20875 Software Engineering

Tutorial 1

1 Boolean logic

1. [trivial] Given the truth table of NAND given in class, prove that `x nand y = not (x and y)`
2. [trivial] Given the truth table of NOR given in class, prove that `x nor y = not (x or y)`
3. [trivial] Prove that `x and y = y and x`
4. [trivial] Prove that `x and (y and z) = (x and y) and z`
5. [trivial] Prove that `x or y = y or x`
6. [easy] Prove that `x or (y or z) = (x or y) or z`
7. [easy] Prove that `x and (y or z) = (x and y) or (x and z)`
8. [easy] Prove that `x or (y and z) = (x or y) and (x or z)`
9. [easy] Prove that `(not x) and (not y) = not (x or y)`
10. [easy] Prove that `(not x) or (not y) = not (x and y)`
11. Invent a binary Boolean operator that is distinct from AND, OR, XOR, NAND and NOR.
12. Find an equivalent Boolean expression for the above operator that only uses the operators NOT, AND and OR.
13. [long] List all the possible binary Boolean operators, including trivial ones. For each, give its truth table and and equivalent Boolean expression that only uses NOT, AND and OR.
14. Give an expression equivalent to `(not x)` using only the NAND operator.
15. Give an expression equivalent to `(x and y)` using only the NAND operator.
16. Give an expression equivalent to `(x or y)` using only the NAND operator.
17. Give an expression equivalent to `(not x)` using only the NOR operator.
18. Give an expression equivalent to `(x and y)` using only the NOR operator.
19. Give an expression equivalent to `(x or y)` using only the NOR operator.
20. Give the truth table of the following Boolean expression:
    `(((x and (not y)) or z) or ((not (x and y)) and z)) and ((not (y and z)) or (y or z))`
21. Put the above expression into CNF
22. Put the above expression into DNF.
23. Put the following expression into DNF:
    `(x1 or y1) and (x2 or y2) and (x3 or y3) and (x4 or y4)`
24. Put the following expression into DNF: `(x1 or y1) and (x2 or y2) and ...  and (xN or yN)`

25. [hard] Show how any Boolean expression with n variables and k literals using operators NOT, AND, OR can be put into a CNF expression with $n+k$ variables, $3k$ clauses and $7k$ literals or less.

26. [very hard] [long] Write a code that reads a Boolean expression in input and writes an equivalent CNF expression as above on its output.

2 Integer arithmetic

1. Explain why the largest unsigned integer with n bits is $2^n - 1$.

2. Write the following numbers in binary: 7, 8, 9, 15, 16, 17, 184, 127, 128, 129, 255, 256, 257

3. Write the decimal representation of the following unsigned binary numbers:

   (a) `1000 0000`

   (b) `0001 0000`

   (c) `0000 1000`

   (d) `1100 0000`

   (e) `0101 1100`

   (f) `1001 0010`

4. Write the decimal representation of the following 8-bit signed binary numbers:

   (a) `1111 1111`

   (b) `1000 0000`

   (c) `1100 0000`

   (d) `1001 0010`

5. The hexadecimal notation represents numbers in base 16, using the digits {`0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f`}. Write the following numbers in hexadecimal: 7, 8, 9, 15, 16, 17, 184, 127, 128, 129, 255

6. Write the hexadecimal representation of the following unsigned binary numbers:

   (a) `1000 0000`

   (b) `0001 0000`

   (c) `0000 1000`

   (d) `1100 0000`

   (e) `0101 1100`

   (f) `1001 0010`

7. Implement a binary-to-decimal converter in C

8. Implement a decimal-to-binary converter in C17 (`-std=c17` option on a C compiler)

9. Given an integer $k \geq 0$. Show that the representation using $-k$ in two's complement signed integers is the same as the unsigned representation of (`NOT` k) $+ 1$, where the `NOT` operator flips all bits.

10. [long] Implement the addition and subtraction operations for 256-bit unsigned integers in C.

11. [long] [hard] Implement the multiplication operation for 256-bit unsigned integers in C.

12. [long] [very hard] Implement the division operation for 256-bit unsigned integers in C.