Seventh Electrotechnical and Computer Science Conference ERK '98
Portorož, Slovenia

Computing Testing Equivalence with Binary Decision Diagrams

Robert Meolic, Tatjana Kapus, Zmago Brezočnik
Faculty of Electrical Engineering and Computer Science
University of Maribor
Smetanova 17, SI-2000 Maribor, Slovenia
{meolic,kapus,brezocnik}@uni-mb.si

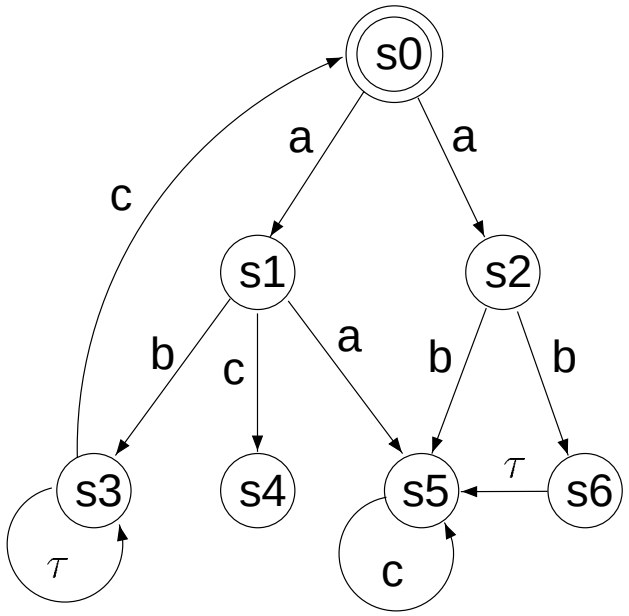
Contents:

Work supported by Ministry of Science and Technology,
Republic of Slovenia

Introduction

- **Process algebras** are a convenient tool for describing and reasoning about the behaviour of concurrent systems.
- Systems are described as transition graphs called **processes**.
- **Equivalence relations** are used to study the relationship between different systems or between different levels of abstractions of the system.

Process

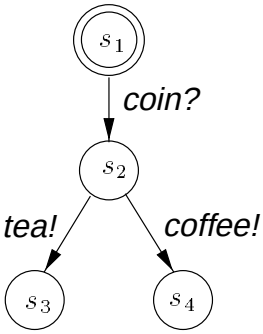


$P = (\mathcal{S}, Act, \delta, p_0)$

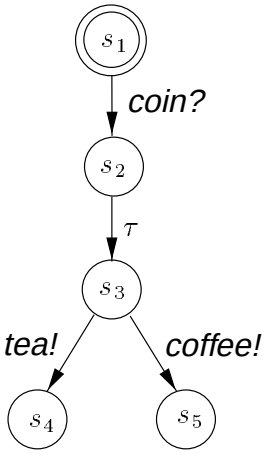
- $\mathcal{S}$: set of states
- Act: set of actions
- δ: transition relation
- p_0: initial state

- Act contains **visible actions** and **silent action** τ.
- p is a **deadlock state** iff no actions can be performed from it.
- p is a **divergent state** iff there is an infinite sequence of silent actions starting in it.

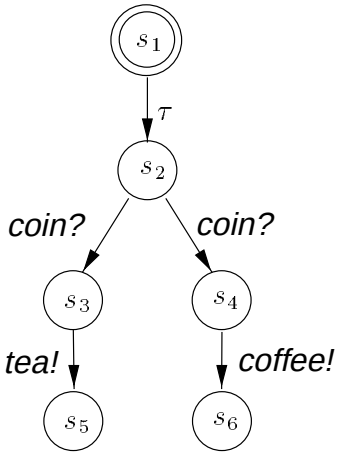
Some different descriptions of drink machine



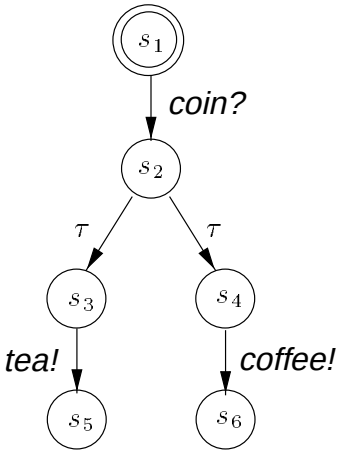
Drink machine A1



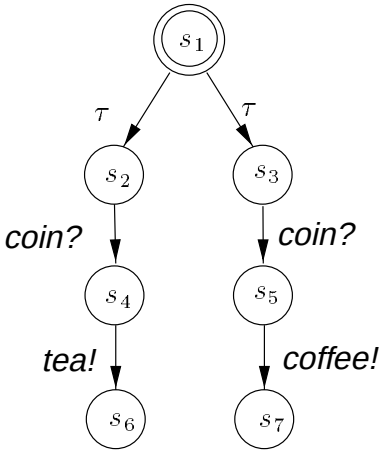
Drink machine A2



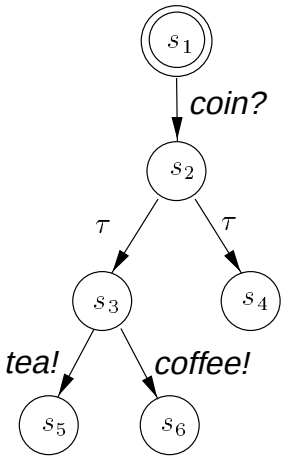
Drink machine A3



Drink machine A4



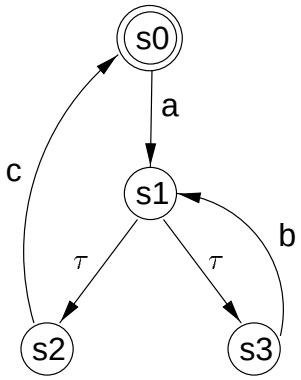
Drink machine A5



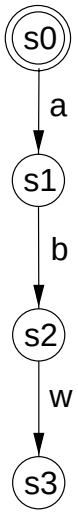
Drink machine A6

Definition of Testing Equivalence

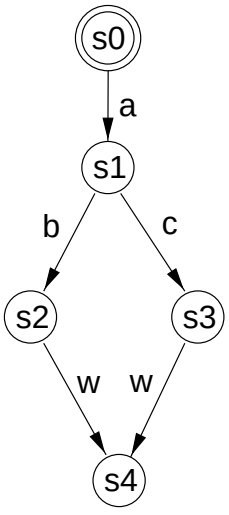
- An **observer** is a process which runs concurrently with the observed process and interacts with it. The observer uses special action w to report to the external world the **success of the observation**.
- The process P **must satisfy** the observer O if during the observation the action w is inevitable. The process P **may satisfy** the observer O if during the observation the action w is possible.
- Two processes are **must equivalent** iff they must satisfy the same sets of observers. They are **may equivalent** iff they may satisfy the same sets of observers. The processes are **testing equivalent** iff they are must equivalent and may equivalent.



P



O_1



O_2

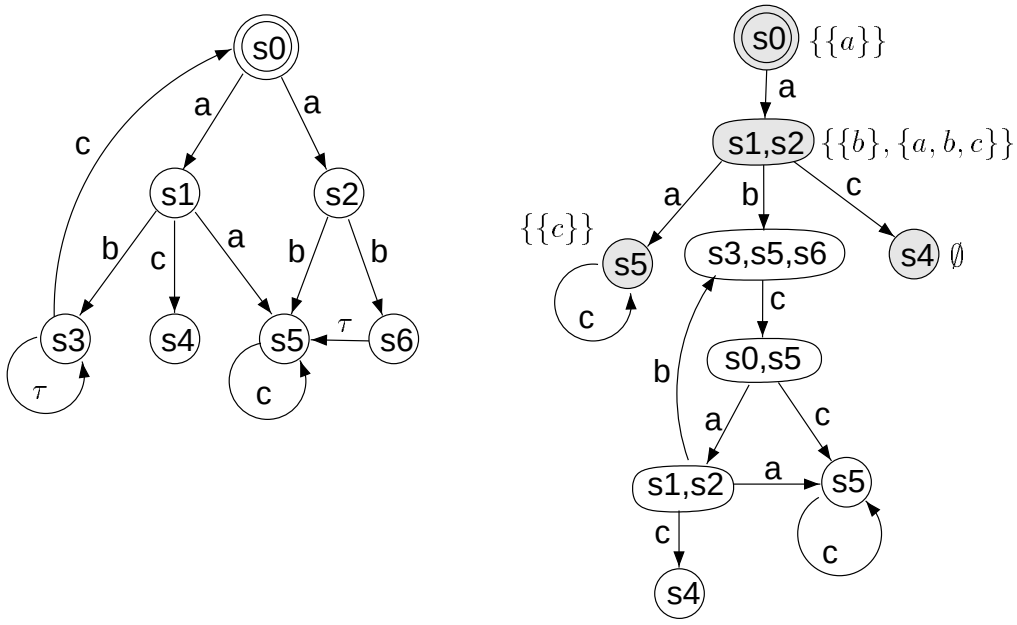
Testing Equivalence via Bisimulation

The acceptance graph P' is an annotated deterministic process obtained from process P as follows:

- Every state in P' is a **set of states** of process P.
- A **visible action** α is possible in the state in P' iff the same action is possible in one of matched states in P.
- The states in P' can be **open** or **closed**. Closed states are annotated with a minimized **acceptance sets**.

The acceptance set of a closed state in P' is a set of actions that are immediately possible from matched states in P.

Two processes are **testing equivalent** if their acceptance graphs are bisimulation equivalent.



Conclusion

- **Testing equivalence** is conceptually simple and close to the concept of testing.
- We made **efficient implementation** of a testing equivalence using Binary Decision Diagrams (BDDs).
- **The algorithm** for testing equivalence is composed of transforming processes into acceptance graphs and then determining bisimulation equivalence between them.
- **Possible extensions:**
  - computing testing preorders,
  - system abstraction due to testing equivalence,
  - generating diagnostics to explain why two systems are not testing equivalent.
- Presented work is a part of **our verification system** also capable of:
  - changing the behaviour of processes,
  - performing parallel composition of processes,
  - determining bisimulation equivalence between processes,
  - reasoning about properties of processes using model checking with ACTL.