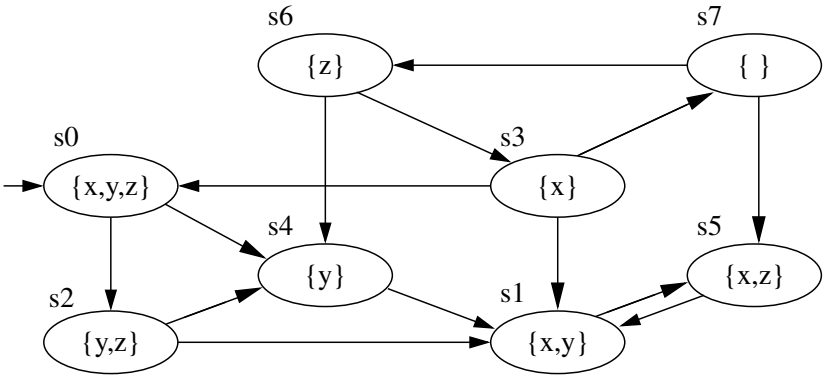
Technische Universität München (I7) **Summer Semester 2019**
J. Esparza / P. Ashok / T. Meggendorfer **08.07.2019**

Model Checking – Exercise sheet 9

The first two exercises in this sheet are based on the following Kripke structure. The goal of this session is to understand the usage of Binary Decision Diagrams (BDDs) in CTL Model Checking. The CTL formula which we shall consider is $\phi = x \ \ \mathbf{EU} \ \ (y \wedge z)$. The following exercises will walk you through all the steps involved in the algorithm.



Exercise 9.1

Compute a multi-BDD for the three sets of states satisfying y, z, and $y \wedge z$, respectively. For this, first construct the multi-BDD for the sets y and z, and then compute the BDD for $y \wedge z$ using the algorithm for intersection described in the lecture. Encode each state $s_0, \ldots, s_7$ using three bits in the obvious way:

$$s_0 \mapsto 000, s_1 \mapsto 001, \ldots, s_7 \mapsto 111$$

Draw first the full binary tree for the set, then merge all isomorphic subgraphs, and finally prune all redundant nodes, including the 0 node.

Exercise 9.2

Recall the steps involved in model checking of **EU**. $[\![p \ \ \mathbf{EU} \ \ q]\!]$ is the least fixed point of the sequence $\emptyset = X_0, \ X_1, \ X_2 \ \ldots$ where $X_{i+1} = \mu(q) \cup (\mu(p) \cap pre(X_i))$.

Let TS be the transition relation of the Kripke structure.

$$TS = \{(s_0, s_2), (s_0, s_4), \ldots, (s_7, s_6)\}$$

1. Compute the set X_1 explicitly (not as a BDD).
2. Compute a multi-BDD for X_1 and $pre(X_1)$, proceeding as follows
   (a) Compute a multi-BDD for TS and $S \times X_1$. Think of states in X_1 being encoded by $a'_0 a'_1 a'_2$; states in S being encoded by $a_0 a_1 a_2$

(b) Transform it into a multi-BDD for TS, $S \times X_1$, and $TS \cap (S \times X_1)$.

(c) Use $\exists b\ B = B[b\backslash 0] \vee B[b\backslash 1]$ to compute $\exists a'_0 a'_1 a'_2$ TS $\cap\ (S \times X_1)$

3. ★ Compute X_2, X_3 and construct a multi-BDD for X_1, X_2, X_3.

Remark: Feel free to make use of Obst (`https://www7.in.tum.de/tools/obst/`) for this exercise

Exercise 9.3

Give a formula containing 3 boolean variables which induces the largest BDD (after merging isomorphic subgraphs, removing redundant nodes and ignoring the $\perp$ or 0 node).

Solution 9.1
Use the Obst tool to build $x = \{0, 1, 2, 4\}$ and $y = \{0, 2, 5, 6\}$. Then run the intersection of the top level nodes. This would give a BDD which must be isomorphic to the BDD you construct by hand.

Solution 9.3
$(x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z) \vee (x \wedge y \wedge z)$