Model Checking WS 2012: Assignment 6

Institute for Formal Models and Verification, JKU Linz

Due 17.01.2013

Exercise 31

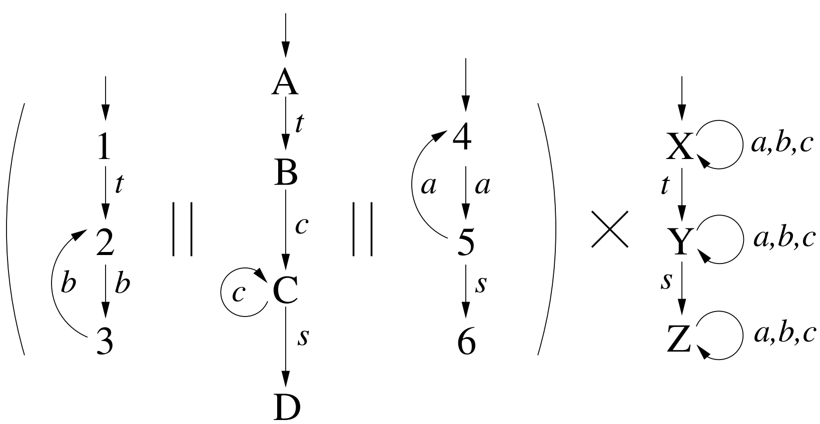
Given the following two LTS:

$A_1 = (\{1,2\},\{1\},\{a,t\},\{(1,a,1),(1,t,2),(2,t,1),(2,a,2)\})$

$A_2 = (\{A,B\},\{B\},\{b,t\},\{(A,b,A),(B,b,A),(B,t,B),(A,t,B)\})$

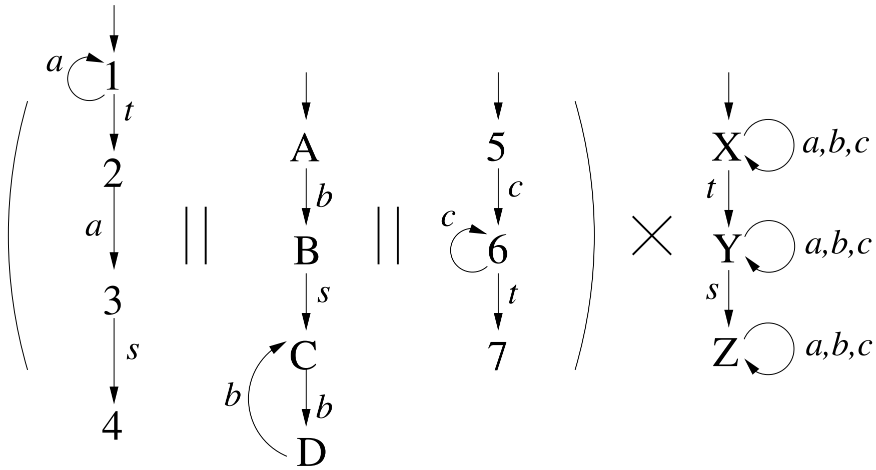
Draw the LTS for $A_1 \mid\mid\mid A_2$.

Exercise 32



For the model checking problem given above, perform reachability analysis *with* on-the-fly generation of states *and* partial order reduction and draw the resulting LTS. If there are multiple choices for local expansion, then choose the *rightmost* among all components in the asynchronous composition which are ready for local expansion.

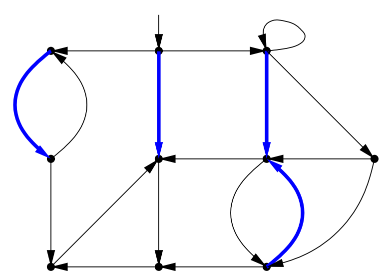
Exercise 33



For the model checking problem given above, perform reachability analysis *with* on-the-fly generation of states *and* partial order reduction and draw the resulting LTS. If there are multiple choices for local expansion, then choose the **rightmost** among all components in the asynchronous composition which are ready for local expansion.

Exercise 34

In the graph shown on the right bold blue edges correspond to fair transitions. Carry out DFS on the given graph and...



a) ...number newly visited nodes with a unique *depth-first search index (DFSI)* in the order as they are visited by DFS.

b) ...mark back edges with 'b' and cross edges with 'c'.

c) ...mark all strongly-connected components (SCCs) with circles.

d) Given the SCCs identified in part c), find infinite paths in the graph which are fair / not fair. Name examples for such paths and justify your answer.

Exercise 35

a) Reformulate the binary logical operators disjunction $\vee$, implication $\rightarrow$, equivalence $\leftrightarrow$ and XOR $\oplus$ using only binary conjunction and negation.

b) For each of the reformulations from a), draw a parse DAG with explicit sharing of nodes. How many nodes could be saved in each parse DAG when implementing negation as an edge attribute (i.e. encoded in the LSB of the pointer) instead of separate NOT-operator nodes? Justify your answer.

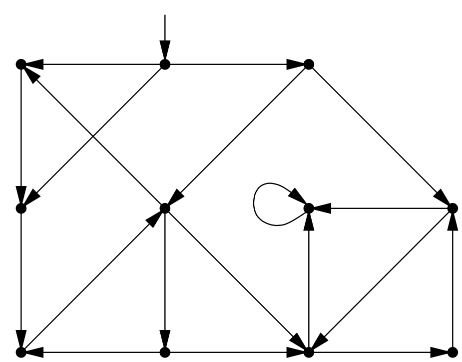
c) Given n distinct parse DAG variable nodes $v_1, v_2, \ldots, v_n$. Let D denote a parse DAG representing the n-ary conjunction $v_1 \wedge v_2 \wedge \ldots \wedge v_n$ over all v_i. Given D, let the *level* of D be the maximal length of a path from the root of D to any v_i in D. What is the *maximal* and *minimal* level of any possible D? Justify your answer.

For the exercise, assume that parse DAGs are implemented like presented in the lecture (e.g. see slide 121).

Exercise 36

The presentation of this exercise is voluntary, because the algorithm will be discussed in the lecture of Janury, 17th. If nobody volunteers, the exercise will be presented by the lecturer. This kind of exercise is part of the 2nd test and of the exam.

1.) Apply Tarjan's SCC decomposition algorithm (see slides 109 and 110) on the given graph and...



a) ...number newly visited nodes with a unique *depth-first search index (DFSI)* in the order as they are visited by DFS.

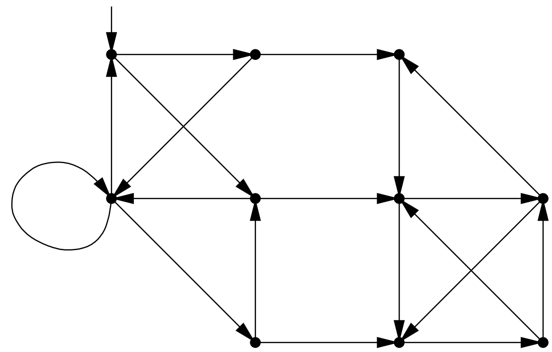
b) ...compute the *minimum reachable DFSI (MRDFSI)* for each node.[1]

c) ...mark back edges with 'b' and cross edges with 'c'.

d) ...mark all strongly-connected components with circles.

[1]Specify this value *before* it is reset to `INF` by `mrdfsi[M] = INF` near the bottom of slide 110.

2.) Apply Tarjan's SCC decomposition algorithm (see slides 109 and 110) on the given graph and...



a) number newly visited nodes with a unique *depth-first search index (DFSI)*.

b) compute the *minimum reachable DFSI (MRDFSI)* for each node and specify this value *before* it is reset to infinity by the algorithm.

c) mark back edges with 'b'

d) mark all strongly-connected components with circles.