To Store or Not To Store

Radek Pelánek

Masaryk University, Brno

Gerd Behrmann, Kim G. Larsen

Aalborg University

Reachability Problem

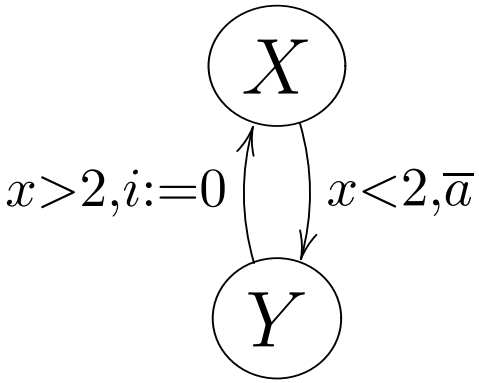
Model: networks of timed automata with variables and handshake communication

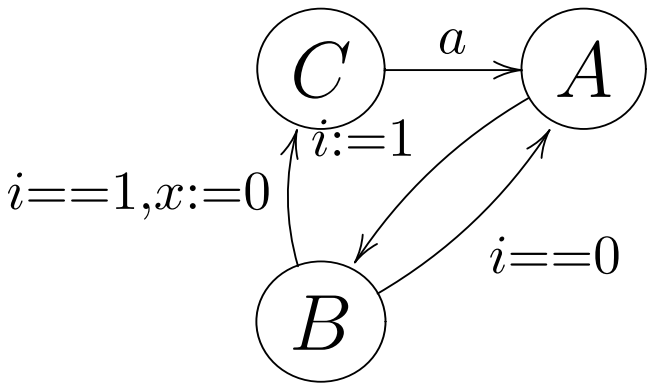
Input: Network N, state formula φ

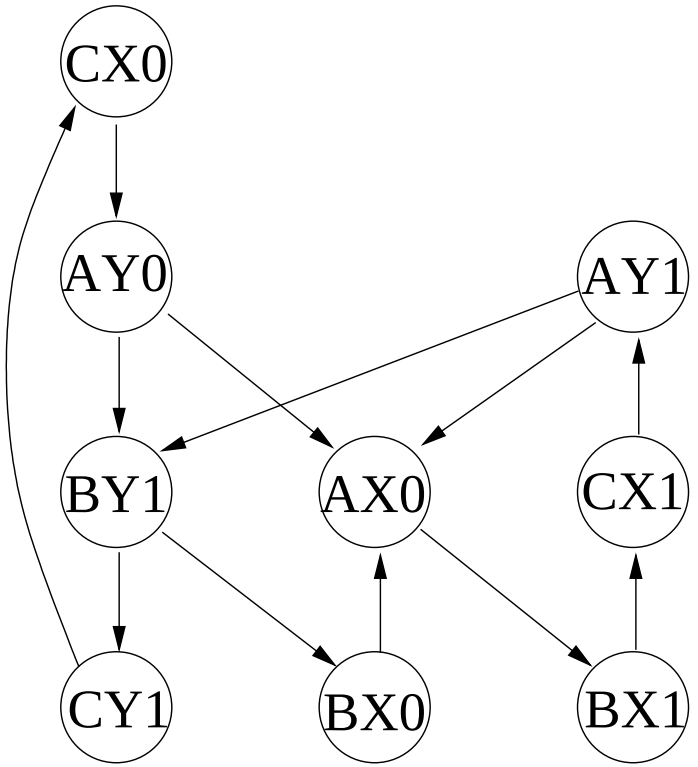
Question: Is there a state s of the network N such that s is reachable from the initial state of N and $s \models \varphi$?

Reachability $\equiv$ verification of safety properties

Example


X
$x>2, i:=0$
$x<2, \overline{a}$
Y

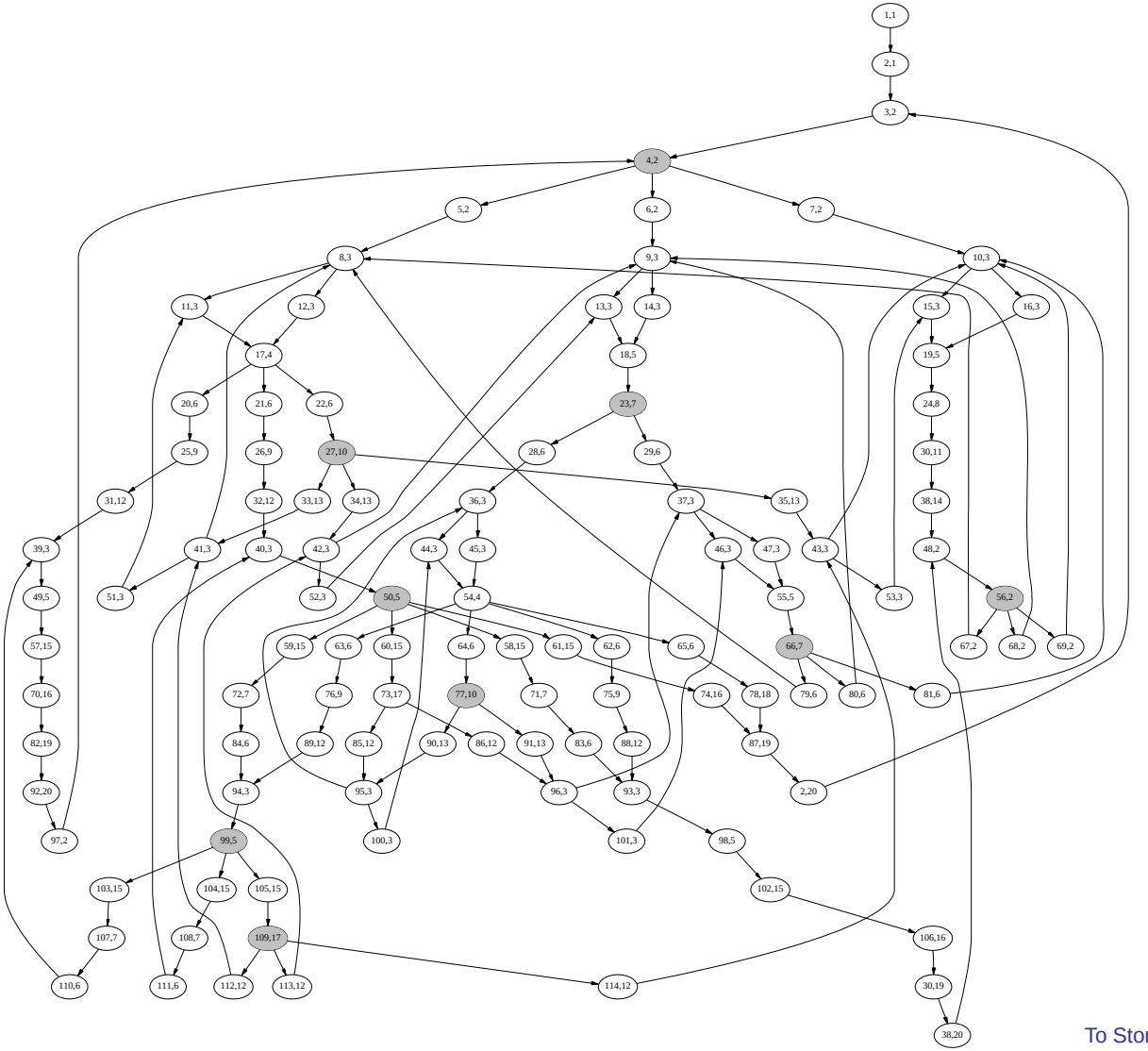

C
a
A
$i:=1$
$i==1, x:=0$
$i==0$
B


CX0
AY0
AY1
BY1
AX0
CX1
CY1
BX0
BX1

Reachability Algorithm

$W = \{s_0\};\ P = \emptyset;$
while $W \neq \emptyset$ **do**
    get s from W
    if $s \models \varphi$ **then** return true **fi**
    $P = P \cup \{s\}$
    foreach $s', t : s \stackrel{t}{\Rightarrow} s'$ **do**
        if $s' \notin P \cup W$ **then** $W = W \cup \{s'\}$ **fi** **od**
od
return false

Example


1,1
2,1
3,2
4,2
5,2
6,2
7,2
8,3
9,3
10,3
11,3
12,3
13,3
14,3
15,3
16,3
17,4
18,5
19,5
20,6
21,6
22,6
23,7
24,8
25,9
26,9
27,10
28,6
29,6
30,11
31,12
32,12
33,13
34,13
35,13
36,3
37,3
38,14
39,3
41,3
40,3
42,3
44,3
45,3
46,3
47,3
43,3
48,2
49,5
51,3
52,3
50,5
54,4
53,3
55,5
56,2
57,15
59,15
63,6
60,15
64,6
58,15
61,15
62,6
65,6
66,7
67,2
68,2
69,2
70,16
72,7
76,9
73,17
77,10
71,7
75,9
74,16
78,18
79,6
80,6
81,6
82,19
84,6
89,12
85,12
90,13
86,12
91,13
83,6
88,12
87,19
92,20
94,3
95,3
96,3
93,3
2,20
97,2
99,5
100,3
101,3
98,5
103,15
104,15
105,15
102,15
107,7
108,7
109,17
106,16
110,6
111,6
112,12
113,12
114,12
30,19
38,20

Passed List

Memory consumption is dominated by the Passed list.
Why do we store states in the Passed list?

- to guarantee termination (at least one state from each cycle has to be stored)
- to reduce the number of revisits

To achieve these goals it is sufficient to store only **some** states.

Reachability with Reduction

$W = \{(s_0, 0)\};\ P = \emptyset;$

while $W \neq \emptyset$ **do**

get (s, flag) from W

if $s \models \varphi$ **then** return true **fi**

if $\mathit{to\_store}(s, \mathit{flag})$ **then** $P = P \cup \{s\}$ **fi**

foreach $s', t : s \stackrel{t}{\Rightarrow} s'$ **do**

if $s' \notin P \cup W$ **then** $W = W \cup \{(s', \mathit{next\_flag}(s, t, \mathit{flag}))\}$ **fi** **o**

od

return false

Strategies

- **Counter** strategy
- **Random** strategy
- **Distance** strategy
- **Successors** strategy
- **Covering set** strategy
- **Combined** strategies

Distance Strategy

Observation

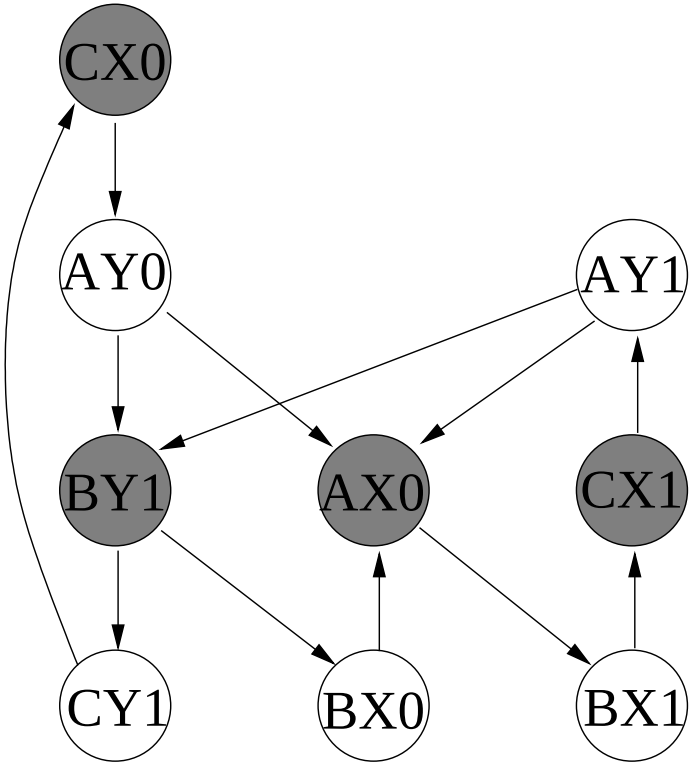
- cycles are rather long
- it is not necessary to store states close to each other

Strategy

- computes the distance from the last stored state
- stores states with the distance equal to parameter k

Distance Strategy

$k = 2$



Successors Strategy

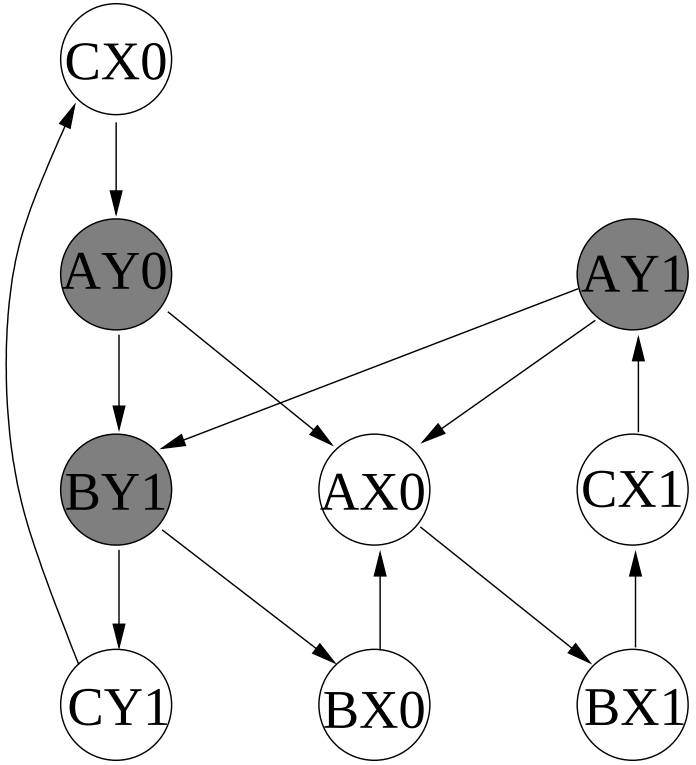
Observation

- chains of states with only one successor
- it is practically useless to store all states from such a chain

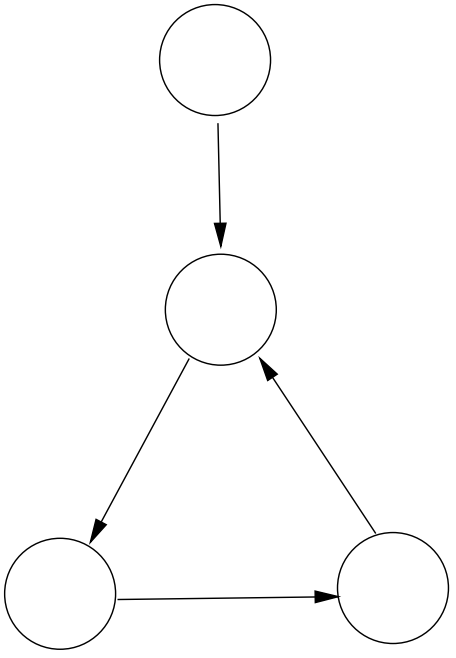
Strategy

- stores only states with more than one successor

Successors Strategy



(a)



(b)

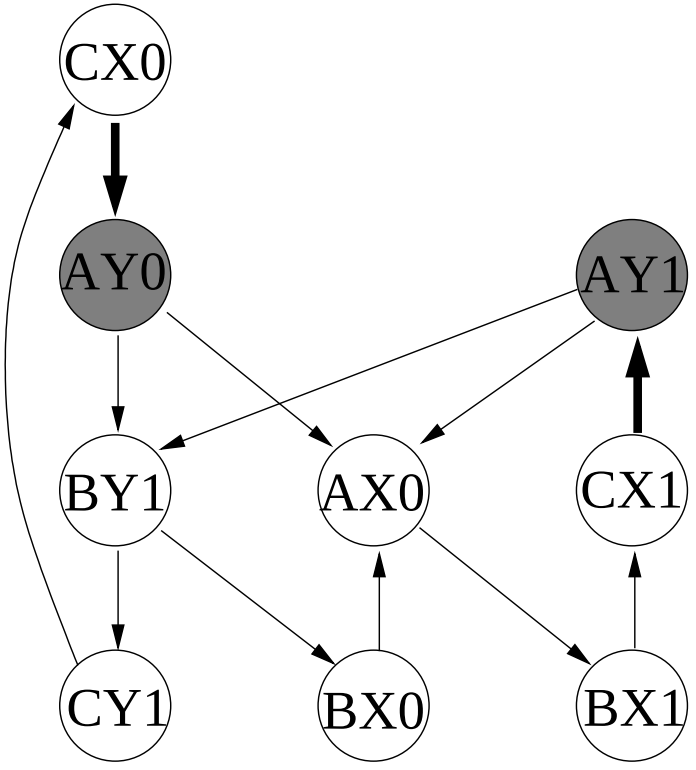
Covering Set Strategy

Let $Cover$ be a set of transitions, such that each cycle in the state space cointains at least one transition from this set.

It is sufficient to store states that are targets of such transitions.

Covering Set Strategy – Example

$$Cover = \{(C, A)\}$$



Combined Strategies

- compute the distance with respect to covering transitions
- combination of covering sets and successors strategies
- different probabilities according to the number of successors, covering transitions
- . . .

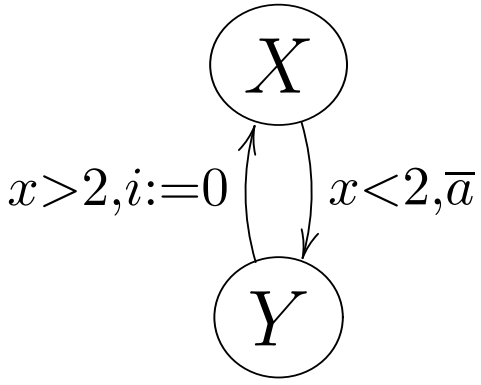
Covering Set

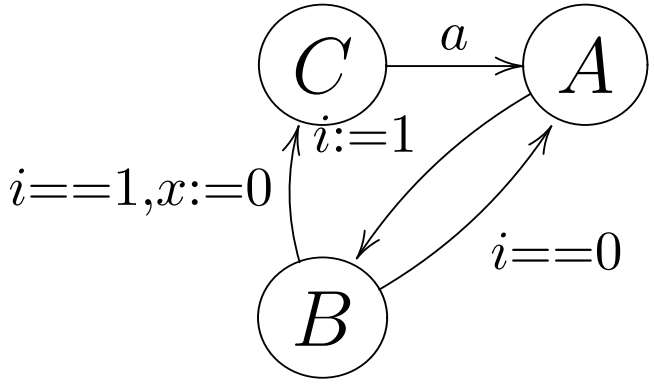
The concept of covering set was originally proposed for static partial order reduction [Kurshan et al., 1998].

- *local cycle* – cycle in an automaton (in syntax)
- *global cycle* – cycle in a state space (in semantics)
- *covering set* – set of transitions T such that each global cycle cointains at least one transition from T

We want to compute T by **static analysis** of local cycles.

Dependencies among Local Cycles





local cycles:

$$(X, Y), (C, A, B), (A, B)$$

set $\{(C, A)\}$ is a covering set

cycle (X, Y) covers cycle (A, B) because of variable i

cycle (C, A, B) covers cycle (X, Y) because of channel a

Construction of Covering Set

Minimal covering set – PSPACE-hard problem
Heuristic construction:

while not (T cover all local cycles of N) **do**

$$T = T \cup \textit{select\_transition}(N);$$

Different heuristics for selection of a next transition

Random Walk Analysis

- Is the minimal covering set the best one?
- "Good" covering set = the number of stored states during the reachability (with reduction) as low as possible
- Frequencies of transitions in the state space are more important than the size of the set
- Difficult to estimate from static analysis $\Rightarrow$ **Random Walk Analysis**

Experiments

Experiments done within UPPAAL
12 different models, including industrial case studies

- Covering set construction
- Comparison of strategies

Experiments – Covering Sets

9 different heuristics for the construction
Observations:

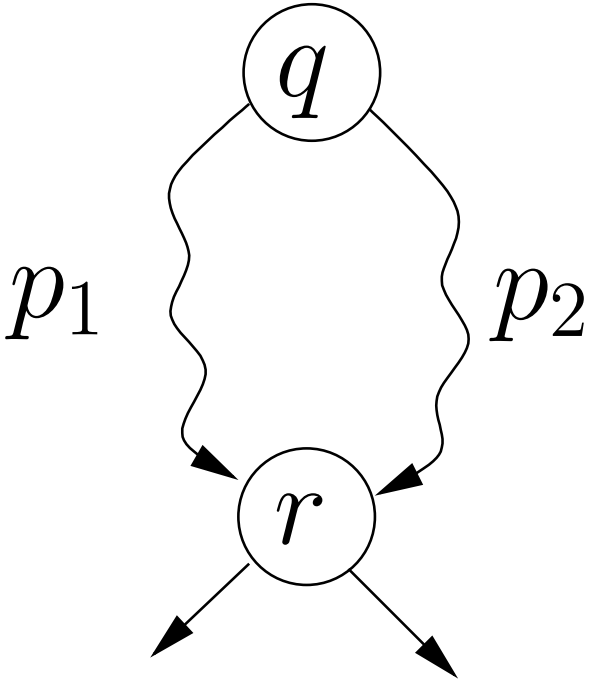
- Very small number of transitions is needed to cover all cycles
- Random walk coeficients better measure of the ’quality’ of covering set than its size
- None of the heuristics is dominant in all cases

Experiments – Storing Strategies

- It is usually possible to store less than 10% of states with reasonable increase of number of revisits
- Run-time is sometimes even faster for reachability with reduction
- None of the strategies is dominant for all examples
- Tradeoff space $\times$ time
- The best results are usually obtained for suitably choosen combination strategy

Experiments – Remarks

Order of visits is important! Breadth-first order keeps the number of revisits small.



Size of the waiting list

For some models the size of the waiting list dominates the memory consumption.

Solution: priority queue

Conclusions

- General framework for reachability analysis with reduction
- Several strategies how to decide on-the-fly whether "to store or not to store" a state
- Use of static analysis and random walk analysis for reduction during state space exploration