CSC 236 October 21, 2004
St George Campus University of Toronto

Assignment #3
Due: November 4, 2004, by 6:00 PM
(in the CSC 236 drop box in BA 2220)

Question 1. (10 marks)

As noted in section 5.6 of the notes, some boolean connectives are associative. An operator "$\Diamond$" is associative iff for all propositional formulas P, Q, and R, $(P \Diamond Q) \Diamond R$ is logically equivalent to $P \Diamond (Q \Diamond R)$.

"$\wedge$" and "$\vee$" are associative. Which other common operators are associative? Consider $\rightarrow$, $\leftrightarrow$, $\oplus$, and $|$ (***nand***). State whether or not the operator is associative and prove your statement.

Question 2. (20 marks)

Classify each of the following statements as either a tautology (true under all truth assignments), a contradiction (false under all truth assignments), or a contingency (satisfiable yet not a tautology). Prove your assertions using the logical equivalences in section 5.6 (in particular, do not use truth tables to prove a tautology or a contradiction).

a. $((x \vee y) \rightarrow z) \ \vee \ (z \rightarrow (x \vee y))$

b. $(x \rightarrow y) \ \vee \ (x \rightarrow \neg y)$

c. $((x \rightarrow y) \wedge (y \rightarrow z)) \ \leftrightarrow \ (x \rightarrow z)$

d. $((x \rightarrow y) \wedge (x \rightarrow z)) \ \rightarrow \ (x \rightarrow (y \wedge z))$

e. $((x \rightarrow y) \vee (x \rightarrow z)) \ \rightarrow \ (x \rightarrow (y \vee z))$

f. $((x \wedge y) \rightarrow (x \wedge z)) \ \rightarrow \ (x \rightarrow (y \rightarrow z))$

Question 3. (10 marks)

a. Use structural induction to prove that $\{\rightarrow\}$ is not a complete set of boolean connectives (operators), as discussed in section 5.11 of the notes.

b. Prove that $\{\rightarrow\}$ *is* a complete set of boolean connectives, in the following unusual sense: It is possible to express any boolean function with a formula consisting only of $\rightarrow$, the propositional variables, and the constants *true* and *false*.

Suggested approach: Present an algorithm (mechanical transformation) which converts an arbitrary boolean function to an equivalent formula of the required type. Your algorithm will probably begin "Convert the function to DNF" (or CNF), as we already know how to do this.

Question 4. (10 marks) Consider truth assignments involving only the propositional variables x_0,x_1,x_2,x_3 and y_0,y_1,y_2,y_3,y_4. Every such truth assignment gives a value of 1 (representing true) or 0 (representing false) to each variable. We can therefore think of a truth assignment τ as determining a four-bit integer x_τ depending on the values given to x_0, x_1, x_2, and x_3, and a five-bit integer y_τ depending on the values given to y_0, y_1, y_2, y_3, and y_4. Specifically, we can define the integers $x_\tau = \tau(x_0) + 2 \cdot \tau(x_1) + 4 \cdot \tau(x_2) + 8 \cdot \tau(x_3)$ and $y_\tau = \tau(y_0) + 2 \cdot \tau(y_1) + 4 \cdot \tau(y_2) + 8 \cdot \tau(y_3) + 16 \cdot \tau(y_4)$.

Write a formula that is satisfied by exactly those truth assignments τ for which $y_\tau = x_\tau + 1$. Your formula may use any of the Boolean connectives discussed in the notes. Explain how you obtained your formula, and justify its correctness.

Note: This can be done by writing down a truth table for nine propositional variables — i.e., a truth table with $2^9 = 512$ rows. This is too much tedious work, and simplifying the resulting formula is an immense task. The problem can be solved more easily (and interestingly) by expressing as a propositional formula the condition under which the five-bit number $y_4y_3y_2y_1y_0$ is exactly one more than the four-bit number $x_3x_2x_1x_0$. (Hint: Ask yourself what this formula would have to say about $y_0, \ldots, y_4$ first in case x_0 is 0, then in case $x_0 = 1$ but $x_1 = 0$, then in case $x_0 = x_1 = 1$ but $x_2 = 0$, and so on.)