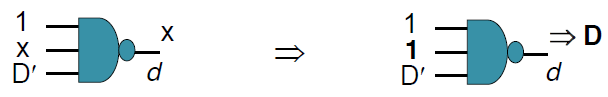
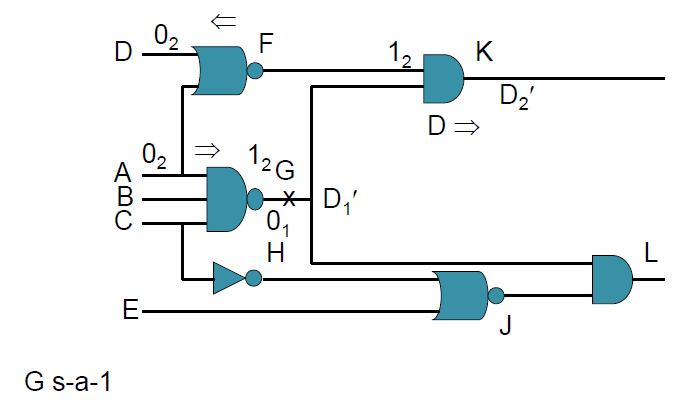
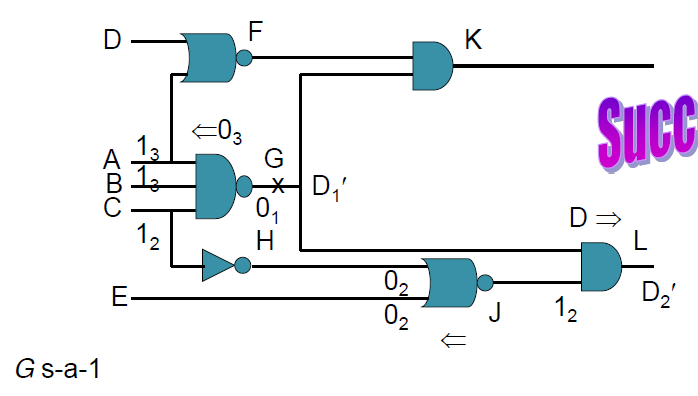
**D-Algorithm review:**

**D-Drive:** Do not forget we are backtracking in D-Algorithm! BTW this is just D-Drive and not D-Alg!





The D-drive did not work in the figure above; we have to backtrack!



This time it worked!

Imply and Check carries out all implications – forward and backward, and also checks for inconsistency.

If there is no inconsistency then the problem is solved; else it fails and we need to backtrack, i.e. take another choice.

Imply and check is a **recursive** procedure. “imply” does implications, “check” checks for consistency. “Imply” is usually implemented using a push down stack. Put lines to be processed into a stack, e.g. imply and check on a simple 2input AND gate (A, B; C), results in pushing the two new line assignments “line A=1” and “line B= 1” onto the stack. Eventually we pop “line A=1” and process it. Eventually we return to pop “line B= 1”. When the stack is empty or an inconsistency occurs, imply and check terminates.

**D-Frontier:**

The D-frontier is a set and contains all gates whose output value is X and have one or more inputs that are assigned the value a D or D’.

If D-Frontier is empty, this implies either no solution exists or backtrack is required.

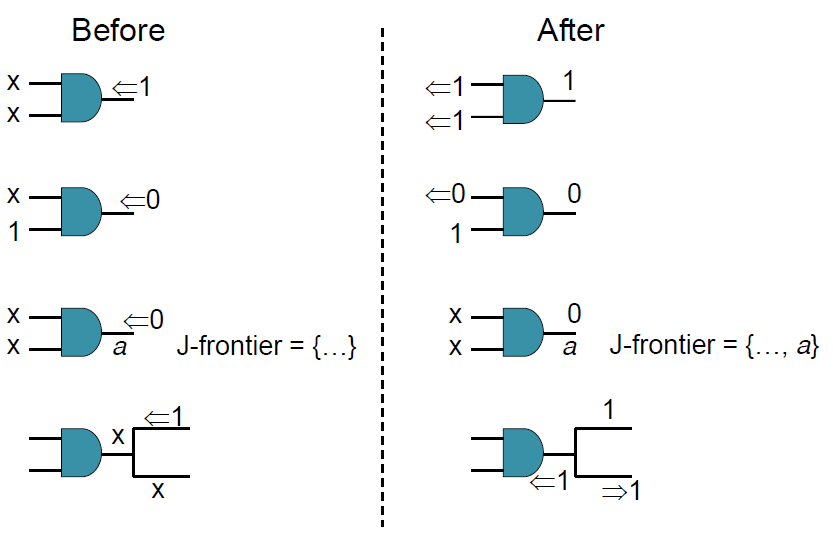
**J-Frontier:**

The J-frontier (J for justify) is the set of all gates having a 0 or 1 on their output but whose input line values do not imply the output value, and in addition, there is no unique input assignment that implies the output value. Thus a choice exists.

**Line Justification:**

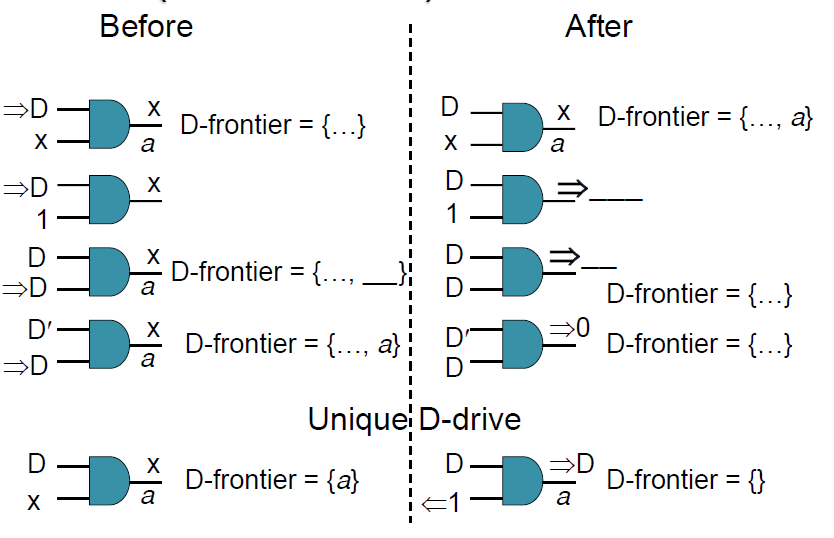
Select a line in the J-frontier and justify its value. There are always choices. If need be, backtrack and select the next choice.

**Backward implication** in imply and check:



**Forward implication** in imply and check:





Imply and check is a **recursive** procedure. “imply” does implications, “check” checks for consistency. “imply” is usually implemented using a push down stack. Put lines to be processed into a stack, e.g.

D-Algorithm:

* Uses implication, D-drive, line justification, D-frontier and J-frontier.
* Multiple path sensitization

Interesting cases for D-drive:

1. Gate “d” is one of several gates in the D-frontier. The D-drive is unique.

D-alg(){ }