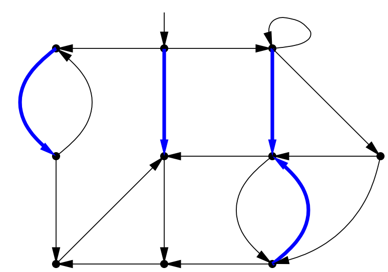
Model Checking WS 2011: Assignment 9

Institute for Formal Models and Verification, JKU Linz

Due 12.01.2012

Exercise 33

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.

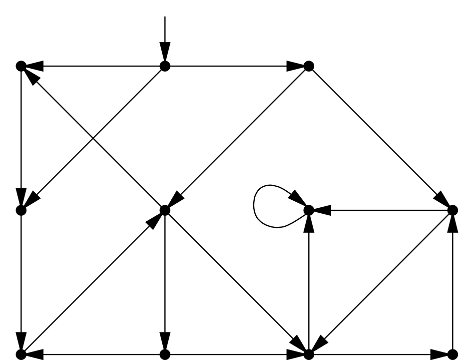
b) . . . mark back edges with 'b' and cross edges with 'c'.

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 34

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.

Comment: *For the following exercises, assume that parse DAGs are implemented like presented in the lecture (e.g. see slide 121).*

Exercise 35

a) Given boolean variables x and y, find two different formulations of the binary XOR-operation $x \oplus y$ using only negation and binary conjunction.

b) Find a DNF representation[2] for the parity function f over four boolean variables: $f(x_1, x_2, x_3, x_4) := x_1 \oplus x_2 \oplus x_3 \oplus x_4$.

c) Given three boolean variables x_1, x_2, x_3, draw all structurally different[3] parse DAGs for representing ternary conjunction over x_1, x_2, x_3 using only binary AND-operator nodes.

Exercise 36

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.

---

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

[2] Note that this exercise can be solved without constructing the truth table of f.

[3] Two parse DAGs are structurally different if the pointers to children in `child[0]` or `child[1]` are different.