CSC322 Assignment 2

Ru Ou

June 2016

1 Do Exercise 7.1.1.45 from the text. Of course, Knuth already provides an answer. Your job is to expand upon that answer, explaining each step carefully (perhaps with an example or two). [Replaces problem that was here before.]

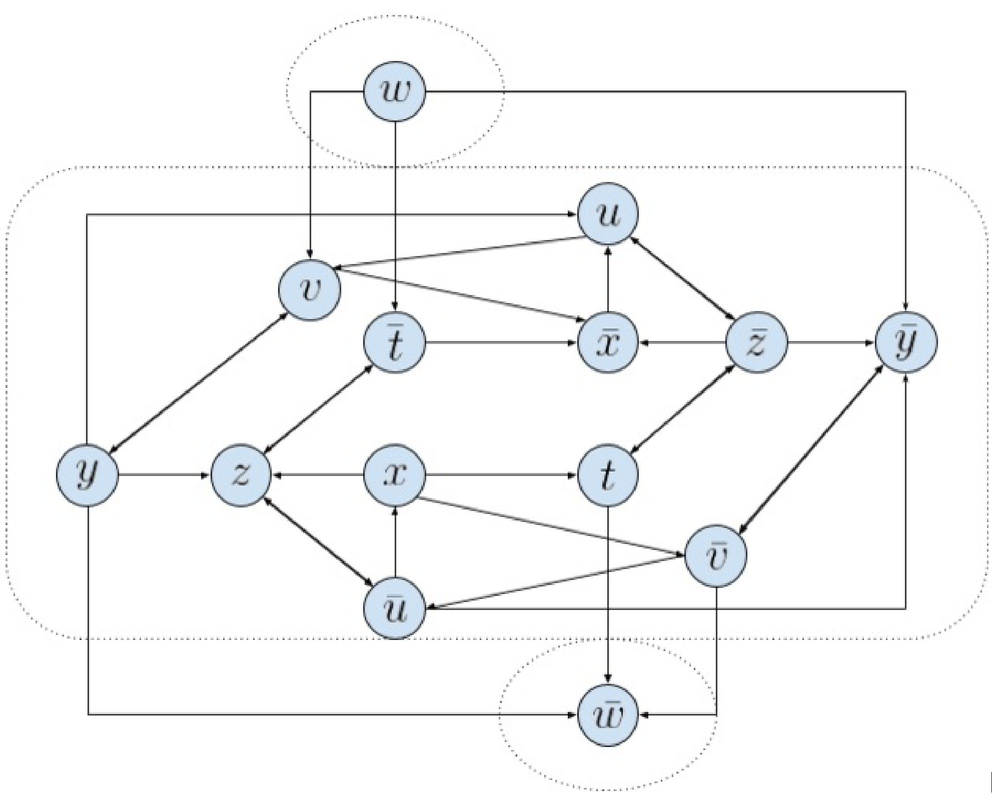
a) Show that exactly half of the Horn functions of n variables are definite.

If $f(x_1, ..., x_n)$ is indefinite horn function if and only if $f(1, ..., 1) = 0$
If $g(x_1, ..., x_n)$ is definite horn function if and only if $g(1, ..., 1) = 1$
If $f(x_1, ..., x_n) = 0$,then $g(x_1, ..., x_n) = f(x_1, ..., x_n) \vee (x_1 \wedge ... \wedge x_n) = f(1, ..., 1) \vee (1 \wedge ... \wedge 1) = 0 \vee 1 = 1$, $g(x_1, ..., x_n)$ is definite horn function and is not equal to $f(x_1, ..., x_n)$.
If $g(x_1, ..., x_n) = 1$, then $x_1 = 1, ..., x_n = 1$ and $f(x_1, ..., x_n) = f(1, ..., 1) = 0$ is indefinite horn function and is not equal to $g(x_1, ..., x_n)$.
So $f \Leftrightarrow g$ is one-to-one.

b)Also show that there are more Horn functions of n variables than monotone functions of n variables (unless n=0).

For indefinite horn function $f(x_1, ..., x_n)$, $f(1, ..., 1) = 0$,but $f(1, ..., 0, ..., 1) = 1$. So indefinite horn function is not monotone. Hence, there are more Horn functions of n variables than monotone functions of n variables.

2 If Vegas was included in the list of comedians, what are the strongly connected components of the digraph from the discussion of Krom clauses?



3 In the previous assignment the programming assignment specified a mapping, call it f, that took a length 2^n binary string x and turned it into another length 2^n binary string $f(x)$. Formulate and prove a hypothesis about what happens as f is successively applied to itself. I.e., what is $f(f(x))$, $f(f(f(x)))$, etc.? You may find it useful to work with a program in exploring this question; my program can be found in resources if you didn't write your own.

$f(f(x)) = x$ To compute the multilinear representation, we divided the original binary function into two part and then two part and then...until each part have only one element. Then we combine each two part to a new part, which is a "mirrored tree method".
We can compute $f(x)$ by a "mirrored tree method" from up to down. Also, we can compute $f(f(x))$ by the "mirrored tree method" of $f(x)$ from down to up. Then we get x itself.