On optimizing computation of semi-block simulation

Igor Konnov

Moscow State University

Faculty of Computational Mathematics
and Cybernetics

INTAS Meeting, Kiev, May 2008

Outline

- Semi-block simulation in Parameterized Model Checking
- Our previous results on using semi-block simulation
- Ways to improve the algorithm
- Comparison to BDD-based algorithm
- Thoughts on using SigRef tool

Parameterized Model Checking

- We study the verification problem for families of distributed systems $\{M_n\}$, $n \geq 1$
- Every system M_n is composed of some distinguished process Q and a number of isomorphic processes that are instances of the same prototype process P: $M_n = Q \,||\, P \,||\, P \,||\, \ldots \,||\, P$.
- *In general, there may be several prototypes*

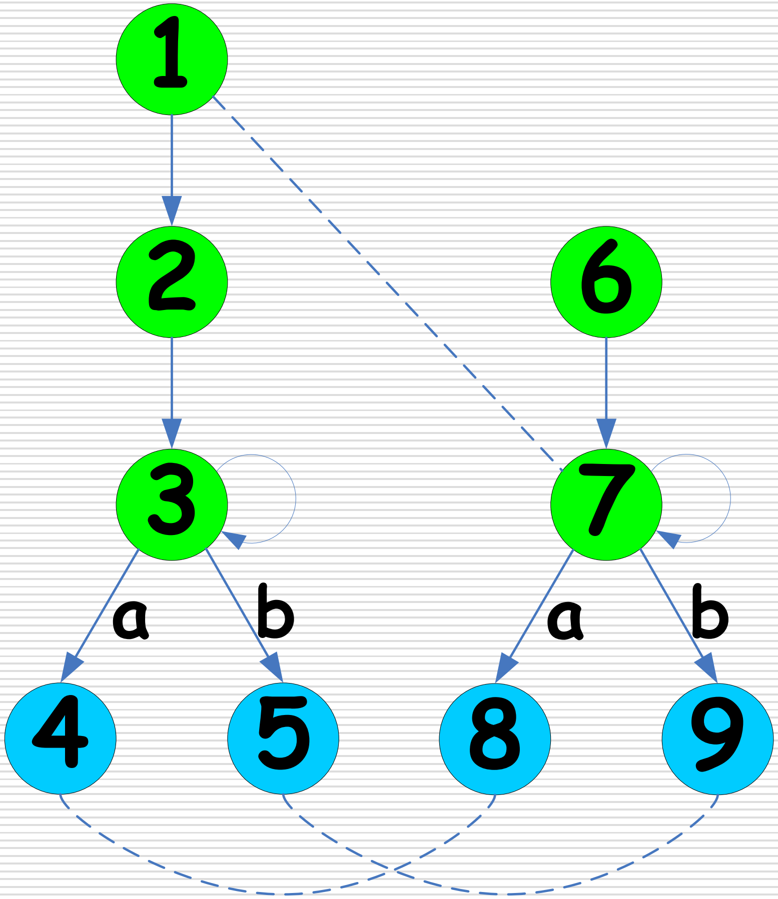
PMC by invariants

- Previously, we proposed several relations on LTSes: quasi-block simulation, block simulation and semi-block simulation
- *Schematic view*: to check that $M_n \models S$ holds for every n it is sufficient to find LTS I *(invariant)* such that $Q \,||\, P < I$ and $I \,||\, P < I$, hold, and check that $I \models S$
- We use framework of network invariants by Clarke, Grumberg and Jha omitting the step of abstraction

Previous results

Semi-block simulation

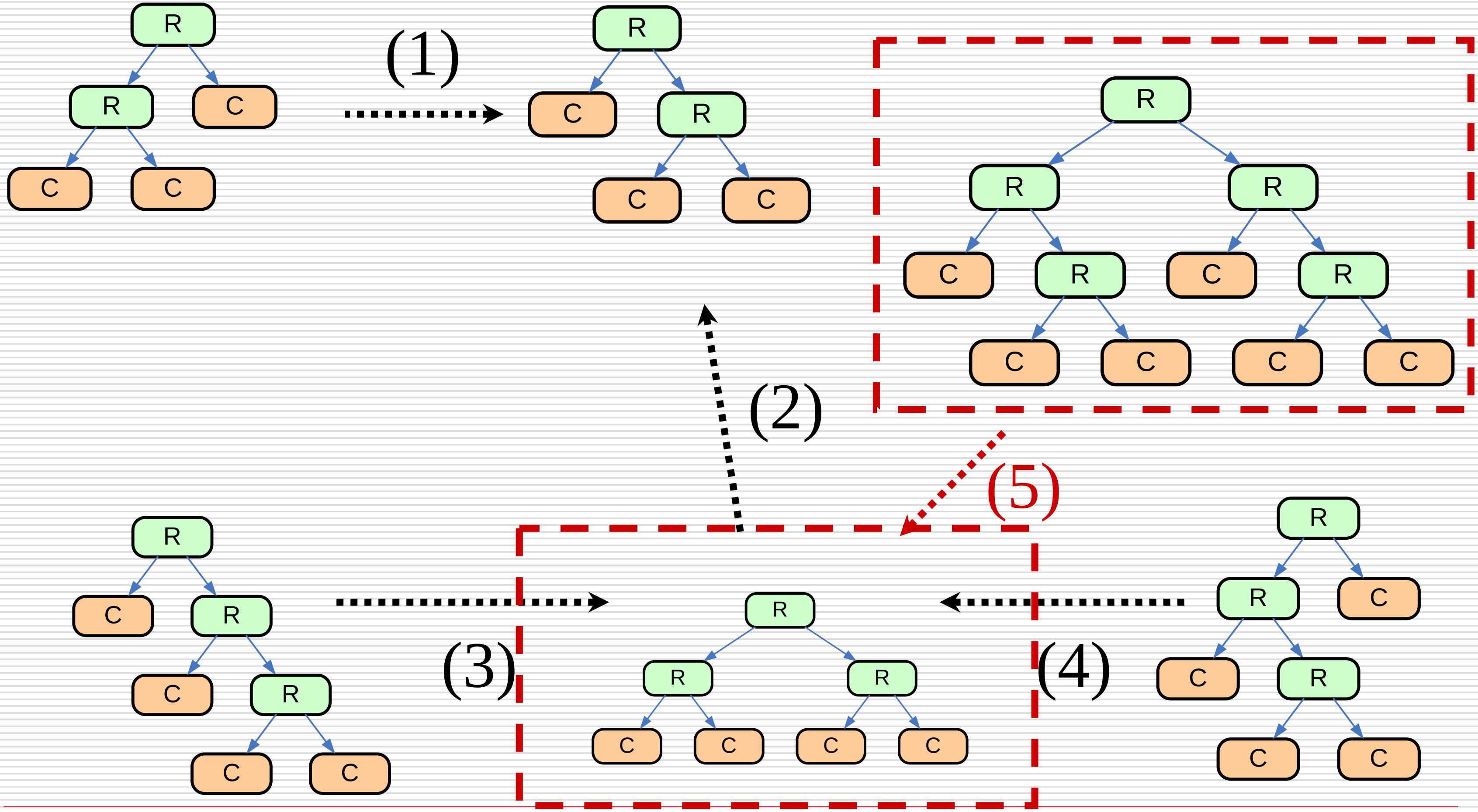
- The relation of semi-block simulation is a key relation to find an invariant.
- It should be built as fast as possible.



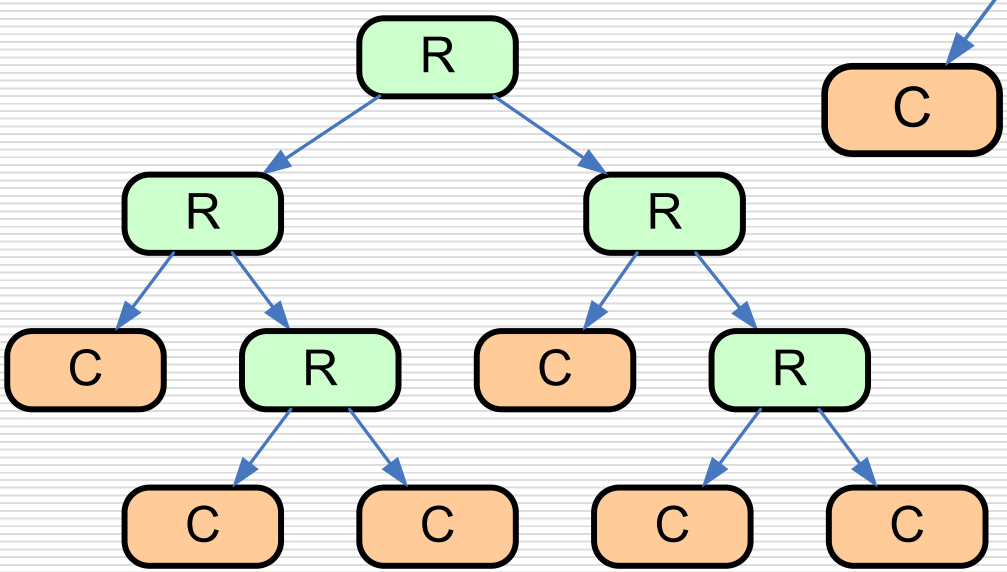
Resource Reservation Protocol

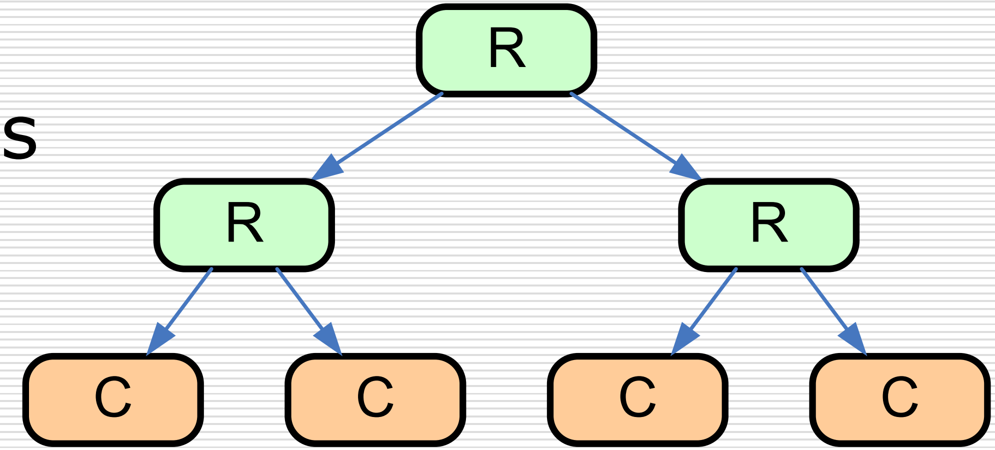
- RFC 2205 defines RSVP protocol, which allows to reserve bandwidth capacity on a route between sender (producer) and receiver (consumer of resources).
- In previous report details of RSVP model and its verification were shown.
- Each model is described in Promela (the language of Spin Model Checker)

Models to compare while finding invariants



Smaller model behaves at least as a bigger model





Relation (5)

Size of Models (Reachable States)

- Models with 2 routers and 3 consumers: 1277, 1732 states
- Models with 3 routers and 4 consumers: 14672, 21659, 24993 states
- Model with 5 routers and 6 consumers: 3816729 states
- It is difficult to count transitions, as they are built on-the-fly

Previous results on building semi-block simulation on RSVP

#	# Pairs in rel.	Time	Memory (DFA)
(1)	15902	2 sec	22M
(2)	223304	30 sec	39M
(3)	1425766	7 min	43M
(4)	$3.8 * 10^6$	20 min	44M
(5)	$3.5 * 10^8$	72 hrs	49M

Ways to improve the algorithm

Basic version of the algorithm

- Relation is computed iteratively using two sets: positive pairs P and negative ones N
- On each iteration the definition is checked against P under $P \cup N$
- When pairs out of $P \cup N$ are requested, they are added to P
- Initially, the set P contains initial states only and N is empty

Combined state storage: DFA + file

- DFA representation:
  - State are stored as words in minimized layered finite state automaton.
  - It is used in Spin Model Checker.
  - Insertion and deletion: O(n)
  - Membership: O(1)
  - Inefficient enumeration
- File representation:
  - Simple enumeration
  - Inefficient insertion, deletion and membership

Partitioning of positive set *P*

- Some pairs may stabilize, i.e. checking them neither adds new positives, nor disproves existing ones
- Therefore, the set of pairs may be split into *stable* and *unstable* subsets
- Stable subset is not checked until unstable subset is exhausted (and all pairs become stable)

Back propagation of negative results

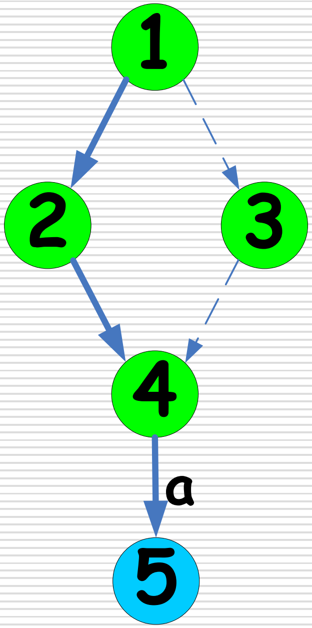
- To reduce a number of pairs to check:
  - check new states,
  - check the states influenced by disproved states.
- We build an over-approximation of states that are potentially disproved by new negatives and check it.
- This approximation helps us to reduce number of iterations.

Cache of pairs

- When checking semi-block simulation for a given pair a lot of pairs is looked up in *P* and *N* several times.
- Caching of queries to *P* and *N* speeds up the overall computation.

Partial Order Reduction

- Partial Order Reduction is a technique to reduce a number of explored paths:
  - A sequence of transitions may be omitted if another sequence leads to the same state and no visible variables were involved in the sequence.
- We have implemented p.o. technique described in:
  - Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled, Model Checking, MIT Press, 1999.



Performance of the optimized version on relation (5)

Techniques	Time	Memory	Disk space
dfa only	27 days	44 M	0G
dfa+file	3 days	44 M	1 G
dfa+file, stable cache, p.o.	1 day	200 M	1 G
dfa+file, stable, back, cache, p.o.	10 h	200 M	1 G

Comparison to BDD-based algorithm

- We have implemented an iterative BDD-based algorithm using CUDD package.
- We need a subset $T_A{}^*$ of transitive closure: $s \to^* s' \to^a t$ (a is visible)
- Computation of $T_A{}^*$ on models in (5) took:
  - more than 9 days,
  - 1G of memory.

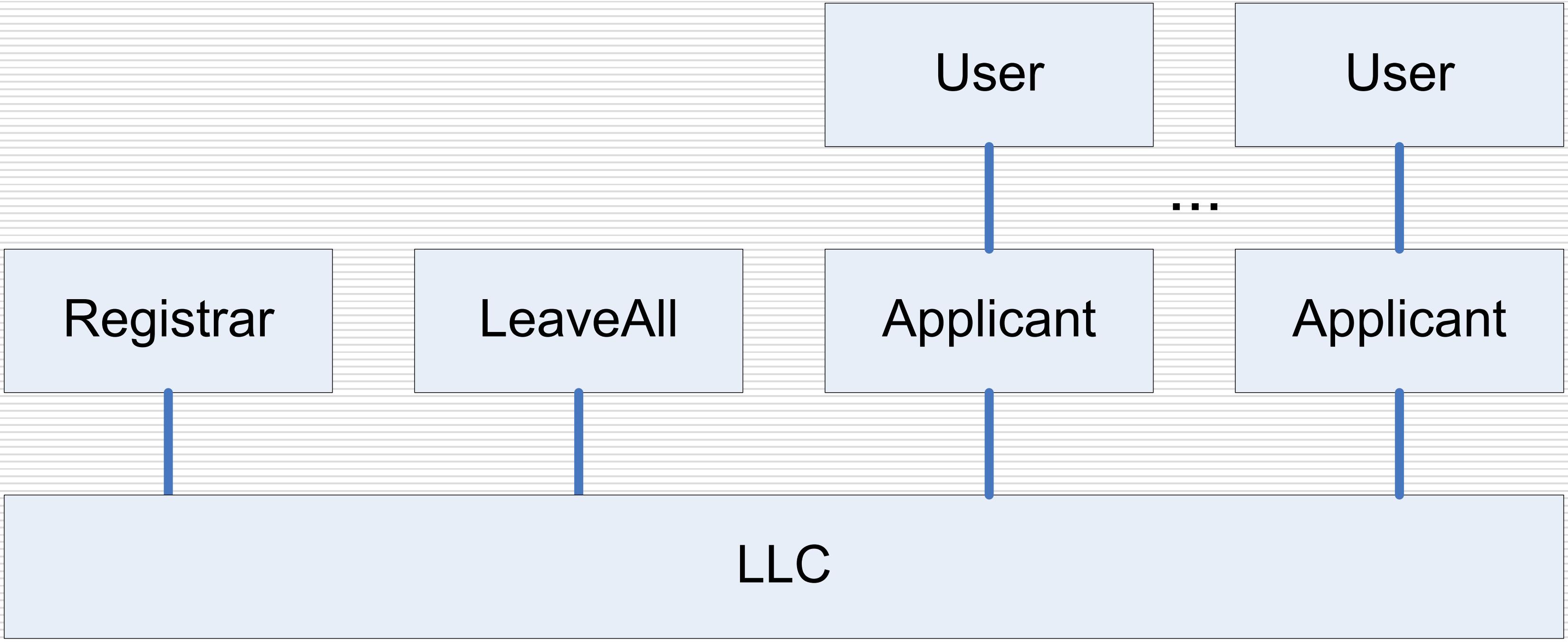
Thoughts on using SigRef tool

- SigRef is a tool for bisimulation minimization.
- Our idea was to find equivalent states in models and check only the representatives of each equivalence class.
- However, minimization step itself was performed on a rather small model (3 routers, 4 consumers) in **1 h 15 min** while our implementation found semi-block simulation on the model in **15 secs**.
- It is a subject to a future research.

Challenge: Model of GARP

- GARP: Group Address Registration Protocol.
- The model in Promela was built by Nakatani:
  - T. Nakatani, "Verification of a Group Address Registration Protocol using PROMELA and SPIN," Proc. Third SPIN Workshop, R. Langerak, ed., Twente Univ., The Netherlands, Apr. 1997.
- This model grows very rapidly in number of states when a number of processes is increased.

Parameterized Model of GARP



Checking GARP

- More than $1.6 \cdot 10^8$ pairs
- More than 16 days

Thank you for your attention!

Upper bounds on complexity

- Time $\leq O(n_{iter} \cdot n_1{}^4 \cdot n_2{}^2 \cdot n_A{}^2)$, where:
  - n_1 – the number of states in the first model,
  - n_2 – the number of states in the second model,
  - n_A – the number of observable actions,
  - n_{iter} – the number of iterations, in the worst case: $n_{iter} \leq n_1 \cdot n_2$.