TESTOR: A Modular Tool for On-the-Fly Conformance Test Case Generation

Lina Marsso, Radu Mateescu,

and Wendelin Serwe

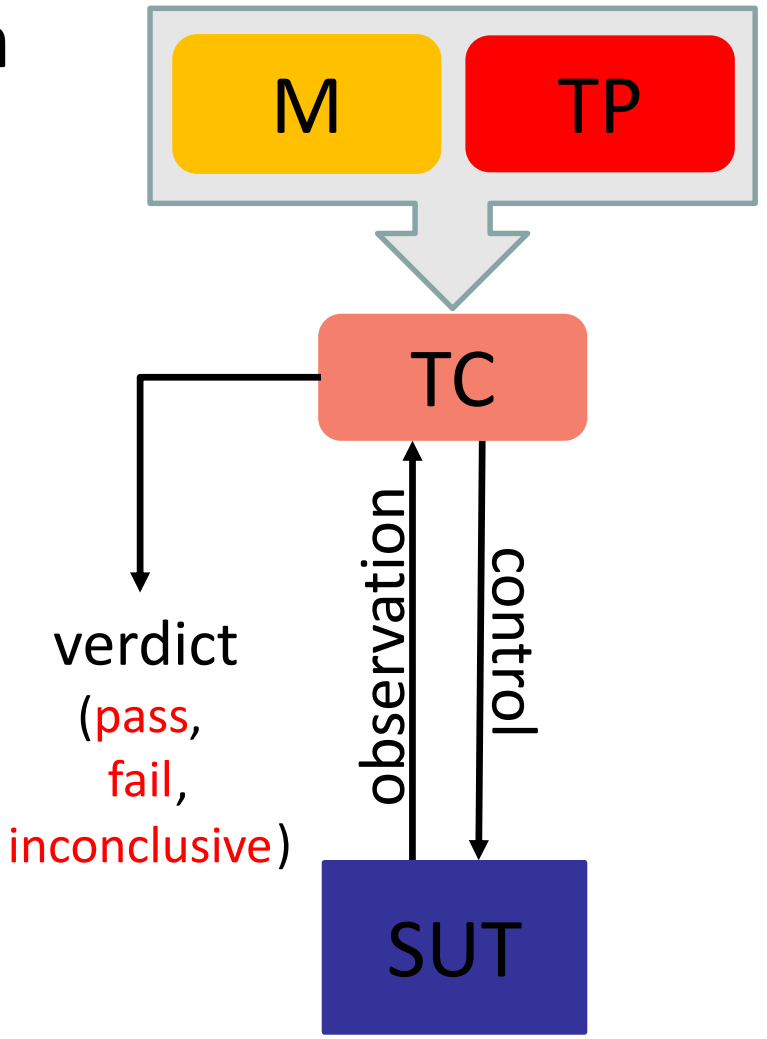
Team **CONVECS (LIG / Inria)**

Outline

- Conformance Testing with Test Purposes
- TESTOR: Overview
- Related Work
- Experimental Evaluation
- Conclusion

Conformance Testing (with Test Purposes)

- Check *conformance* between
  - formal **model** (M)
    and **test purpose** (TP)
  - **system under test** (SUT)
- Test purpose (TP):
  functionality to be tested
- Test case (TC):
  control the SUT
- Verdicts:
  - *fail*: SUT not conform to M
  - *pass*: no error
  - *inconclusive*: no error, but TP not reached



Formal Models: IOLTS

Input-Output Labeled Transition System (**IOLTS**)
(Q, A, T, q_0)

- Q: set of states
- $A = A_i \cup A_o \cup \{\tau\}$: set of actions
  - A_i: **input** action, controllable by the tester (“?”)
  - A_o: **output** action, observable by the tester (“!”)
  - τ: **internal**, unobservable action
- $T \subseteq Q \times A \times Q$: transition relation
- $q_0 \in Q$: initial state

Conformance Relation: ioco

- Observe suspended execution traces of the SUT
- **Suspended trace**: execution up to **quiescence**
- **Quiescence** (δ):
  - **deadlock**: state without successors
  - **outputlock**: state without outgoing output actions
  - **livelock**: cycle of internal actions
- SUT **ioco** M [Tretmans-96]
  if after each suspended trace, SUT exhibits only outputs and quiescences present in M

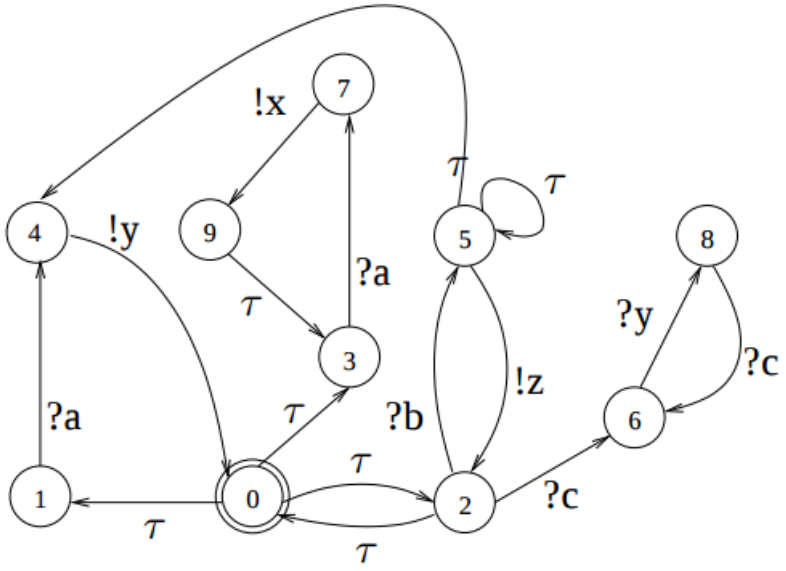
Test Purpose (TP)

- **Deterministic** and **complete** (each state offers all actions) **IOLTS**
- Same action alphabet as M
- Special states
  - Accept states to select desired behaviors
  - Refuse states to cut the exploration of M
- Special transition $q \xrightarrow{*} q'$ matches actions not occurring on any other transition leaving q
- Implicit completion with transitions $q \xrightarrow{*} q$

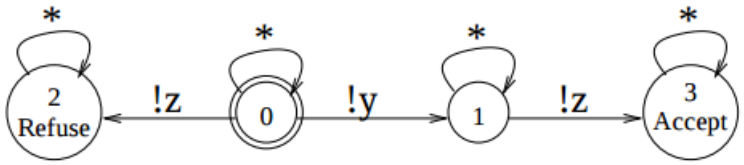
Test Case (TC)

- **IOLTS** with verdict states (pass, fail, inconclusive)
  - from all states, a verdict is reachable
  - fail/inconclusive directly reachable only by outputs
  - no internal actions
- **Controllable:** no choice between two inputs or an input and an output
- **Abstract:** connection to the SUT not provided
- **Complete Test Graph** (CTG)
  - union of all TCs
  - not necessarily controllable

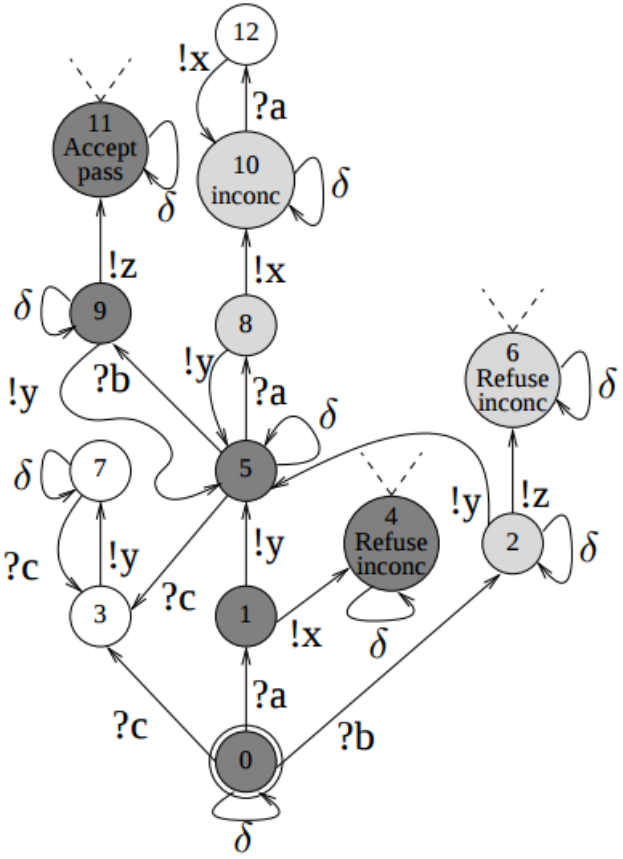
Example



(a) model M

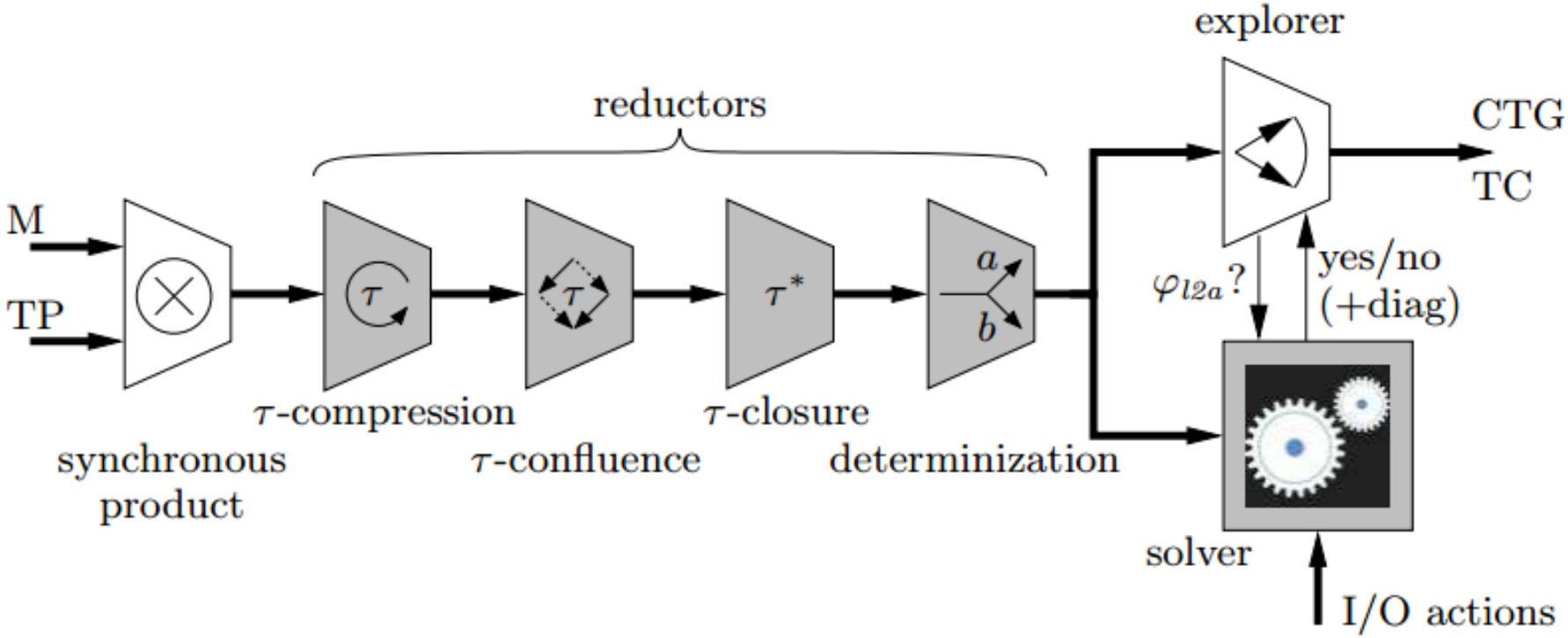


(b) test purpose TP



(c) visible behaviour SP_{vis}, complete test graph CTG (grey), and a test case TC (dark grey)

TESTOR: Architecture



gray components: OPEN/CAESAR libraries of CADP [Garavel-98]
white components: newly developed
(5022 lines of C and **1106** lines of shell script)

CADP (http://cadp.inria.fr)

- Construction and Analysis of Distributed Processes
- Modular toolbox with several
  - Formal specification languages:

    LOTOS, LNT, FSP, π-calculus
  - Verification paradigms:

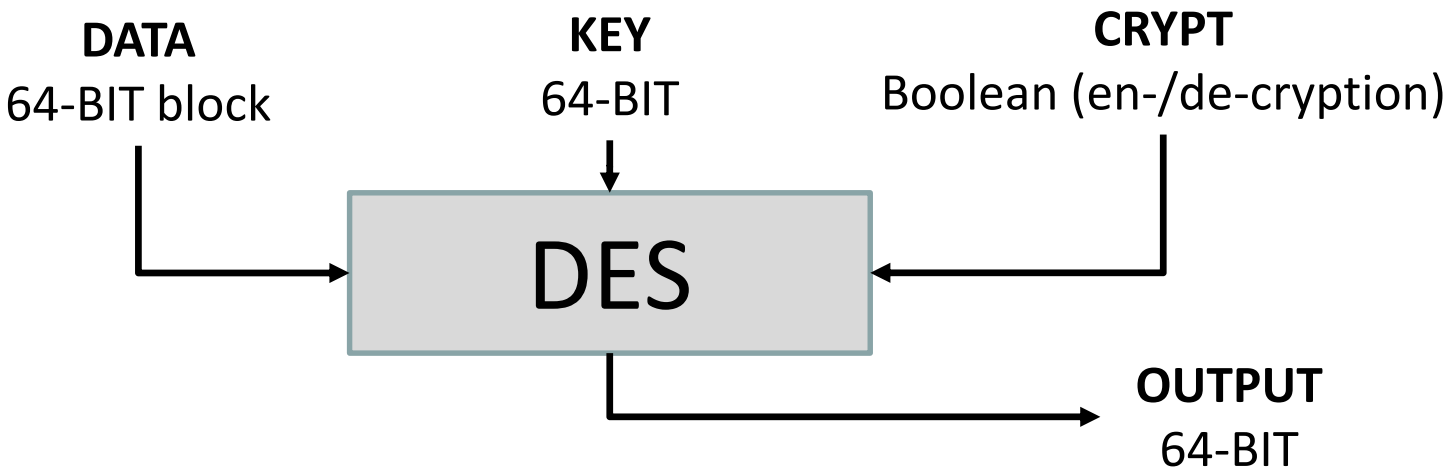
    model checking, equivalence checking, visual checking
  - Analysis techniques:

    reachability, on-the-fly, compositional, distributed, static analysis, code/test generation, performance, evaluation
- Continuous development for more than 25 years
- Many case-studies and 3rd party tools

LNT: "User-friendly" Language

- A safe language for message-passing concurrent systems
- A synthesis between three paradigms:
  1) Process calculi
     nondeterministic choice, asynchronous parallel composition, multiway rendez-vous, disruption
  2) Functional languages
     types defined by free constructors, pattern matching
  3) Imperative languages
     structured programming construct (**if**, **while**, **for**, **case**, etc.), assignments, **in**/**out** parameters, Ada-like syntax for readability
- Supported by CADP: compilers, model-checkers, etc.

DES (Data Encryption Standard)



- Asynchronous implementation in **LNT**
- 16 iterations of the same cipher function
  - each iteration: 48-BIT subkey (64-BIT KEY)
- Test purpose: sequence of an encryption of a data block
  - **DATA** = 0x0123456789abcdef
  - **KEY** = 0x133457799bbcdff1
  - **OUTPUT** = 0x85e81350f0ab405

Simple TP for the DES (1/4)

```
Process PURPOSE1 [CRYPT: CB, KEY, DATA, OUTPUT: C64, SUBKEY: C48,
                  T_ACCEPT, T_REFUSE, OTHERWISE: none] is
   CRYPT (true);
   KEY (C_13345779_9bbcdff1);
   DATA (C_01234567_89abcdef);
   OUTPUT (C_85e81354_0f0ab405);
   loop T_ACCEPT end loop
end process
```

- Sequence of 3 inputs followed by an output
- But:
  - TP completed with special transitions $q \xrightarrow{*} q$
  - More complex TC than expected
  - CRYPT(TRUE); CRYPT(FALSE); …

Simple TP for the DES (2/4)

```
Process PURPOSE2 [CRYPT: CB, KEY, DATA,
OUTPUT: C64, SUBKEY: C48, T_ACCEPT,
T_REFUSE, OTHERWISE: none] is
   select  -- refuse any rendez-vous
           -- but ''CRYPT (TRUE)''
      CRYPT (true)
   [] OTHERWISE; loop T_REFUSE end loop
   end select;
   select  -- refuse any rendez-vous
           -- but 'KEY (C_13345779_9bbcdff1)''
      KEY (C_13345779_9bbcdff1)
   [] OTHERWISE; loop T_REFUSE end loop
   end select;
   …
end process
```

- Explicitly complete the TP
- OTHERWISE: match special label *
- T_REFUSE: cut undesirable behavior

Simple TP for the DES (3/4)

- \+ Multiway-rendezvous 
  - replace synchronous product by parallel composition
  - compositional annotation of the model
  - cut undesired branches: LNT operational semantics

- ≈ Test purpose 2 (LNT parallel composition):

```
par CRYPT, KEY, DATA, OUTPUT in
   DES [CRYPT, KEY, DATA, OUTPUT, SUBKEY]
|| PURPOSE1 [CRYPT, KEY, DATA, OUTPUT, T_ACCEPT]
end par
```

Simple TP for the DES (4/4)

Multiway rendezvous enables Data handling !

```
process PURPOSE3 [CRYPT: CB, KEY, DATA,
                  OUTPUT: C64, T_ACCEPT: none] is
  var  C: BOOL, D, K: BIT64 in
    CRYPT (?C);
    KEY (?K);
    DATA (?D);
    OUTPUT (DES(C, K, D));
    loop T_ACCEPT end loop
  end var
end process
```

Model-Based Testing Tools

- MBT tools using the **ioco** conformance relation
- MBT tools using symbolic test generation
- MBT tools for synchronous models
  - Gatel
  - JTorX
  - Lutess
  - Lurette
  - STG
  - TGV
  - TorX
  - TorXakis
  - T-Uppaal
  - Uppaal-Cover
  - Uppaal-Tron
  - Uppaal-Yggdrasil

TGV

- Conformance test generation with test purposes
- TESTOR : reimplementation of TGV's approach
- Enhancements brought by TESTOR:
  - on-the-fly computation of a controllable test case
  - modular architecture based on existing libraries
  - flexible specification of accepting/refusal states
    - dedicated synchronous product (similar to TGV)
    - LNT parallel composition and multiway rendezvous: data handling test purposes

Experimental Evaluation

- TESTOR **correctness** using bisimulation checking:
  - each **TC** is **included** in the **CTG**
  - **compared** TCs & CTG generated by TESTOR & TGV
- Academic examples and realistic case studies
- Test purposes:
  - taken from case studies
  - automatically generated
- Experiments carried out using Grid'5000
- Runtime+memory, average of 10 executions

TP taken from Case-Studies

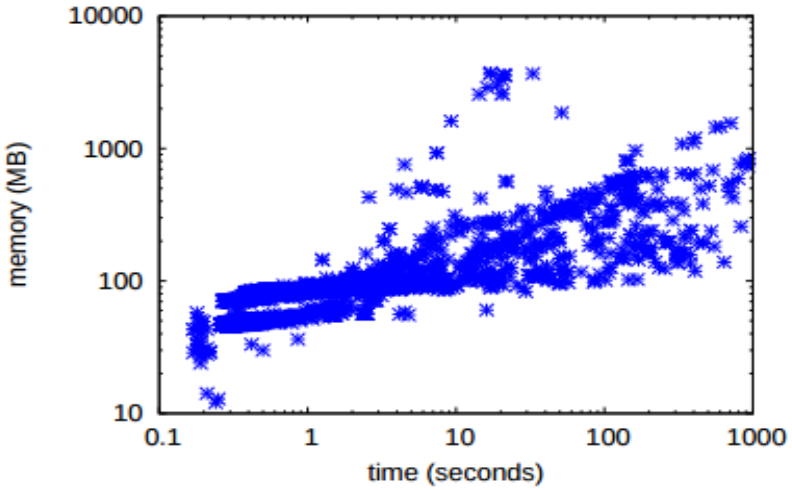
example	TESTOR				TGV			
	test case		CTG		test case		CTG	
	time	mem.	time	mem.	time	mem.	time	mem.
EnergyBus	3	81	182	181	2	137	52	858
EnergyBus (with REFUSE)	1	67	1	66	0	66	0	43
ACE UniqueDirty	45	121	346	451	75	159	3047	643
ACE SharedDirty	384	510	342	529	3821	746	3920	746
ACE SharedClean	298	415	325	523	2820	628	3474	663
ACE Data Inconsistency	24	116	580	711	24	142	6701	894
DES (PURPOSE1)	22109	300	>1week		>43GB		>220GB	
DES (PURPOSE2)	27344	332	27	86	24	6177	24	6176
DES (PURPOSE3)	2	74	4	100	not applicable			

Execution time is given in seconds and memory usage in MB.

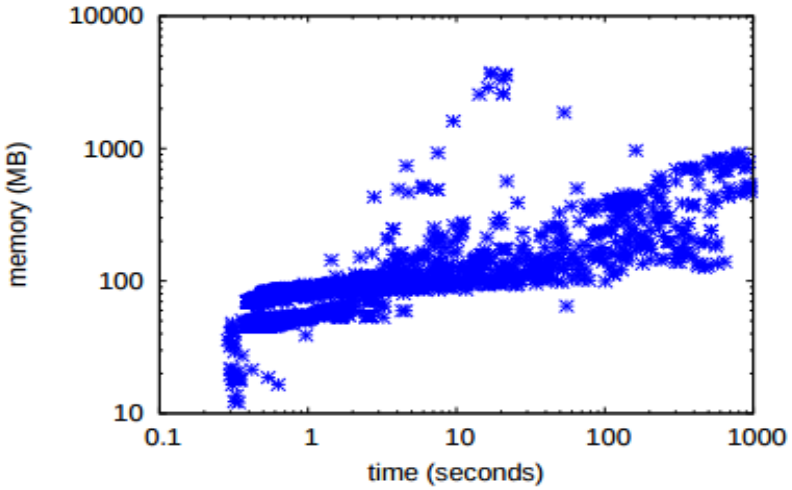
TP Automatically Generated (1/2)

- 9791 **LTSs** with **<= 50 million** transitions
  (from non-regression test-base for **CADP**)
- Automatically generate **2 TPs** for each LTS:
  1. reachability of an action
     (first action, alphabetically)
  2. presence of an execution sequence
     (extracted with **EXHIBITOR**, **<= 1000** visible **actions**)
- Discard the pairs (M, TP) for which
  - automatic generation of test purpose (TP) fails
  - computation (of TC or CTG) is too expensive

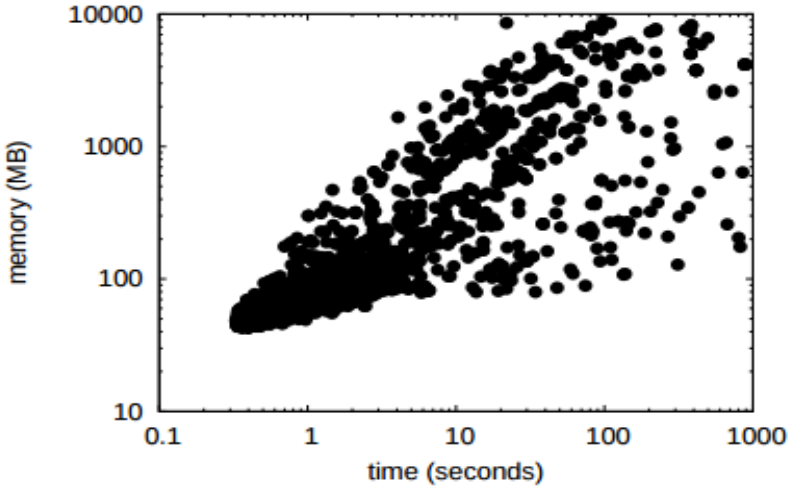
TP Automatically Generated (2/2)



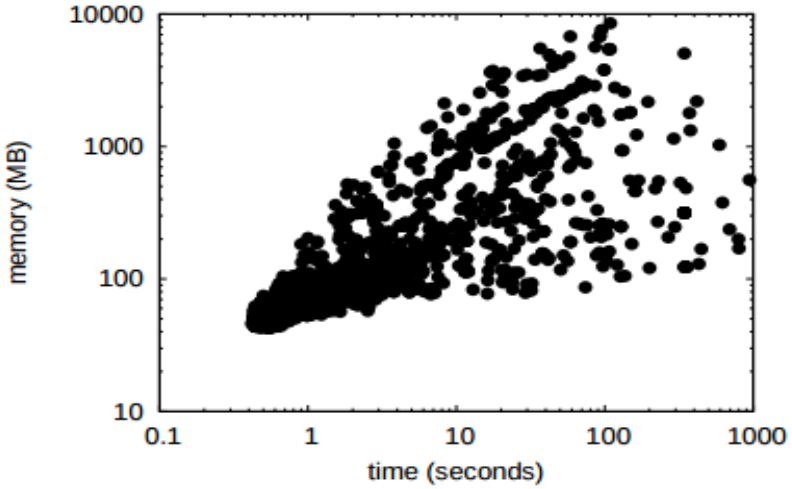
(a) test case (TESTOR)



(b) complete test graph (TESTOR)



(c) test case (TGV)



(d) complete test graph (TGV)

Conclusion

- Contributions
  - online conformance testing using on-the-fly test case generation directed by a test purpose
  - **TESTOR** tool with a modular architecture based on OPEN/CAESAR components of **CADP**
  - versatile specification of test purposes using LNT and the multiway rendezvous
- Future work
  - improve performance: state space caching, …
  - derive test purposes from temporal logic properties