Complexity of Equivalence Checking Problems

Zdeněk Sawa

Department of Computer Science

Technical University of Ostrava, 17. listopadu 15

708 33 Ostrava–Poruba, Czech Republic

zdenek.sawa@vsb.cz

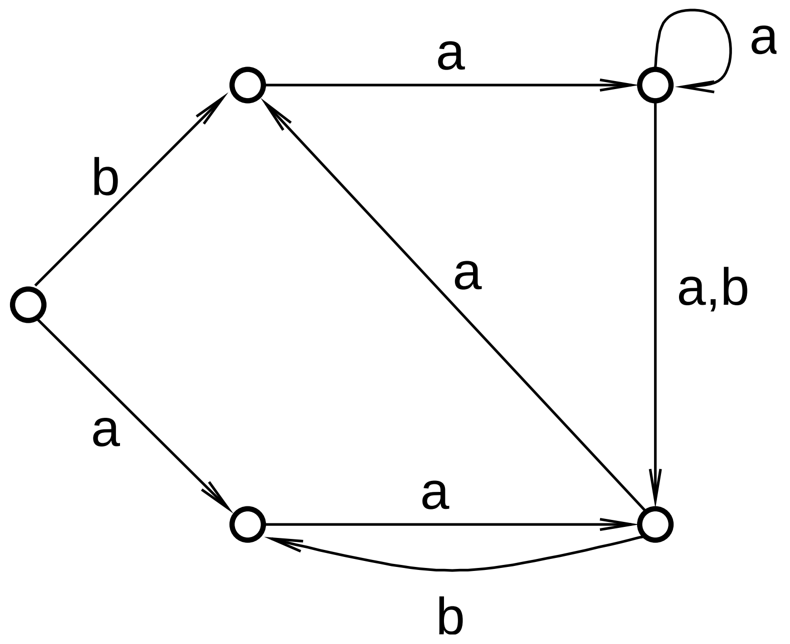
Examples of complicated systems:

- operating systems
- network communication protocols
- microprocessors
- parallel algorithms
- distributed algorithms
- traffic control systems
- ...

Necessity of formal methods:

- testing and modeling explore ***some*** of possible behaviours
- formal methods allow to verify ***all*** possible behaviours
  - construction of rigorous mathematical proofs
  - may be automated (to some extend)

An example of a labelled transition system:



How a labelled transition system can be described:

- automata (finite state automata, pushdown automata, counter machines, ...)
- process algebras (CCS, CSP, π-calculus)
- Petri nets

Two main types of problems:

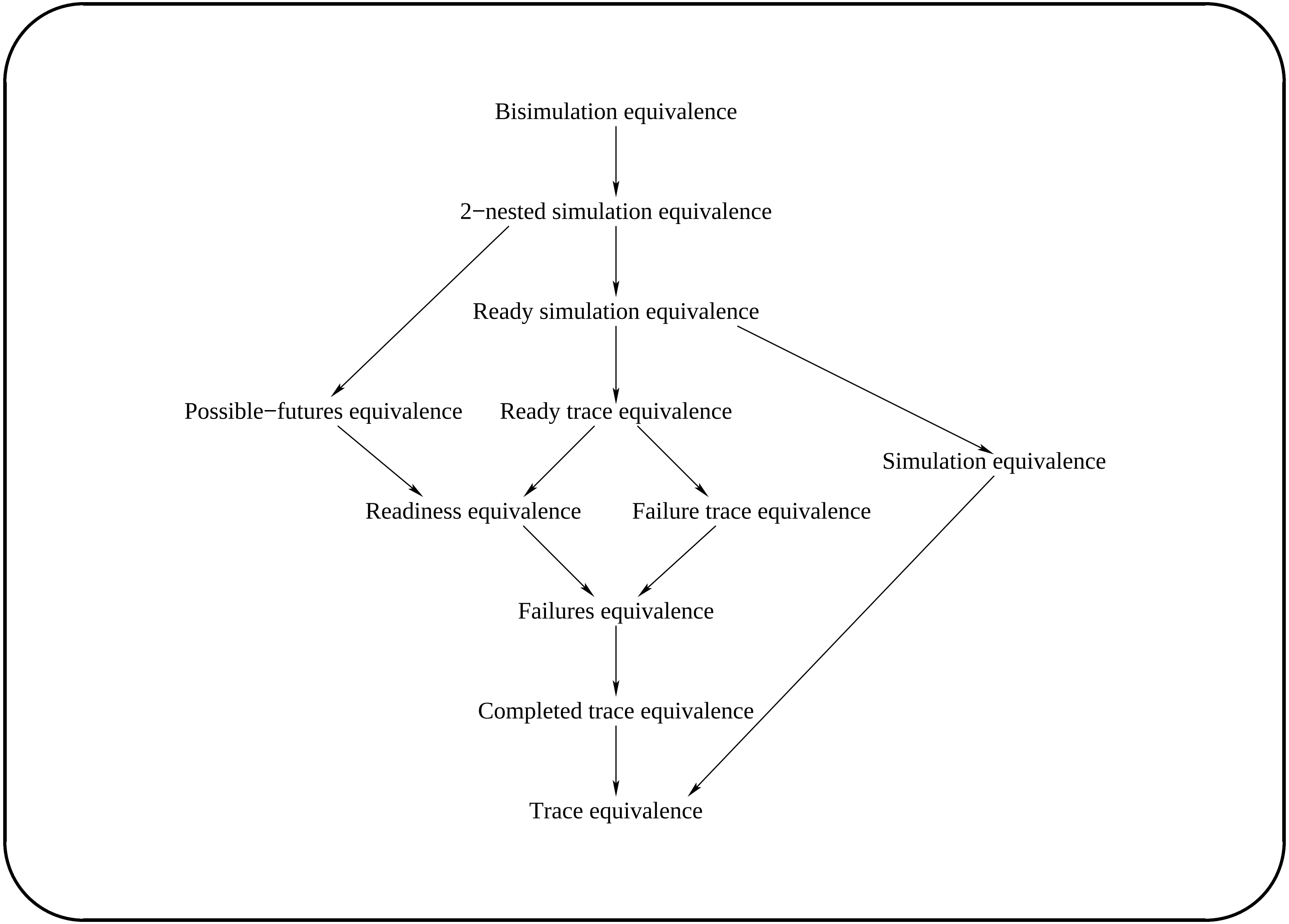
- ***Model checking***

INSTANCE: a labelled transition system $\mathcal{T}$ and a formula ϕ
QUESTION: Does $\mathcal{T}$ satisfy ϕ ?

Types of temporal logics: LTL, CTL, CTL*, μ-calculus, ...

- ***Equivalence checking***

INSTANCE: two labelled transition systems $\mathcal{T}_1, \mathcal{T}_2$
QUESTION: Is $\mathcal{T}_1$ equivalent to $\mathcal{T}_2$?



Bisimulation equivalence
2−nested simulation equivalence
Ready simulation equivalence
Possible−futures equivalence
Ready trace equivalence
Simulation equivalence
Readiness equivalence
Failure trace equivalence
Failures equivalence
Completed trace equivalence
Trace equivalence

Interesting questions:

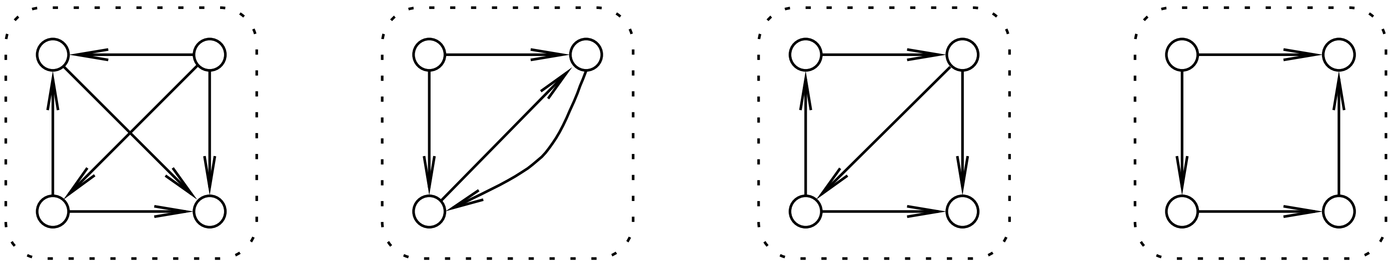
- Where are the limits of automated verification ?
- What problems are decidable ?
- What is the computational complexity of decidable problems ?

Overview of own results:

- *EXPTIME*-hardness of equivalence checking of non-flat systems (CONCUR 2003)
- *PTIME*-hardness of equivalence checking of flat systems (SOFSEM 2001)
- *DP*-hardness of problems concerning one-counter automata (FOSSACS 2002)
- undecidability of deciding simulation equivalence for one-counter automata (SOFSEM '99)

EXPTIME-hardness of equivalence checking of non-flat systems:

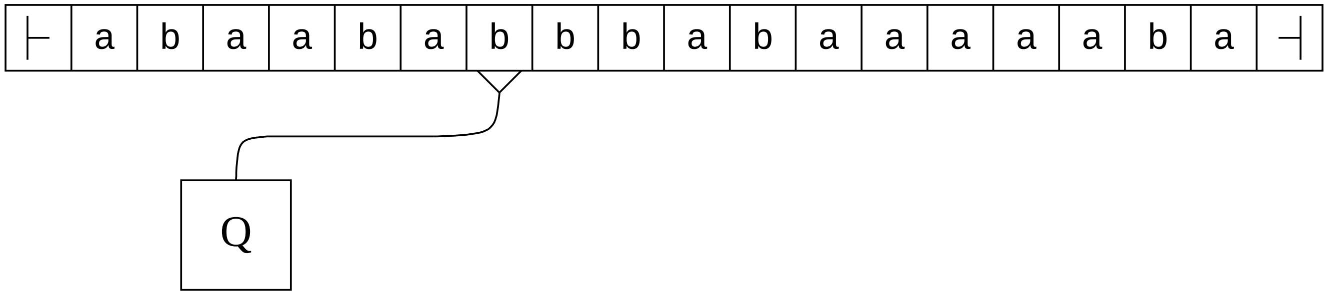
Parallel composition with hiding:



$\mathcal{A}_1$ $\mathcal{A}_2$ $\mathcal{A}_3$ $\mathcal{A}_4$

EXPTIME-hard for every relation between bisimilarity and trace preorder (conjectured by A. Rabinovich [Rab97])

Reactive linear bounded automata (RLBA):



- a new auxiliary model introduced in the proof
- considerably simplifies the proof
- allows simple generalization to other types of non-flat systems (labelled 1-safe Petri nets)

PTIME-hardness of equivalence checking of flat systems:

- ***Flat systems*** – states and transitions are given explicitly.
- The problem is *PTIME*-hard for every relation between bisimilarity and trace preorder.
- Implies that equivalence checking can not be efficiently parallelized – there is no efficient parallel algorithm unless $NC = PTIME$.

A method for proving DP-hardness of verification problems concerning one-counter automata:

- ***One-counter automaton*** – a finite state automaton equipped with a counter
- ***One-counter net*** – can not test for zero, corresponds to a Petri net with at most one unbounded place

General idea:

OCL (One-Counter Logic) – a fragment of Presburger arithmetic, reductions from the deciding of the truth of formulas in OCL

The method was used to show DP-hardness of:

- equivalence checking of one-counter nets for any relation between bisimilarity and simulation preorder
- deciding simulation equivalence and simulation preorder for a one-counter automaton and a finite state system (in both directions)
- model checking for a one-counter net and a formula from EF (a fragment of CTL)