3 important computational problems

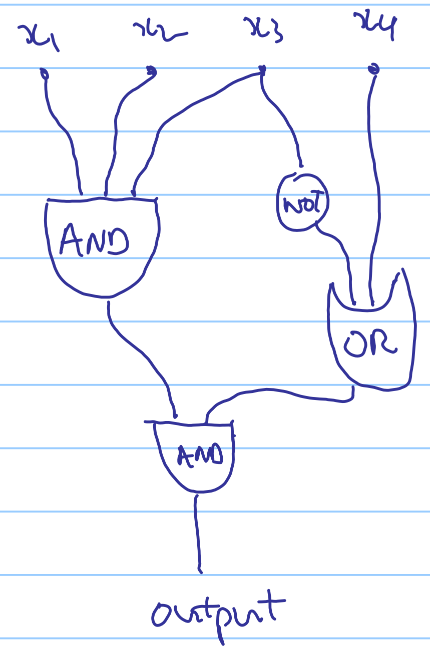
Note Title 4/7/2014

① CircuitSAT (circuit satisfiability)

Main idea: Given a Boolean circuit, can we set the inputs so that the output is a 1? If so, how?

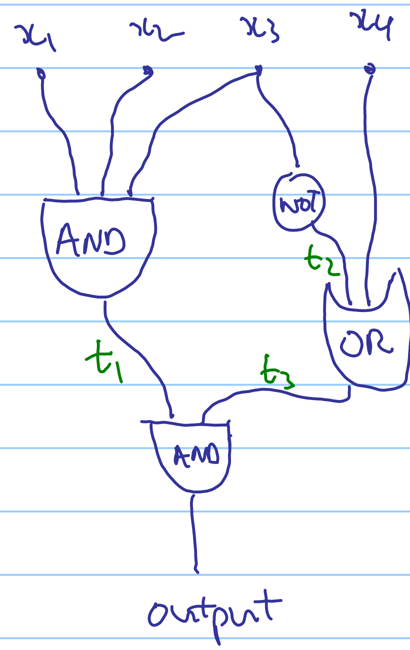
Details: A Boolean circuit has some binary inputs $x_1, x_2, \ldots x_k$, some AND, OR and NOT gates, and a single binary output.

e.g.



In this example, we can make the output 1 by setting all inputs to 1.

The circuit can be described as a string using any reasonable conventions. One easy way is to label all connecting wires with labels like $t_1, t_2, \ldots$ and write out how to compute all the values. The above example would first become



Then write out as:

$t_1 = x_1$ AND x_2 AND x_3 ;

$t_2 =$ NOT x_3 ;

$t_3 = t_2$ OR x_4 ;

output $= t_1$ AND t_2 ;

② SAT (satisfiability)

Main idea: Can a given Boolean formula be true? If so, how?

Details: A Boolean formula has some Boolean variables $x_1, x_2, \dots x_k$. They are combined in a formula with $\vee, \wedge, \neg$ (symbols for AND, OR, NOT). The formula will be in a particular format called CNF (conjunctive normal form) – details given below.

We want to know how to choose values for $x_1, x_2, \dots x_k$ such that the formula is true.

example: $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_4) \wedge (x_5)$

In this example, we can make the formula true by taking
$x_1 = 0$
$x_2 = 0$
$x_3 = 0$
$x_4 = 1$
$x_5 = 1$

A clause is some variables ORed together e.g. $x_1 \vee x_2 \vee \neg x_3$
(some of the variables can be negated).

CNF means some clauses are ANDed together. The above example is in CNF.

(3) 3-SAT

Same as SAT, but each clause has at most 3 variables.

e.g. the above example is a 3-SAT instance, but the following is not:

$$(x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee x_5) \wedge (\neg x_1 \vee \neg x_5)$$