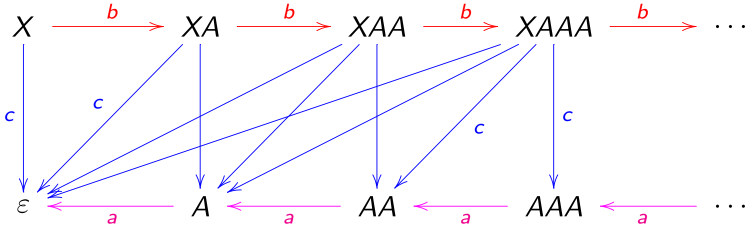


Type 0 system: finite sets of rules $w_1 \xrightarrow{a} w_2$
(the same class of generated graphs)

An involved result: Bisimilarity is decidable
(Sénizergues (1998,2005), then Stirling)
(After the famous decidability of DPDA language equivalence)

Type -1 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)

Jančar, Srba: it is undecidable

	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

Article to appear in JACM

Jančar P., Srba J.: Undecidability of bisimilarity by Defender's forcing;

to appear in Journal of the ACM, vol. 55 (2008), No. 1 (February 2008)

JACM (http://jacm.acm.org/)

The Journal of the Association for Computing Machinery provides coverage of the most significant work going on in computer science, broadly construed. We publish original research papers of lasting value in computer science.

There are three main criteria for a paper to be accepted:

1. the paper must be among the best papers of the year in its area,
2. the paper must be of interest to the broad community,
3. the presentation must be effective.

(Vol. 1 - 1954, Vol. 2 - 1955, . . . , Vol. 54 - 2007, Vol. 55 - 2008)

(approx. 30 - 40 articles per year)