Exercises Lecture 6
Intelligent Systems Programming (ISP)

Exercise 1 (adapted from A97 2.1-2)

Describe a polynomial time algorithm for determining whether a DNF is satisfiable. Describe a polynomial time algorithm for determining whether a CNF is a tautology.

Exercise 2 (adapted from A97 2.5)

Explain how the question of tautology and satisfiability can be decided if we are given an algorithm for checking equivalence between two Boolean expressions.

Exercise 3 (adapted from A97 3.1)

Show how to write the Boolean expressions $\neg x$, $x \wedge y$, $x \vee y$, $x \Rightarrow y$, and $x \Leftrightarrow y$ using the if-then-else operator, tests on un-negated variables, and the constants 0 and 1. Draw the ROBDDs for the expressions.

Exercise 4

Draw the ROBDD for $(x_1 \wedge y_1) \vee (x_2 \wedge y_2)$ using the ordering x_1, x_2, y_1, y_2.

Exercise 5 (adapted from A97 3.2)

Draw the ROBDD for $(x_1 \Leftrightarrow y_1) \wedge (x_2 \Leftrightarrow y_2) \wedge (x_3 \Leftrightarrow y_3)$ using the ordering x_1, y_1, x_2, y_2, x_3, y_3 and x_1, x_2, x_3, y_1, y_2, y_3

Exercise 6

Ignoring cycles, variable orderings and reduction rules, give an upper bound on the number of ROBDDs on n variables with g internal nodes. As for circuits during lecture, use this estimate to show that the fraction of Boolean functions with ROBDDs that have polynomial size in n goes to 0 as n goes to infinity.

Mandatory assignment

A threshold function $f^k(x_1, x_2, \dots, x_n)$ is a Boolean function on *n* Boolean variables that is true if at least *k* of the Boolean variables are true.

Thus,

$$f^k(x_1, x_2, \dots, x_n) \equiv (|\{x_i = 1\}| \geq k).$$

1) Draw the ROBDD of $f^3(x_1, x_2, x_3, x_4, x_5)$ using the variable order $x_1 \prec x_2 \prec x_3 \prec x_4 \prec x_5$.
2) Label each internal node of your ROBDD with the number of true variables on the path leading to the node.
3) Argue that the size (number of nodes) of the ROBDD of $f^k(x_1, x_2, \dots, x_n)$ is O(*kn*).
   Hint: Use your solution from 2) to get an idea.