1 SAT-encoding of ROBDD

The SAT-encoding is based on this idea, that every path from the root of the ROBDD to the 1-sink represents a model (or a class of models, if we admit shortcuts/longedges).

For every BDD node N we introduce a Boolean variable c_N with the meaning:

$$\neg c_N \iff \text{no path from root to 1-sink can go throuh } N \tag{1}$$

We distinguish between reachability upwards (from the 1-sink) and downwards (from the root). First we consider reachability from the 1-sink and assume N is a node with decision variable x and successor nodes T (for *high-successor*) and F (for *low-successor*). Obviously, the 0-node is not reachable from the 1-sink, that means $\neg N_0$ is a unit clause, and the 1-sink is reachable. We obtain the following rules for the inner nodes.

- if both successors are not reachable, then the node itself is also not reachable:
$$\neg c_T \wedge \neg c_F \Rightarrow \neg c_N \tag{2}$$
- if the the high-successor is not reachable and the decision variable is known to be true, then N is not reachable:
$$\neg c_T \wedge x \Rightarrow \neg c_N \tag{3}$$
- analogously for the low-successor:
$$\neg c_F \wedge \neg x \Rightarrow \neg c_N \tag{4}$$

Now we consider reachability from the root. Obviously the root R itself is reachable, thus c_R is a unit clause. For all other nodes we observe the following condition. We assume $P_1, \ldots, P_k$ are the parents of N and $l_1, \ldots, l_k$ are the decisions that are necessary to go from P_i to N. More precisely, if l_i is a positive literal, then N is the high-successor of P_i (case for l_i negative is analogous). Then N is not reachable passing P_i if P_i is not reachable or l_i is known to be false:

$$\bigwedge_{i=1}^{k} (\neg c_{P_i} \vee \neg l_i) \Rightarrow \neg c_N \tag{5}$$

Of course this rule has to be expanded by a Tseitsin transformation to CNF.

Now we need one more rule to prune variables also in from the BDD's domain. Basically we can prune a variable x if it is known that no path from the root to the 1-sink can go through an edge that is labelled with x (analogously for $\neg x$). We have to pay special attention to longedges that jump the variable x in the BDD. Let $(A_1, l_1, B_1), \ldots, (A_k, l_k, B_{,k})$ be the set of longedges that jump x, in particular in node A_i the high-successor (if l_i is positive) is B_i. Further let $(C_1, x, D_1), \ldots, (C_m, x, D_m)$ be the set of all edges that are labeled with x. We have now:

$$\bigwedge_{i=1}^{k} (\neg c_{A_i} \vee \neg c_{B_i} \vee \neg l_i) \wedge \bigwedge_{i=1}^{m} (\neg c_{A_i} \vee \neg c_{B_i}) \Rightarrow \neg x \tag{6}$$