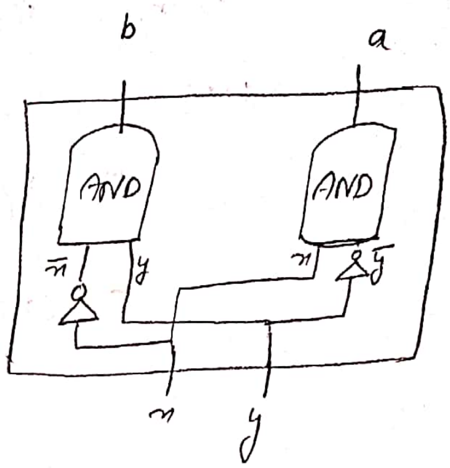
AMIT YADAV

Assignment 3

(H3.1) First of all, a circuit to compare two single bit x and y.

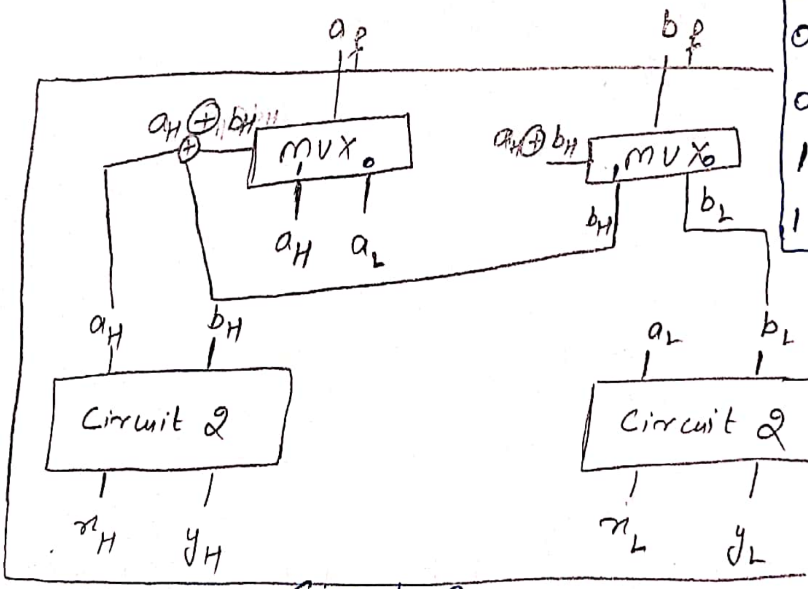
x	y	$x>y$: a	$y>x$: b
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0



Circuit 1

Using this we can compare 2 n-bit numbers using divide and conquer. The main idea is to compare ~~to~~ ~~B~~ lower and higher half in parallel and use their results to get final result.

a_H	b_H	a_f	b_f
0	0	a_L	b_L
0	1	a_H	b_H
1	0	a_H	b_H
1	1	X	X



Circuit 2

Our final answer will be a_f. i.e. $f(x,y) = a_f$

Depth analysis:

Let depth for comparing two n-bit no. is $D(n)$.

Then, in our circuit,

$$D(n) = D(n/2) + c$$

where 'c' includes depth of MUX and XOR gates.

$$= O(c \log n)$$

$$= O(\log n)$$

H3.2

(1) # functions $f: \{0,1\}^n \longrightarrow \{0,1\}$

Total no. of inputs possible $= 2^n$

For each input, we have two choices.

So, # diff outputs possible

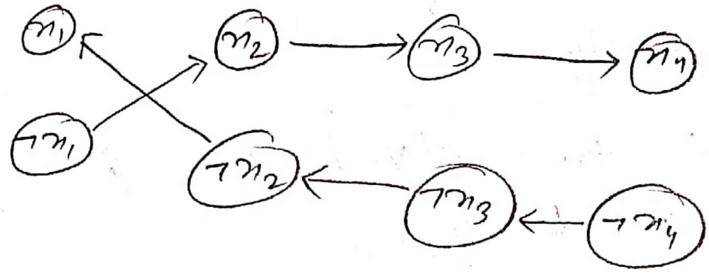
$$= 2^{(2^n)}$$

$$\therefore \# f = 2^{(2^n)}$$

H3.3 Given a boolean formula ϕ in 2CNF, we convert it to a directed graph as follows:

1. Make $2|X|$ nodes, where $X = \{x_1, x_2, x_3 \dots x_n\}$ is set of bool. variables.
2. For any clause $(a \vee b)$, write it as $(\neg a \rightarrow b)$ and $(\neg b \rightarrow a)$. Make an edge ~~for~~ in the direction of implication

So, for example, $\phi = (x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (x_4 \vee \neg x_3)$



2-SAT

To check SAT of ϕ, do this:

1. Find all strongly connected components in the directed graph.
2. If for any bool. variable, x_i and $\neg x_i$, belong to same SCC, then return UNSAT, else SAT.

Correctness

1. Point 2 above is necessary. Because if $\exists x_i$ such that x_i and $\neg x_i$ belong to same SCC, then both need to be given same assignment (i.e. true or false) (why? This follows from the way we constructed the graph) so, ϕ will be UNSAT.

(2) Point (2) above is sufficient.
If $\nexists\, x_i$ s.t. x_i and $\neg x_i$ belong to same SCC, then we can find the truth assignment of variables which satisfy ϕ.
Using algorithm ~~as~~ given in (Aspvall et. al, 1979)
(A linear-time algorithm for testing the truth of certain quantified boolean formulas)

Time Complexity

Finding SCCs takes $O(V+E)$ time for a graph with V vertices and E edges.
For a formula ϕ with $|X|$ variables,

$$V = 2|X|$$

$$E = 2 \times \text{no. of clauses}$$

Checking if $\exists\, x_i$ ~~belongs~~ s.t. x_i and $\neg x_i$ belong to same SCC take time $O(2|x|) = O(|x|)$

$\therefore$ Total time for checking SAT $= O(2|x| + \text{no. of clauses})$ which is linear w.r.t. # clauses.