

Towards an Implementation of Solving Petri Games

Manuel Gieseeking
joint work with Prof. Dr. Bernd Finkbeiner

Group: Correct System Design
University of Oldenburg
AVACS Autumn Plenary 2014

Solving Petri games

Input: A Petri game with an underlying

- **safe** (1-bounded) and
- **concurrency-preserving** ($|\bullet t| = |t\bullet|$ for all transitions t)

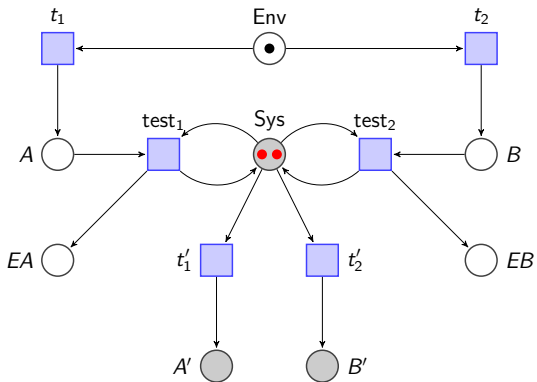
Petri net.

Output: Does a deadlock-avoiding winning strategy exist in the Petri Game? **Yes:** Return all strategies.

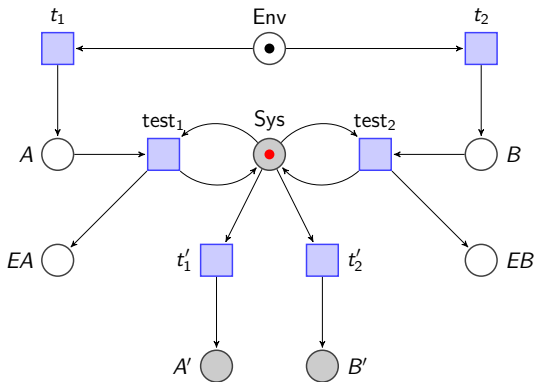
Idea:

- Playing a modified token game in the Petri net.
- States (*decision sets*) are sets of
 - ▶ **places** annotated with a
 - ▶ **set of allowed outgoing transitions** (*commitments*).
- Transitions with an environmental place in its preset (*environment transition*) are only allowed to fire if an mcut is reached.

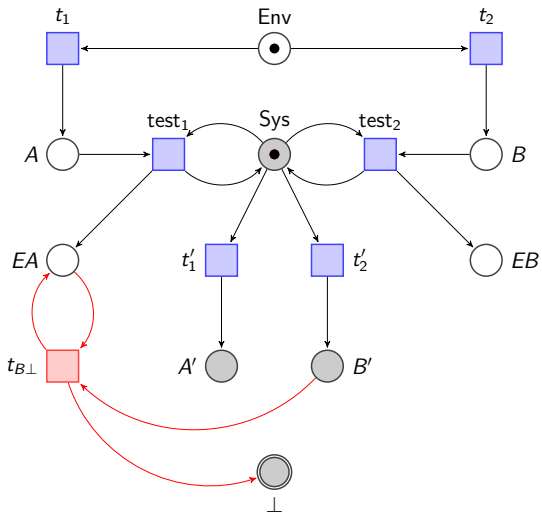
Modified example of [FO14]



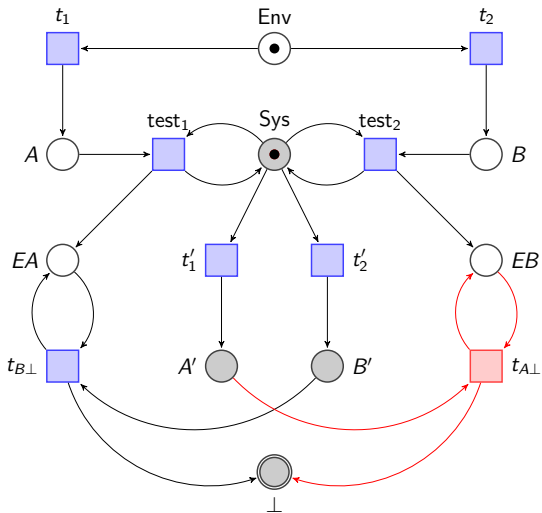
Modified example of [FO14]



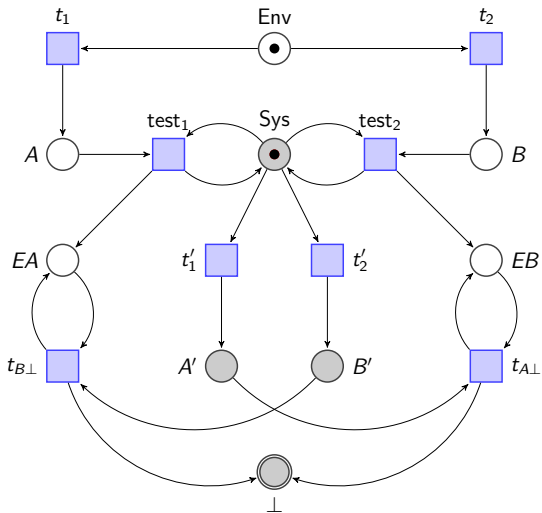
Modified example of [FO14]



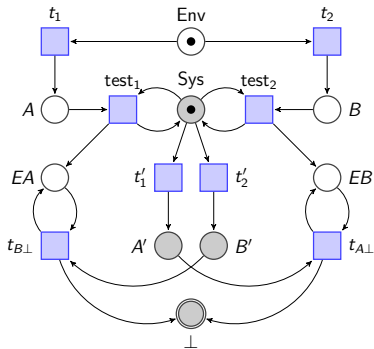
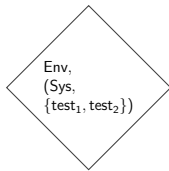
Modified example of [FO14]



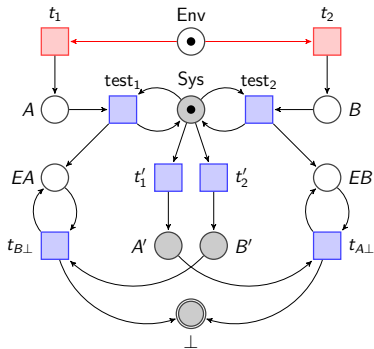
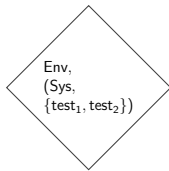
Modified example of [FO14]



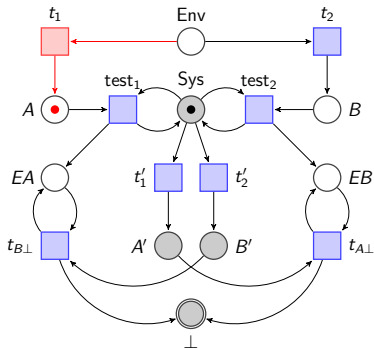
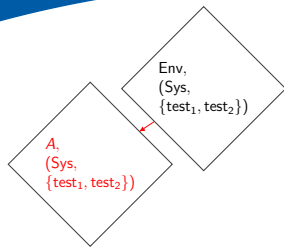
Constructing a finite-graph game [FO14]



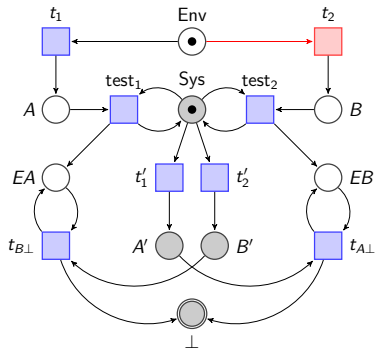
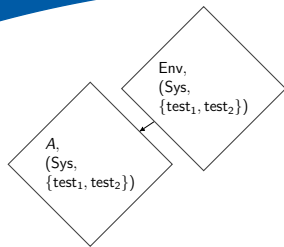
Constructing a finite-graph game [FO14]



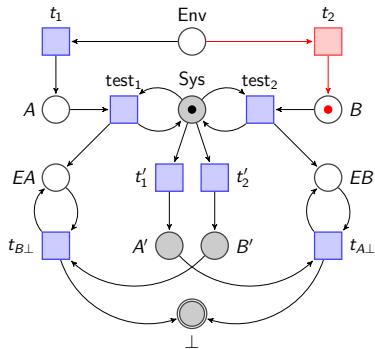
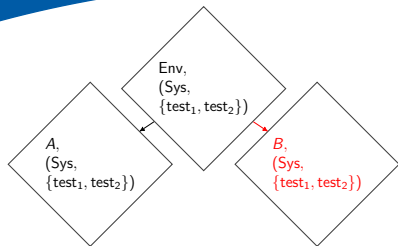
Constructing a finite-graph game [FO14]



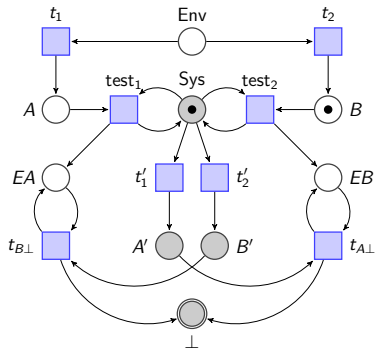
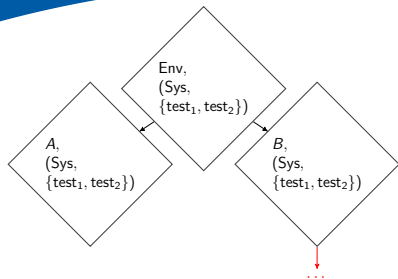
Constructing a finite-graph game [FO14]



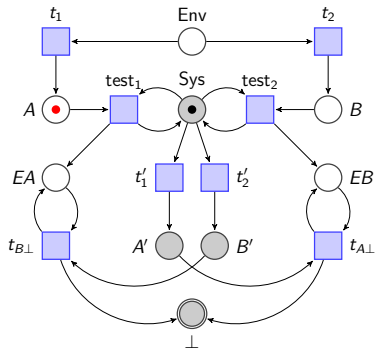
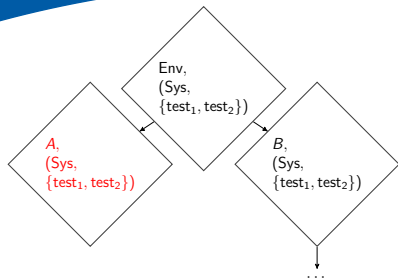
Constructing a finite-graph game [FO14]



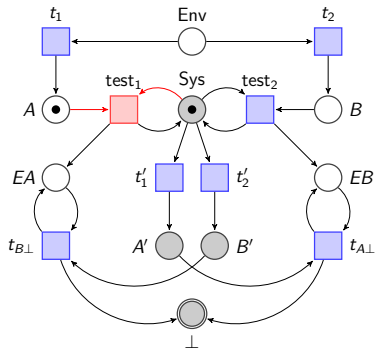
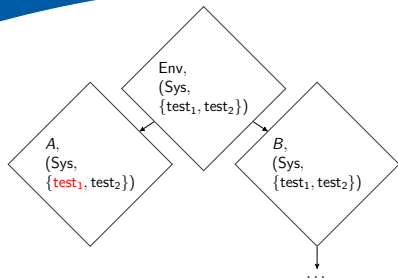
Constructing a finite-graph game [FO14]



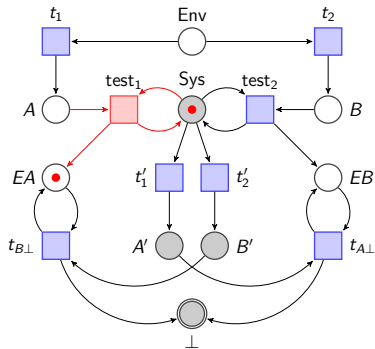
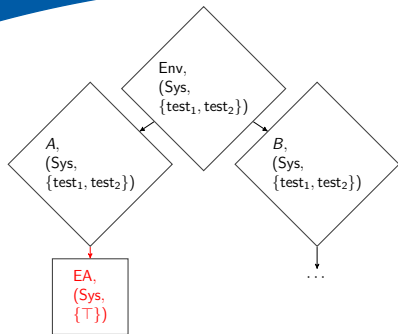
Constructing a finite-graph game [FO14]



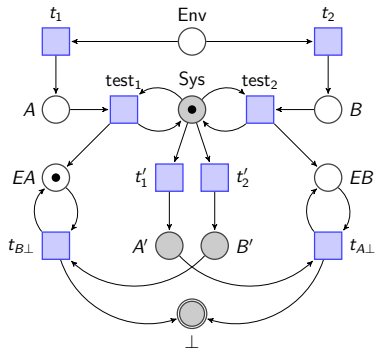
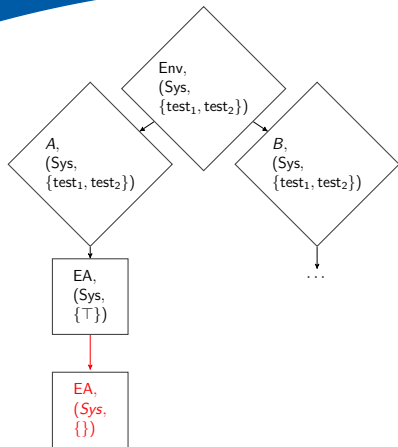
Constructing a finite-graph game [FO14]



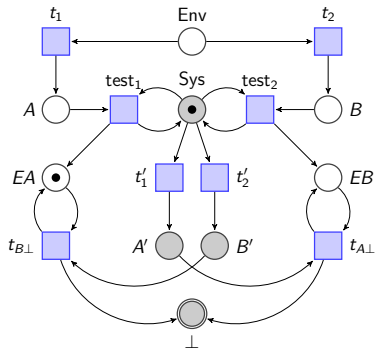
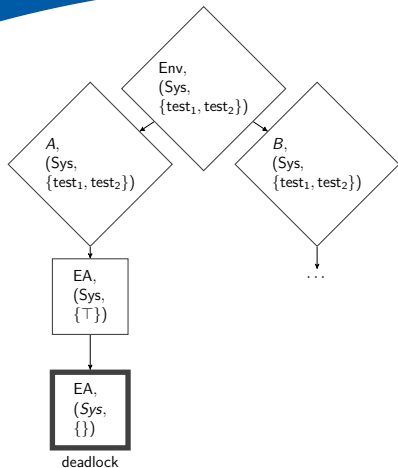
Constructing a finite-graph game [FO14]



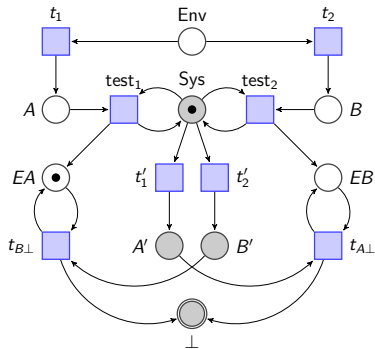
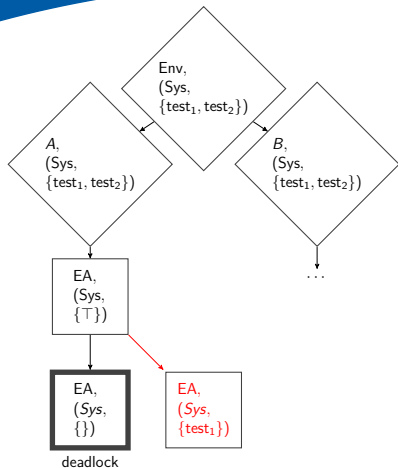
Constructing a finite-graph game [FO14]



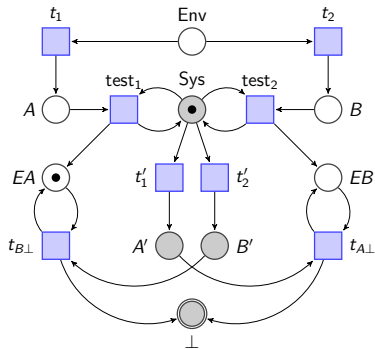
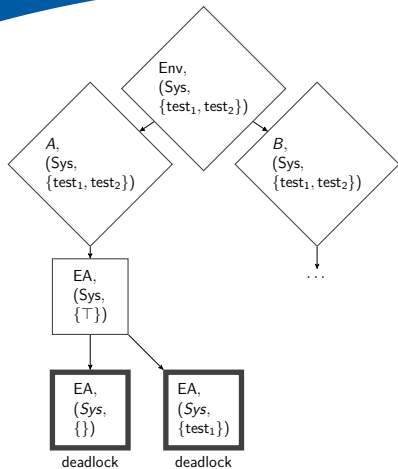
Constructing a finite-graph game [FO14]



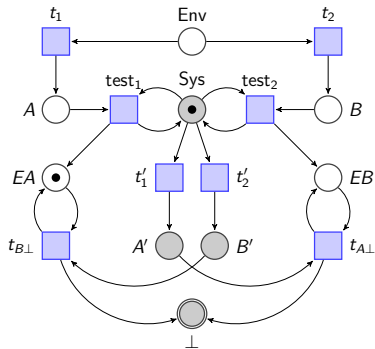
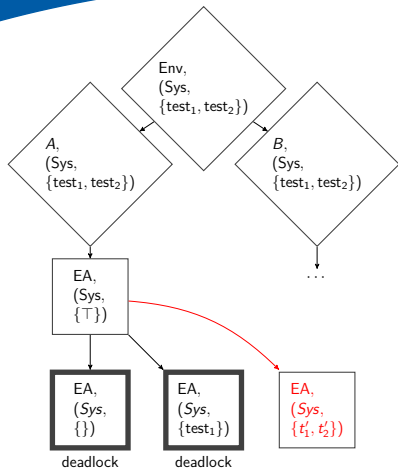
Constructing a finite-graph game [FO14]



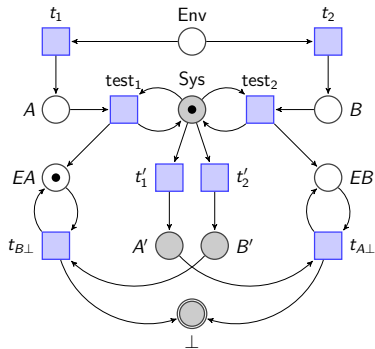
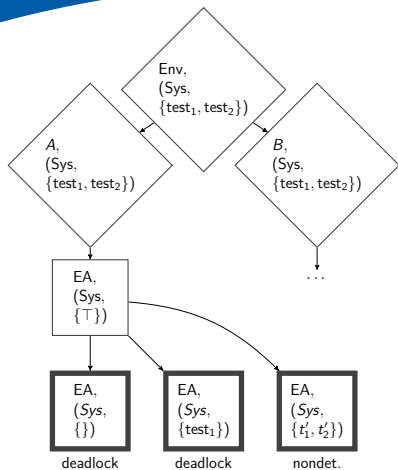
Constructing a finite-graph game [FO14]



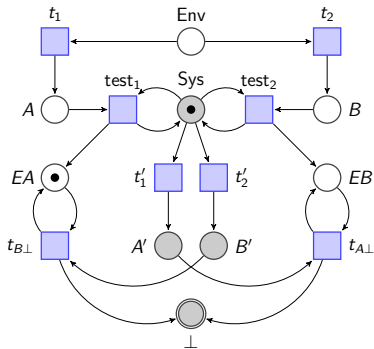
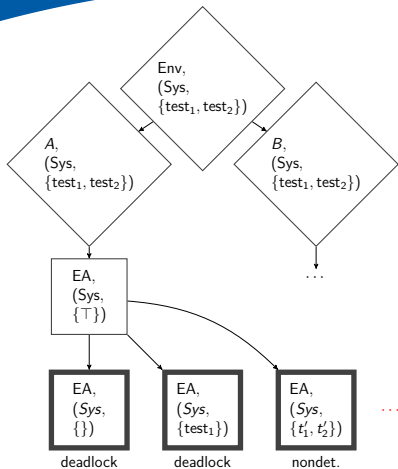
Constructing a finite-graph game [FO14]



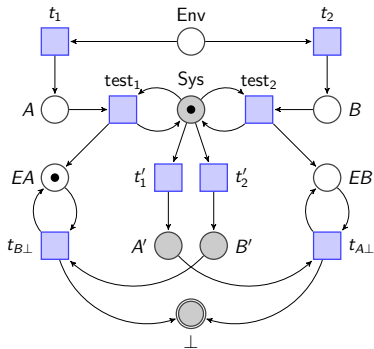
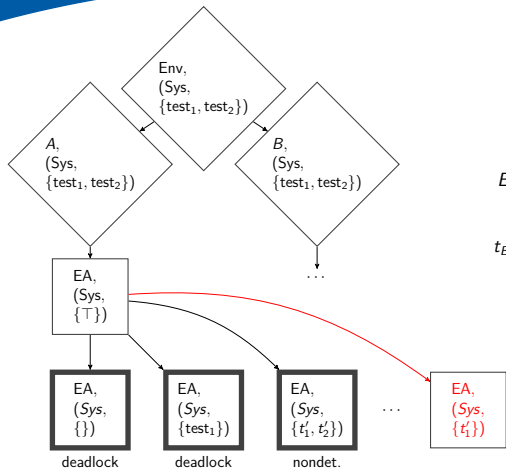
Constructing a finite-graph game [FO14]



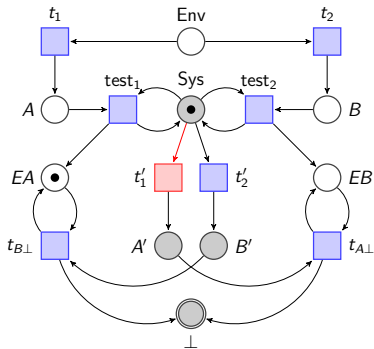
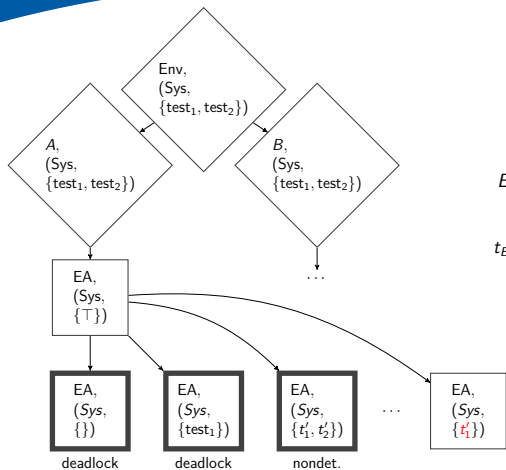
Constructing a finite-graph game [FO14]



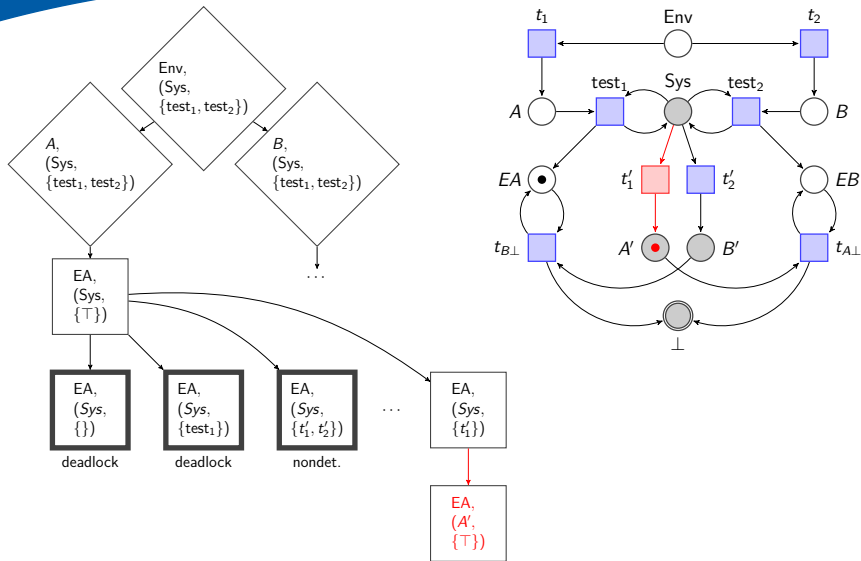
Constructing a finite-graph game [FO14]



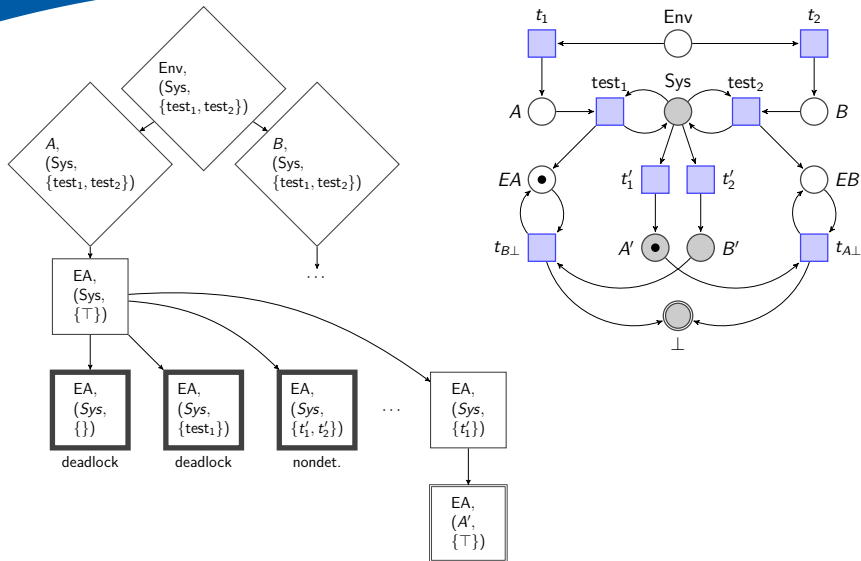
Constructing a finite-graph game [FO14]



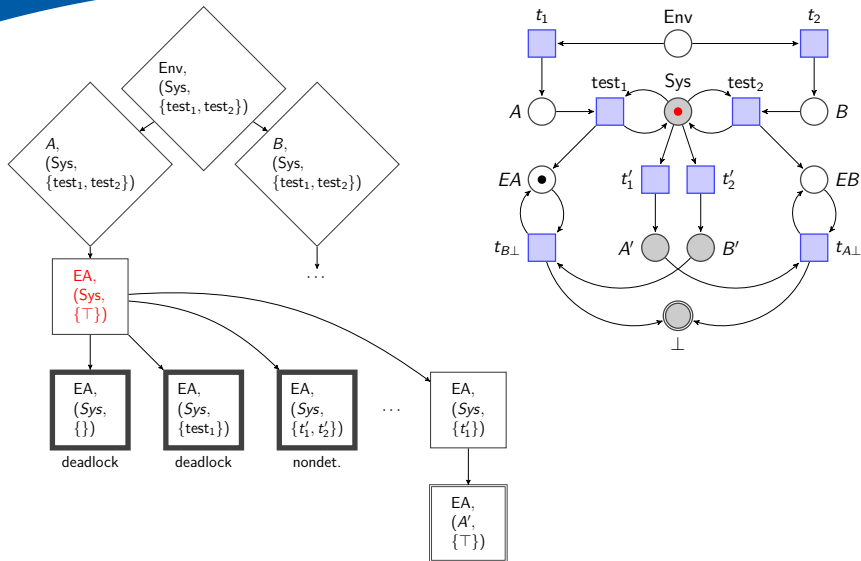
Constructing a finite-graph game [FO14]



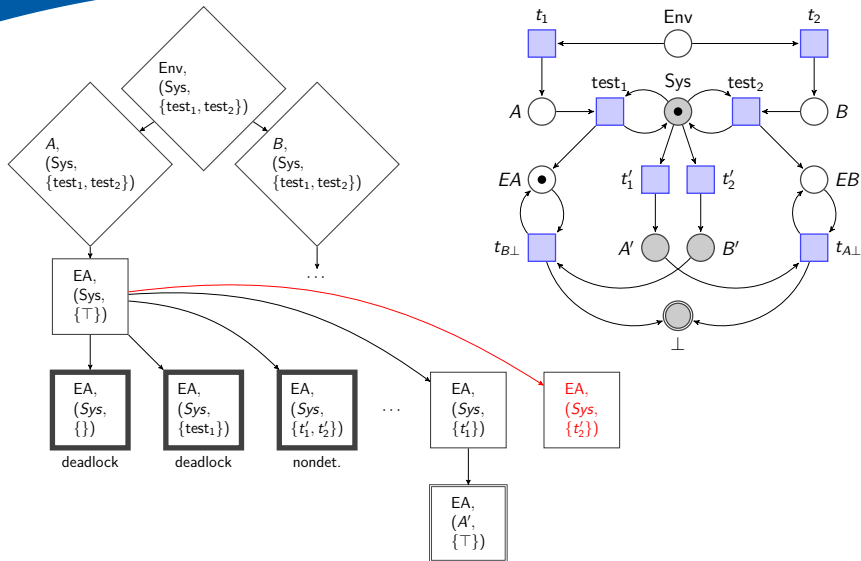
Constructing a finite-graph game [FO14]



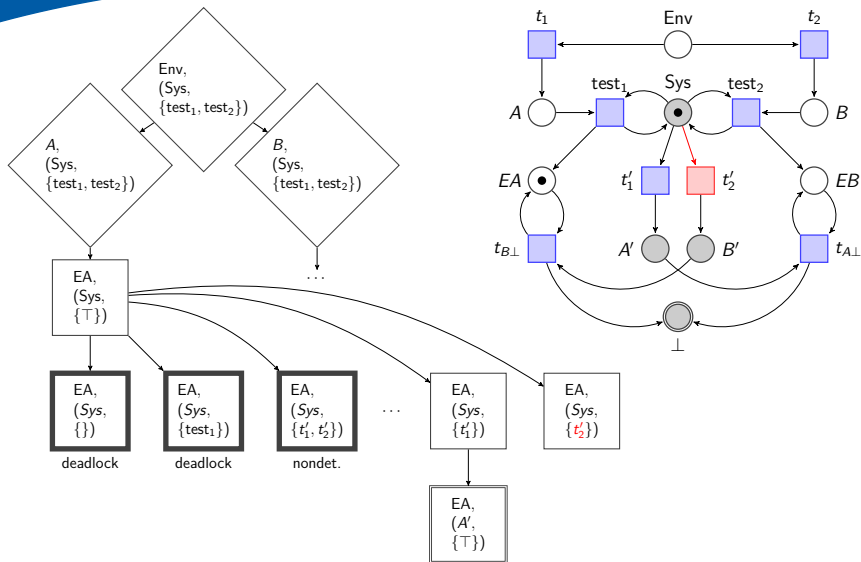
Constructing a finite-graph game [FO14]



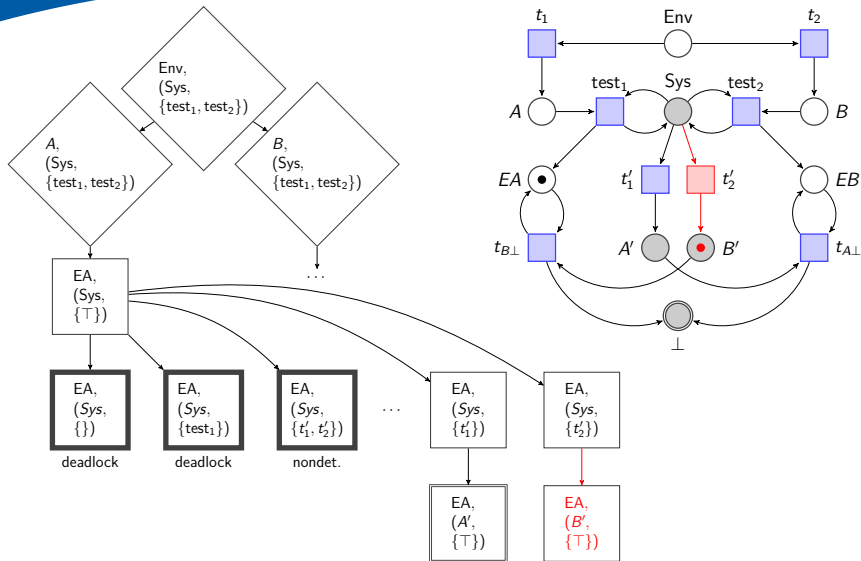
Constructing a finite-graph game [FO14]



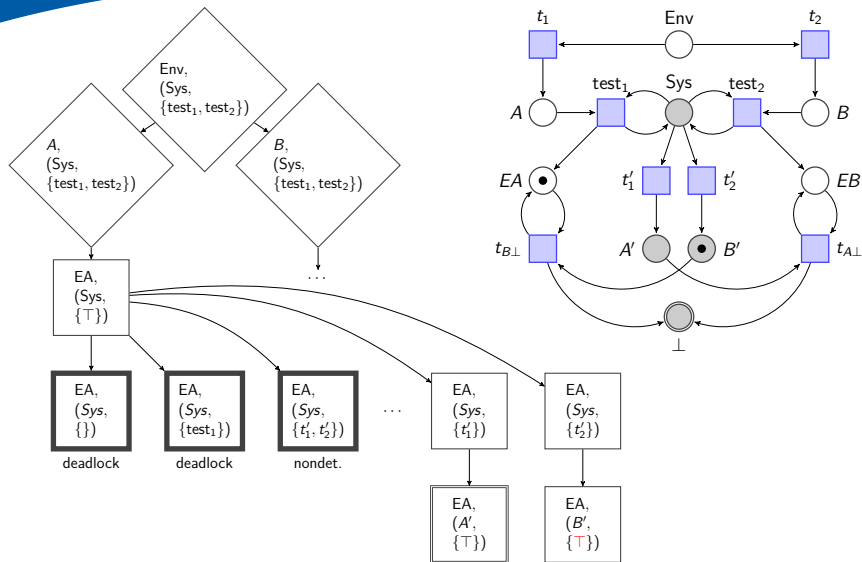
Constructing a finite-graph game [FO14]



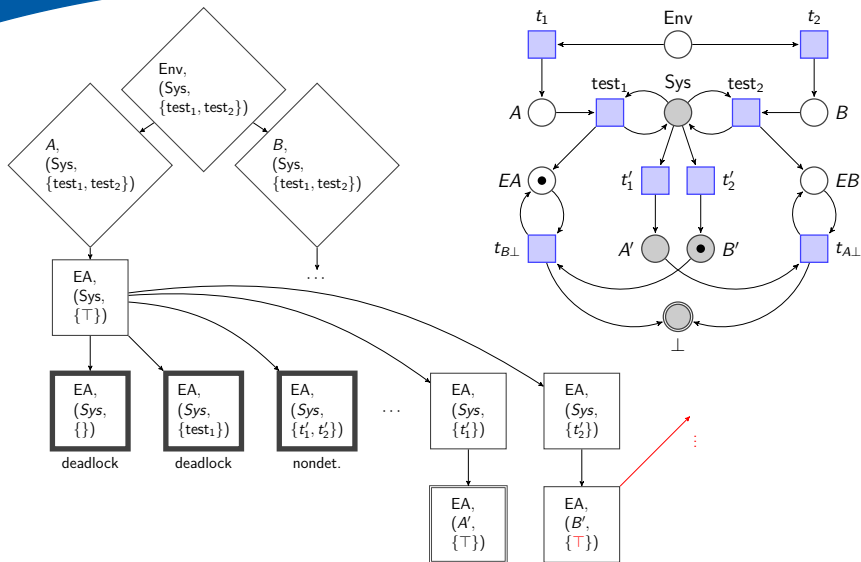
Constructing a finite-graph game [FO14]



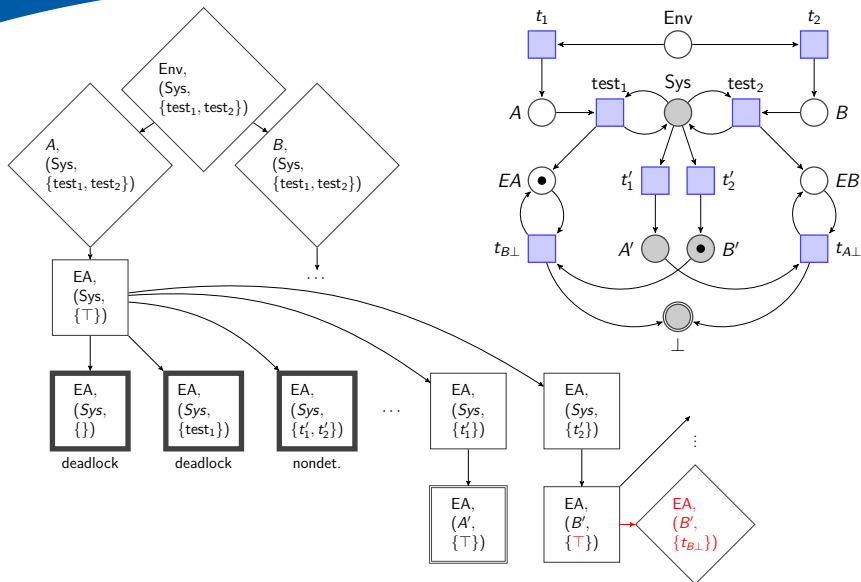
Constructing a finite-graph game [FO14]



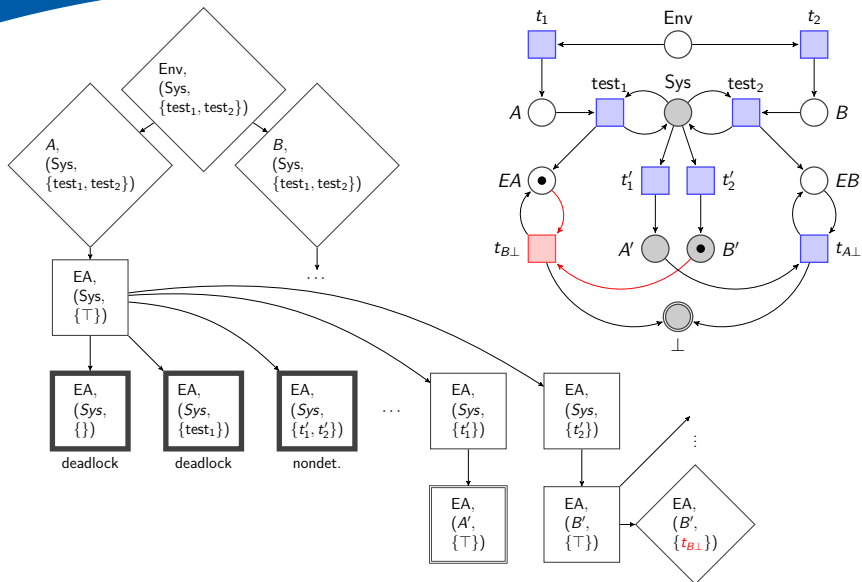
Constructing a finite-graph game [FO14]



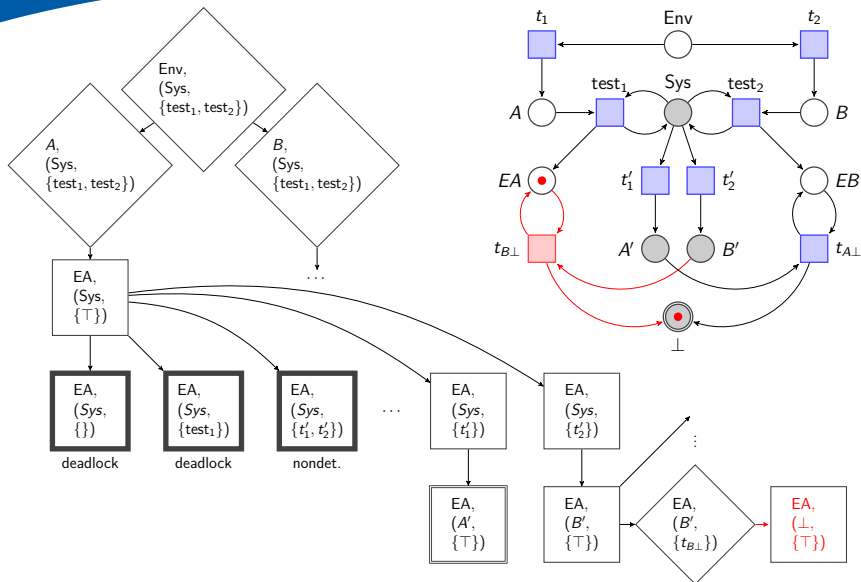
Constructing a finite-graph game [FO14]



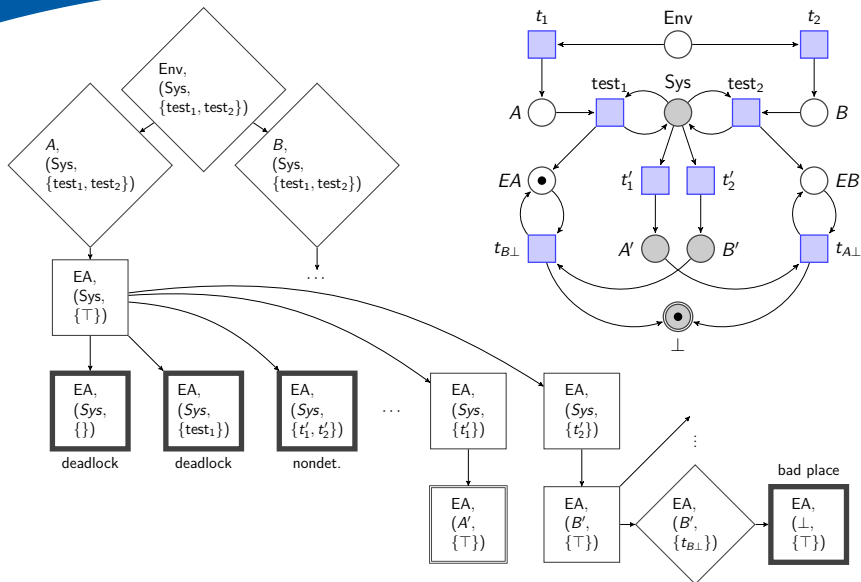
Constructing a finite-graph game [FO14]



Constructing a finite-graph game [FO14]



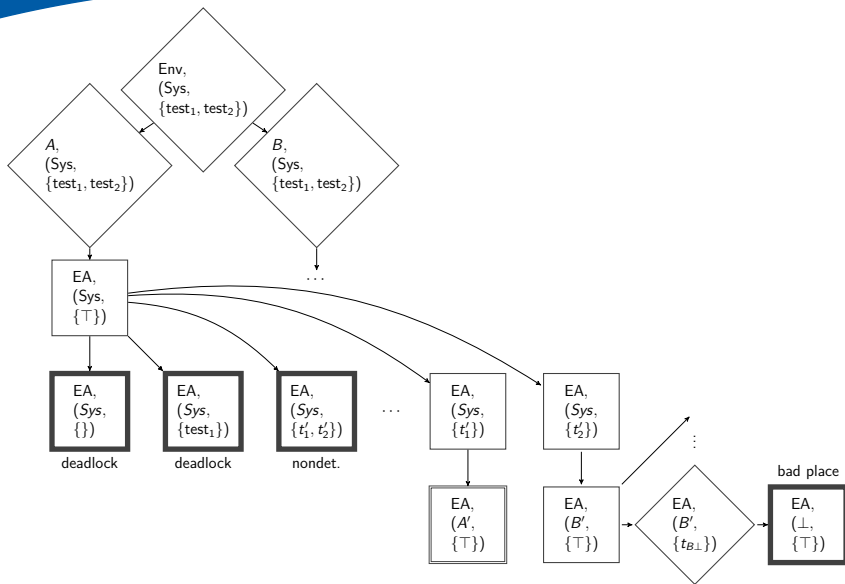
Constructing a finite-graph game [FO14]



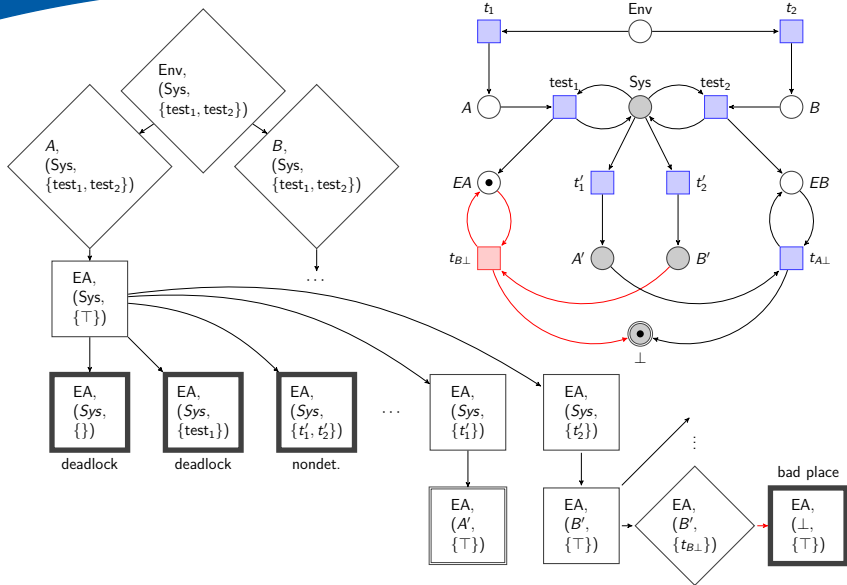
Environment wins:

- Q_0 : bad, nondeterministic and deadlocked (system) decision sets
- Q_{i+1} : add predecessors of environment transitions leading to Q_i and predecessors with all successors of system transitions leading to Q_i .

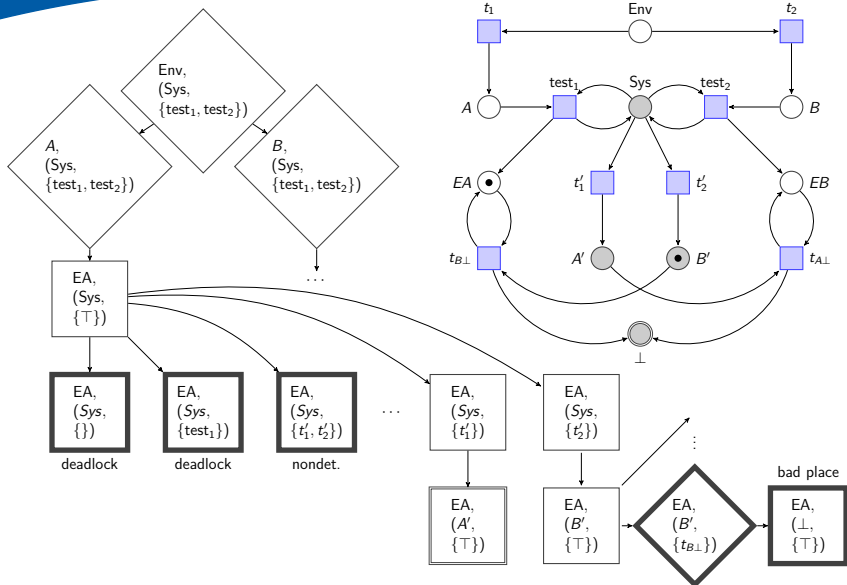
Least fixed point algorithm – Example



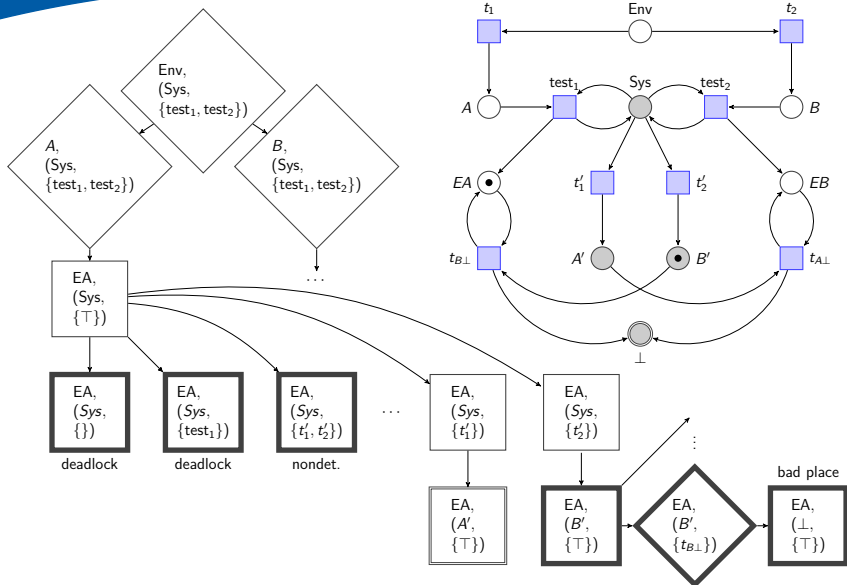
Least fixed point algorithm – Example



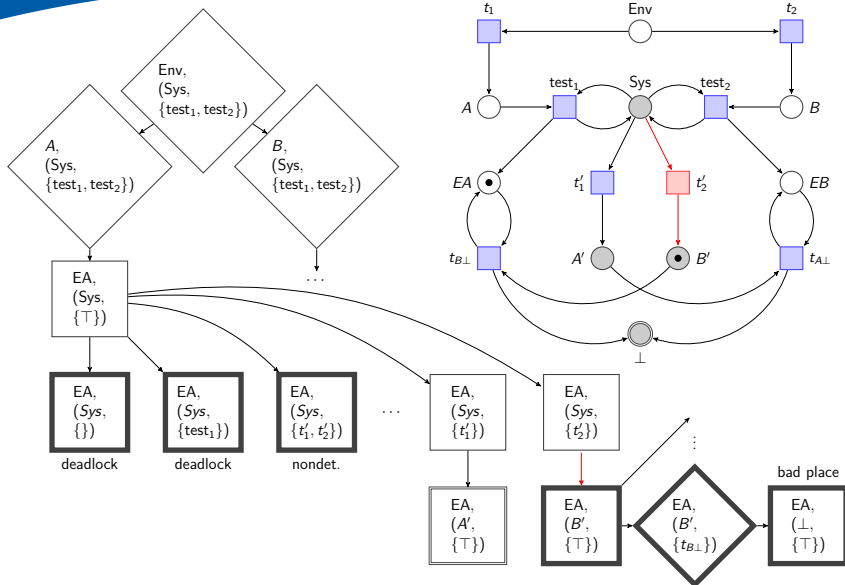
Least fixed point algorithm – Example



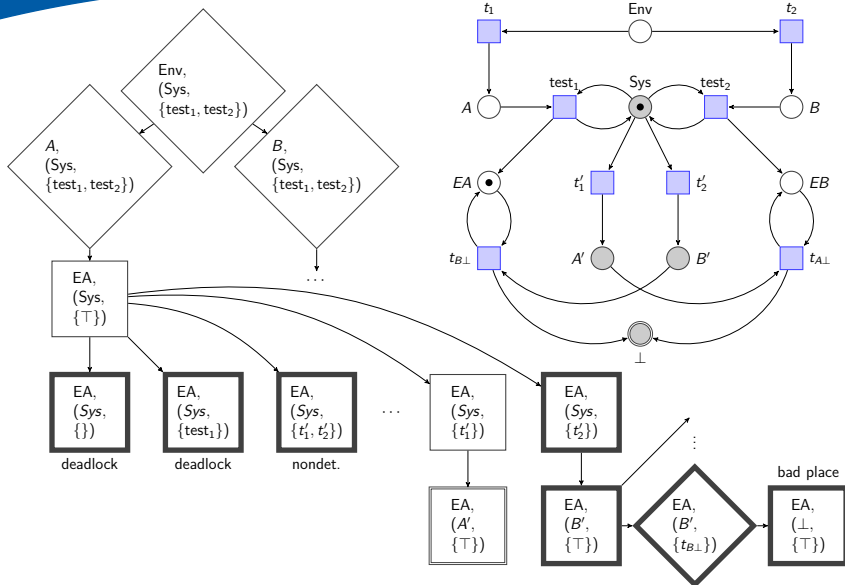
Least fixed point algorithm – Example



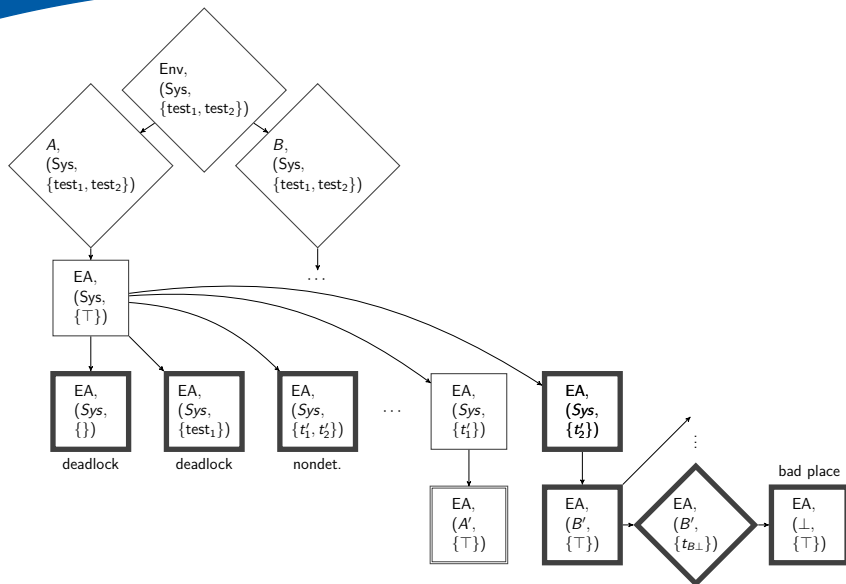
Least fixed point algorithm – Example



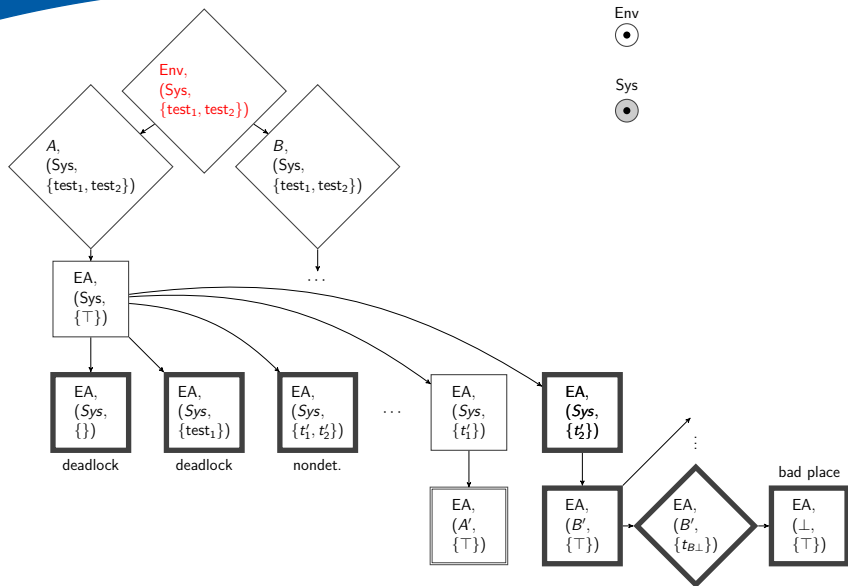
Least fixed point algorithm – Example



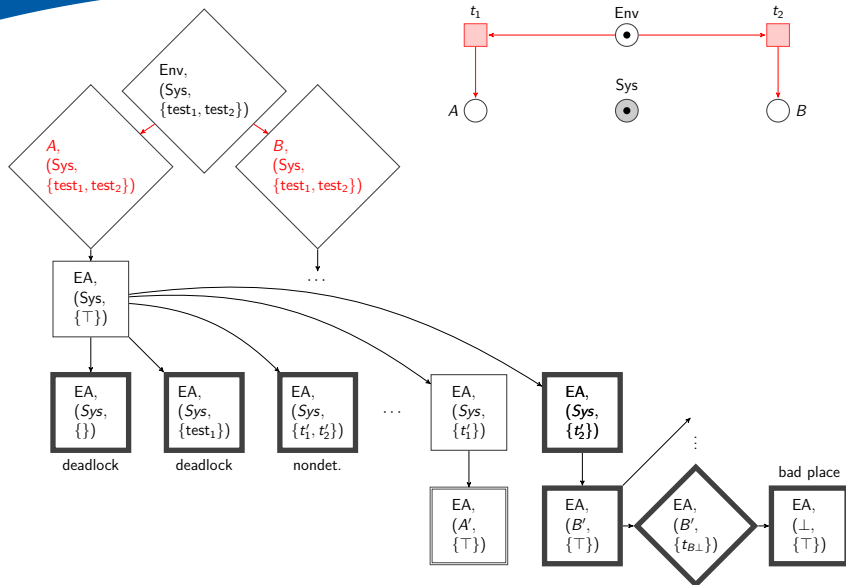
Revealing a strategy



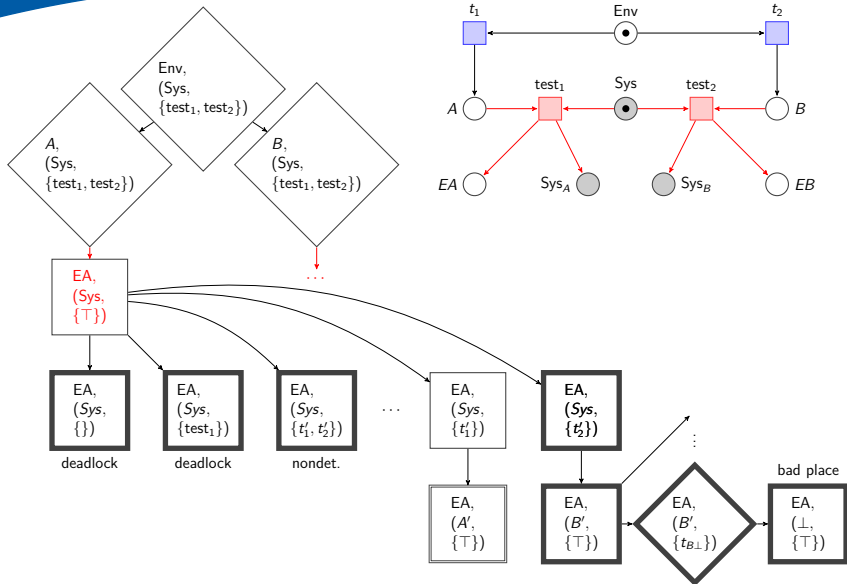
Revealing a strategy



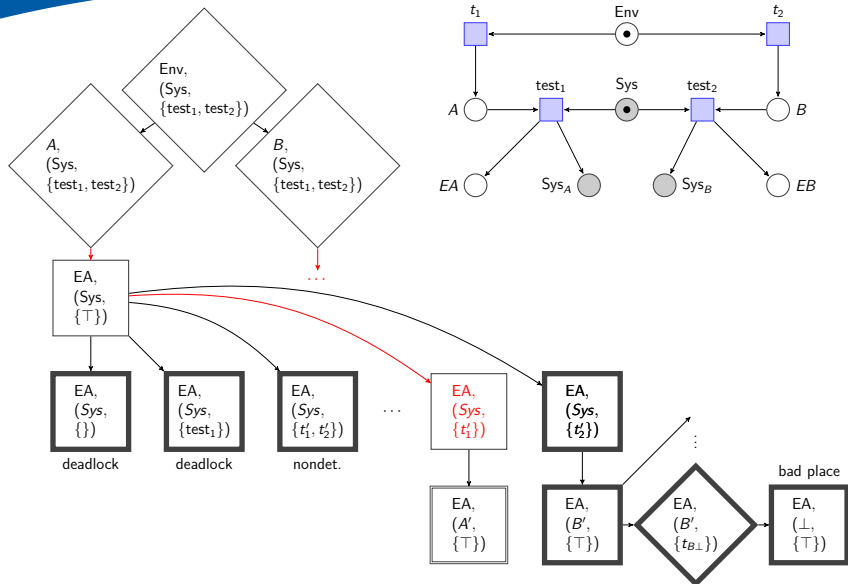
Revealing a strategy



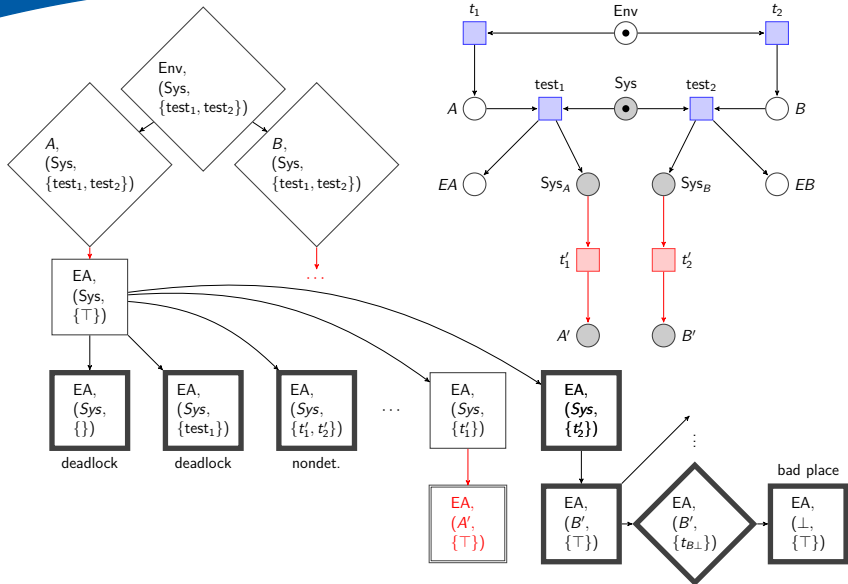
Revealing a strategy



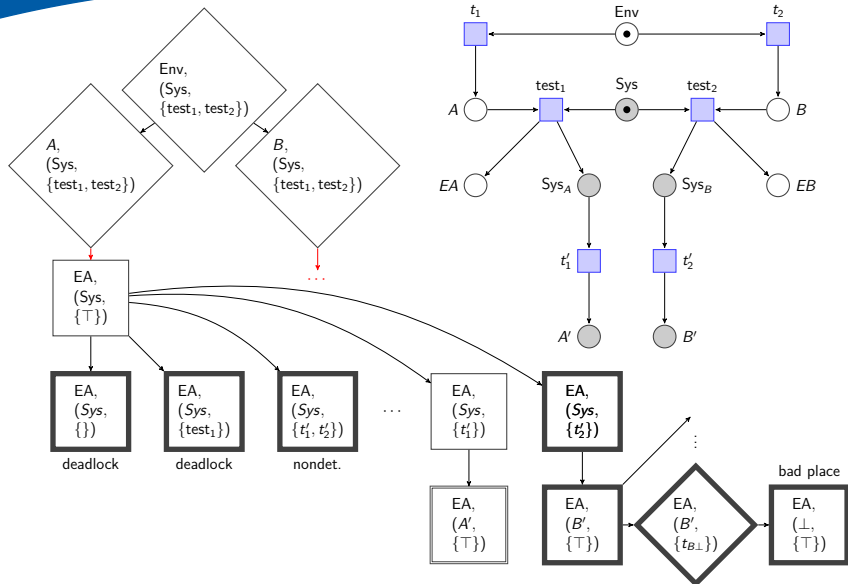
Revealing a strategy



Revealing a strategy

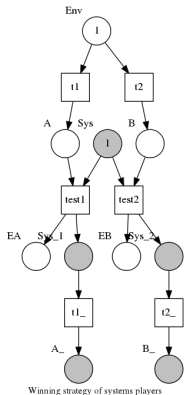


Revealing a strategy



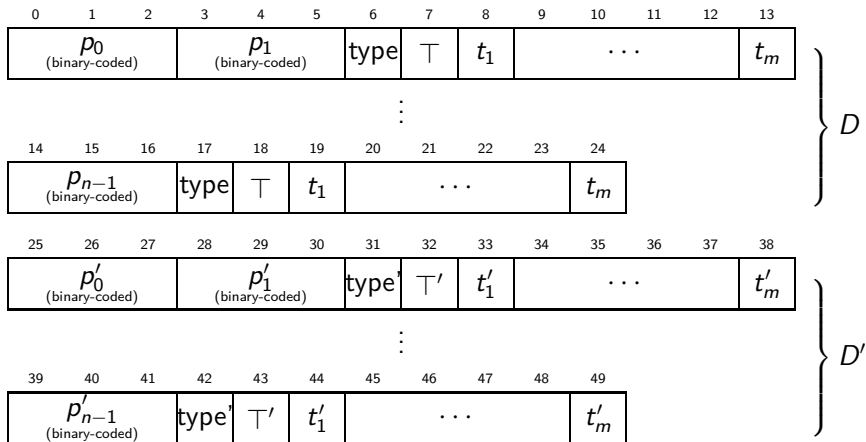
Implementation

- Prototype
- Work in progress
- APT [Old12]
 - ▶ File format
 - ▶ Import/Export
 - ▶ Data structure for Petri net
- JavaBDD [Wha]
 - ▶ Java implementation based on BuDDy [Lin] or
 - ▶ JNI interfaces for BuDDy, CUDD[Som], and CAL[Ber].
- DOT [ABL] export for Graphviz



Coding decision sets

Coding of $D \rightarrow D'$, n token, m transitions and at most 8 places.
 $((\log |\mathcal{P}| + 2 + |\mathcal{T}|) \cdot (n - 1) + \log |\mathcal{P}|) \cdot 2$ variables.



Formulas for the BDDs

0	1	2	3	4	5	6	7	8	9	10	11
p_i (binary-coded)		type	\top	t_1	\dots			$t_{\#(t')_i}$	\dots		t_m

A transition t is enabled:

$$D\{t\}_1 ::= \bigwedge_{p \in \bullet t} \left(\bigvee_{i=1, \dots, n-1} (p = p_i \wedge \text{type}_i = 1) \vee p = p_0 \right) \wedge \text{WELLFORMED}$$

A transition t' is chosen:

$$D\langle t' \rangle ::= \bigwedge_{p \in \bullet t'} \left(\bigwedge_{i=1, \dots, n-1} p = p_i \Rightarrow t_{\#(t')_i} = 1 \right) \wedge \text{WELLFORMED}$$

A transition t is fireable:

$$D[t]_1 ::= D\{t\}_1 \wedge D\langle t \rangle$$

- AT&T ; BELL-LABS:
DOT file format for GraphViz – Graph Visualization Software.
<http://www.graphviz.org/>,
- BERKELEY, University of:
CAL – California Binary Decision Diagram package; University of California, Berkeley.
http://embedded.eecs.berkeley.edu/Research/cal_bdd/,
- FINKBEINER, Bernd ; OLDEROG, Ernst-Rüdiger:
Petri Games: Synthesis of Distributed Systems with Causal Memory.

In: *Proceedings Fifth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2014, Verona, Italy, September 10-12, 2014.*, 2014, 217–230

- LIND-NIELSEN, Jørn:
BuDDy – Binary Decision Diagram package; IT-University of Copenhagen.
<http://sourceforge.net/projects/buddy/>,
- OLDENBURG, University of:
APT - Analyse von Petri-Netzen und Transitionssystemen; University of Oldenburg.
<https://github.com/renke/apt>, 2012

- SOMENZI, Fabio:
CUDD – CU Decision Diagram Package; University of Colorado, Boulder.
<http://vlsi.colorado.edu/~fabio/CUDD/cuddIntro.html>,
- WHALEY, John:
javaBDD – Java Binary Decision Diagram package;
<http://javabdd.sourceforge.net/index.html>,