

# STRUCVAL (simple examples for StrucVal approach, i.e. structure-driven validation of systems)

(Robert Meolic, 2004, 2013)

This is a collection of small examples to test the algorithms for structure-driven validation of systems.

The following algorithms are implemented as part of model checking module:

- Witness and counterexample automaton (WCA) generation

The following algorithms are implemented as part of versis module:

- Witness and counterexample automaton (WCA) minimisation

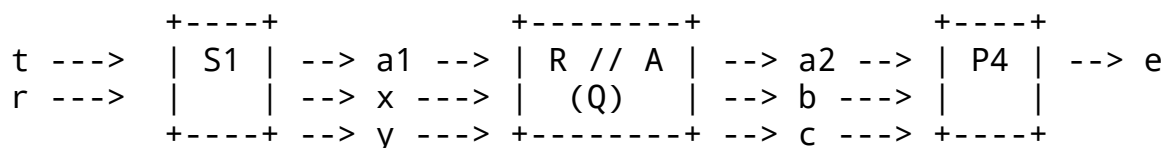
The following algorithms are implemented as part of strucval module:

- SyncProduct - synchronous product of an LTS and a finite automaton
- ReduceStates - reduces states of synchronous product (bottom-up approach)

The following operators are supported in the CCS parser.

- // - CCS parallel composition  
all actions sync and non-sync
- || - LOTOS synchronisation  
only tau goes alone
- ||| - LOTOS interleaving  
all actions go alone, only
- automaton WCA\_TOPSYS = F1 @ TOPSYS  
generate WCA for formula F1 on process TOPSYS
- product QA = (Q \* A)\a\b\c  
create synchronous product between process Q and WCA A (a,b,c are internal actions)

File **topsys.ccs** describes the following system (it is a chain):



Here is the description:

```

agent S1 = (?t.!x.!y.S1 + ?r.!a1.!x.?t.!y.S1)
agent R = ?a1.!a2.R
agent A = ?x.!b.?y.!c.A
net Q = (R // A)
agent P4 = (?a2.?b.P40 + ?b.(?b.P40 + ?c.P41))
agent P40 = !e.?a2._P41
agent P41 = ?b.P4
net TOPSYS = (P4 // S1 // Q)\a1\a2\b\c\x\y

```

For this test we are using ACTL formula  $EF \ EX \ \{e!\} \ \text{true}$ .

## Here is the log from EST:

Efficient Symbolic Tools, 2nd Edition, Copyright (C) 2003-2013 UM-FERI  
This is free software, and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute EST; type "license" for details.

Running on i686 (Linux, 3.2.0-39-generic-pae) with tcl 8.5.11 and tk 8.5.11.

Initialization of GUI package... OK  
Initialization of BDD package... OK  
Initialization of Process\_Algebra package... OK  
Initialization of Versis package... OK  
Initialization of Model checking package... OK  
Initialization of Strucval package... OK  
Initialization of CCS package... OK  
Ready.

```
>cd "/home/meolic/est/est-2ed/data/strucval"; source "topsys.tcl"; cd "/home/meolic/est/est-2ed/data"
```

```
Reading file: topsys.ccs
  Parallel composition SORT ...
    WARNING: Added empty process STOP ...OK
  Process S1 ... OK
  Process R ... OK
  Process A ... OK
  Net Q
    Composition ... OK
    Creating process Q ... OK
  Process P4 ... OK
  Process P40 ... OK
  Process P41 ... OK
  Net TOPSYS
    Composition ... OK
    Creating process TOPSYS ... OK
  Property F1
    F1 = EF EX{e!} true;
  Automaton WCA_TOPSYS
    Creating automaton F1 @ TOPSYS ...
    EF EX{e!} true ==> TRUE
    Witness automaton WCA_TOPSYS has been constructed.
  Automaton WCA_P4
    Creating automaton F1 @ P4 ...
    EF EX{e!} true ==> TRUE
    Witness automaton WCA_P4 has been constructed.
  Product WCA_Q
  Product WCA_S1
```

### STATISTICS ABOUT PROCESSES

PROCESS S1  
Number of states: 7  
Number of transitions: 8

PROCESS Q  
Number of states: 8  
Number of transitions: 16

PROCESS P4  
Number of states: 6  
Number of transitions: 8

PROCESS TOPSYS  
Number of states: 63  
Number of transitions: 83

### STATISTICS ABOUT WCAs

PROCESS WCA\_TOPSYS  
Number of states: 20  
Number of final states: 6  
Number of transitions: 22

PROCESS WCA\_P4  
Number of states: 7

Number of final states: 1  
Number of transitions: 9

PROCESS WCA\_Q  
Number of states: 16  
Number of final states: 4  
Number of transitions: 20

PROCESS WCA\_S1  
Number of states: 20  
Number of final states: 6  
Number of transitions: 22

#### EQUIVALENCE CHECKING

Strong observational equivalence checking between WCA\_TOPSYS and WCA\_S1... OK

AND NOW... THE SAME BUT STEP-BY-STEP

ACTL/ACTLW model checking on process TOPSYS  
EF EX{e!} true ==> TRUE  
Witness automaton WC1TOPSYS has been constructed.  
Minimization of automaton WC1TOPSYS... OK

ACTL/ACTLW model checking on process P4  
EF EX{e!} true ==> TRUE  
Witness automaton WC1P4 has been constructed.  
Minimization of automaton WC1P4... OK

Product Q1... OK  
WCA minimization of Q1... OK  
Product Q2... OK  
WCA minimization of Q2... OK  
Strong observational equivalence checking between MIN\_WC1TOPSYS and MIN\_Q2... OK