SoftCOM 2000

8th International Conference Software, Telecommunications
and Computer Networks, Split, Croatia

THE EFFICIENT SYMBOLIC TOOLS PACKAGE

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

Contents:

THE EFFICIENT SYMBOLIC TOOLS PACKAGE

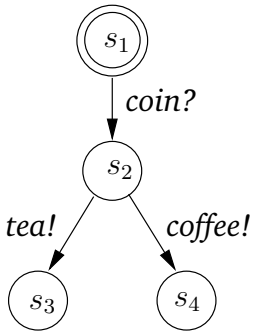
Introduction

- **Process algebras** are widely used formalisms in the verification of concurrent systems.
- A process is an entity which **performs actions** and **transits** between its internal states.
- Processes **synchronise** with each other by simultaneously performing synchronisation actions.

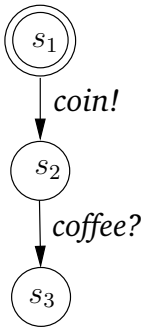
Milner's CCS (Calculus of Communicating Systems, 1980)
$Z = \tau \cdot (\alpha(x) \cdot \overline{\delta}(0) + \beta(x) \cdot \overline{\delta}(1)) \cdot Z$

Hoare's CSP (Communicating Sequential Processes, 1985)
$T = left\,?\,x \longrightarrow right\,!\,0 \longrightarrow right\,!\,1 \longrightarrow T$

Simple process algebra



A drink machine

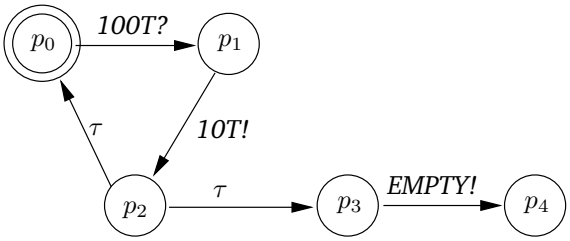


A user

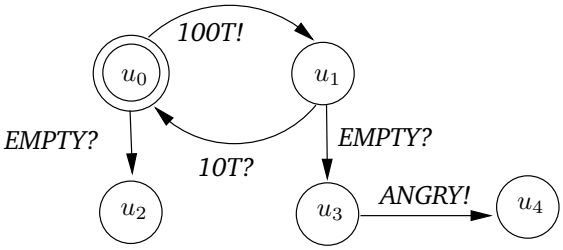
```
PROCESS Drink_Machine
INITIAL STATE s1
TRANSITIONS s1 = coin?.s2
            s2 = tea!.s3
            s2 = coffee!.s4

PROCESS User
INITIAL STATE s1
TRANSITIONS s1 = coin!.s2
            s2 = coffee?.s3
```

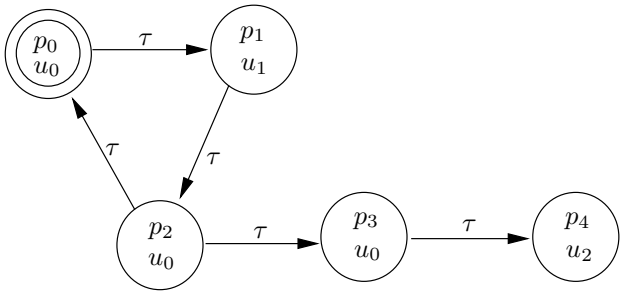
Parallel composition



AUTOMAT

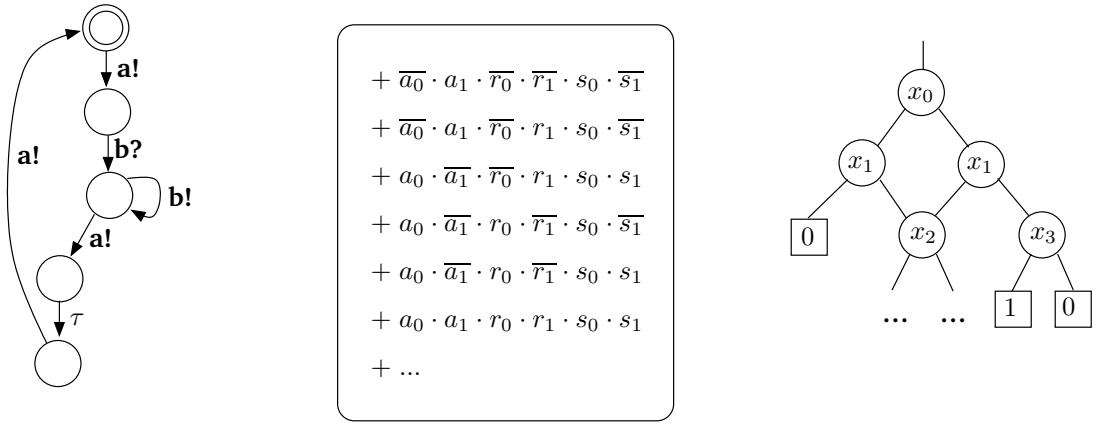


USER



Parallel composition

Symbolic verification



- States and transitions are encoded by Boolean functions rather that being explicitly enumerated.
- Further, Boolean functions are represented with binary decision diagrams (BDDs).
- Thus, operations on processes are performed as operations with Boolean functions, which are actually performed as manipulations with BDDs.

Verification Methods

- By equivalence testing:
  - trace equivalence,
  - observational equivalences,
  - testing equivalence.

- By model checking:
  - with ACTL,
  - with μ-calculus.

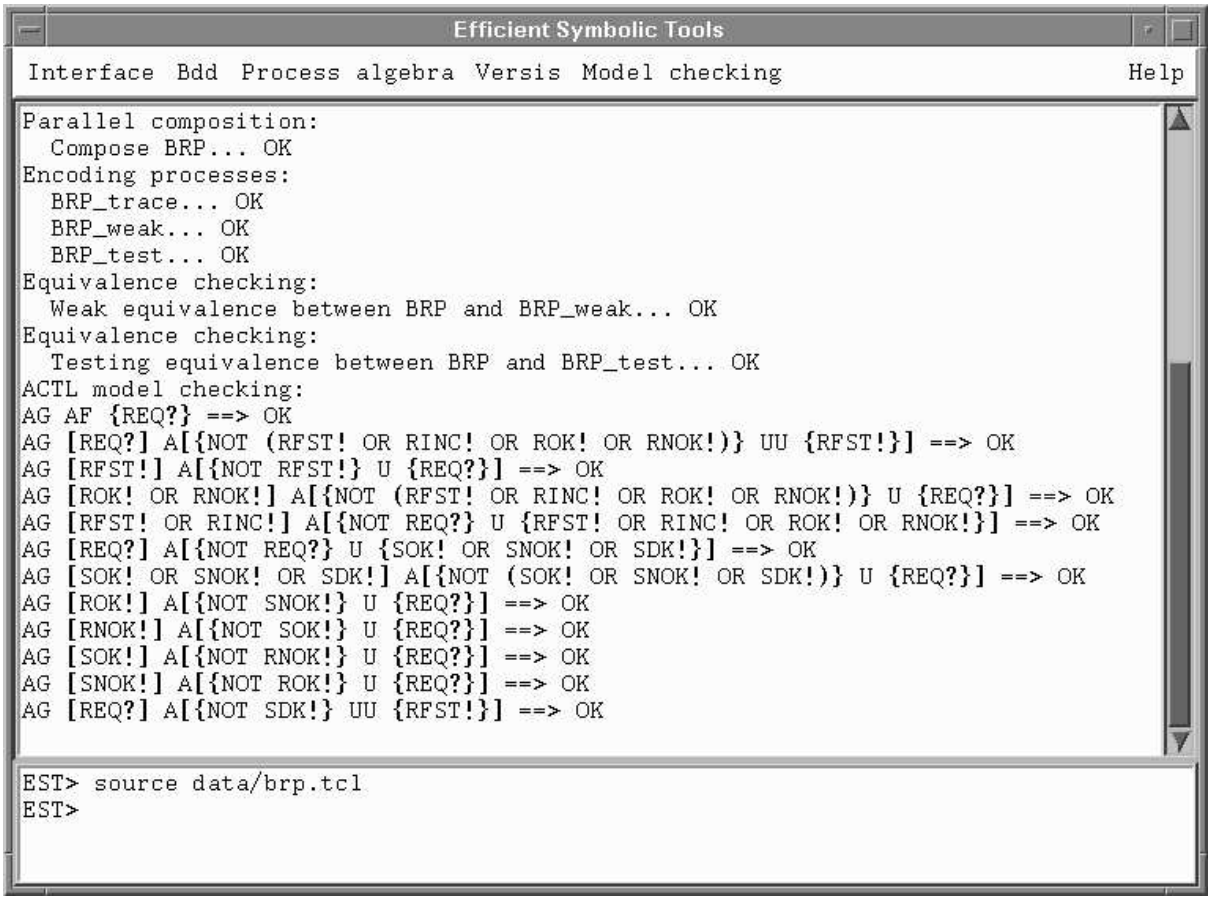
EST - Efficient Symbolic Tools

- EST is a new tool for the verification of concurrent systems, which has not been widely presented yet.
- EST is a relatively small package written in C with a Tcl/Tk user interface.
- Main advantages: flexibility, portability and an efficient memory management.

EST is a modularized package:

- **Binary Decision Diagrams** module is a general purpose BDD package for the manipulation of Boolean functions,
- **Process Algebra** module is a framework for representing processes,
- **Versis** module implements operations on processes,
- **Model Checking** module provides functions for ACTL model checking,
- **My Interface** module implements the user interface.

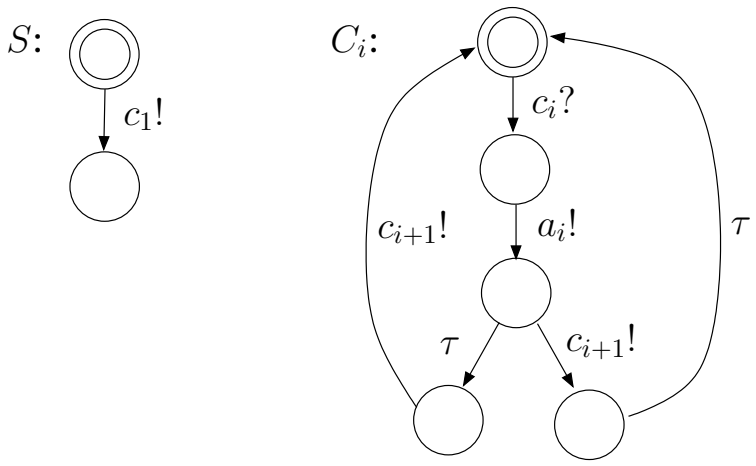
An overview of EST



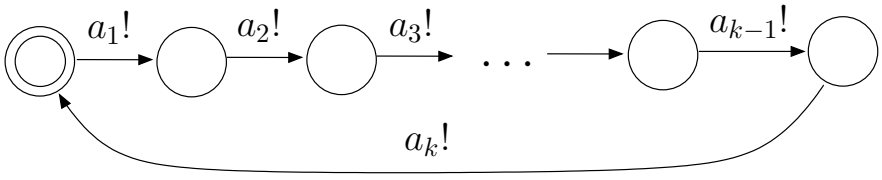
An example of verification

Milner's simple distributed scheduler consists of

- starter S,
- cyclers C_i.



External behaviour of the system with k cyclers



The results of verification

- HP 715/100 with 128 MB RAM,
- The program was allowed to have at most 500000 BDD nodes at once, so that the total memory consumption never exceeded 32 MB.
- 7 times better results on an overclocked Pentium II 266 with Linux.

k	**states**	**transitions (without τ)**	**nodes in BDD**	**parallel composition**	**weak obs. equivalence**	**testing equivalence**
4	97	241 (32)	272	0.5s	0.3s	0.4s
8	3073	13825 (1024)	707	2.1s	0.8s	1.3s
12	73729	479233 (24576)	1245	7.1s	1.6s	2.9s
16	1572865	13369345 (524288)	1912	19.5s	2.8s	4.9s
20	31457281	(10485760)	2674	47.5s	4.1s	8.2s
24	603979777	-	3548	97.8s	5.8s	12.8s
28	-	-	4534	205.1s	8.0s	18.7s
32	-	-	5665	452.8s	10.6s	26.9s

Conclusion

- EST project started in 1992,
  see `http://www.el.feri.uni-mb.si/est/`

- EST distinguishes itself as a small and efficient package with an easily readable source code and well implemented algorithms.

- EST uses symbolic methods to represent and manipulate processes.

- EST has already been successfully used for the verification of some larger concurrent systems, for example the problem of simple crossing of a road and a railway and formal verification of bounded retransmission protocol.

- **The future work:**
  - diagnostic (witnesses, counterexamples),
  - introduction of explicit data-passing.