מבנה מחשבים

תרגול מספר 4

Question 2.3

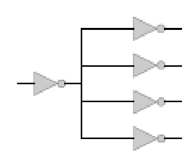
Prove that the total amount of time spent in the relaxation steps is linear in the number nodes if the fan-in of each gate is constant (say, at most 3).

Note that it is not true that each relaxation step can be done in constant time if the fan-in of the gates is not constant.

Prove linear running time in the number of nodes if (i) every net feeds a single input terminal and (ii) the number of outputs of each gate is constant. (You may not assume that the fan-in of every gate is constant.)

פתרון לתרגיל בית 2.3

רשת e	Fan-out(e)
שער G	In-degree(G),out-degree(G)



פתרון לתרגיל בית 2.3

רלקסציה של הרשת ei

- עדכון tpd לכל כניסה שמוזנת ע"י ei.
- סימולציה של פונקציונליות: חישוב פונקציה בוליאנית על מספר הקלטים של שער g שפולט את ei.
- הסיבוכיות הכוללת

$$\sum_{e_i} O(\text{fanout}(e_i)) + O(\text{in}-\deg\text{ree}_{G\_feeds\_e_i}(G))$$

פתרון לתרגיל בית 2.3

$$\sum_{e_i} O(\text{fanout}(e_i)) + O(\text{in-degree}(G_{feeds\_e_i}))$$

$\sum_{e_i} O(\text{fanout}(e_i)) =$ מספר הכניסות במעגל $\le c \cdot n$

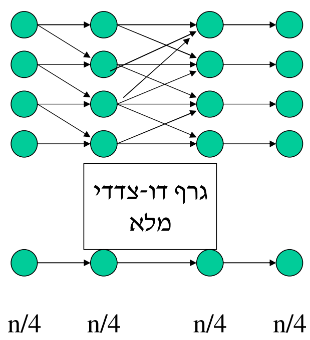
$$\sum_{e_i} O(\text{in-degree}(G_{feeds\_e_i})) = \sum_{G} O(\text{in-degree}(G) \cdot \text{out-degree}(G))$$

$$\le c \sum_{G} \text{out-degree}(G) \le c \sum_{G} \text{in-degree}(G) = O(n)$$

כל יציאה מזינה לפחות כניסה אחת.

פתרון לתרגיל בית 2.3

- דוגמא ל in-degree(G) לא חסום



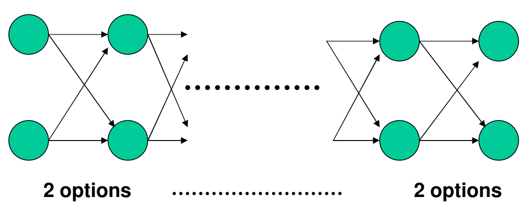
פתרון לתרגיל בית 2.3

$\text{Fanout}(e_i)=1$

Out-degree(G)

- התנאי 1 שקול לכך שמספר היציאות שווה למספר הכניסות.
- דרגת היציאה של כל שער היא קבוע ולכן מספר הקשתות בגרף ליניארי במספר הצמתים.
- סיבוכיות האלגוריתם היא ליניארית במספר הצמתים.

A circuit with $2^{n/2}$ paths



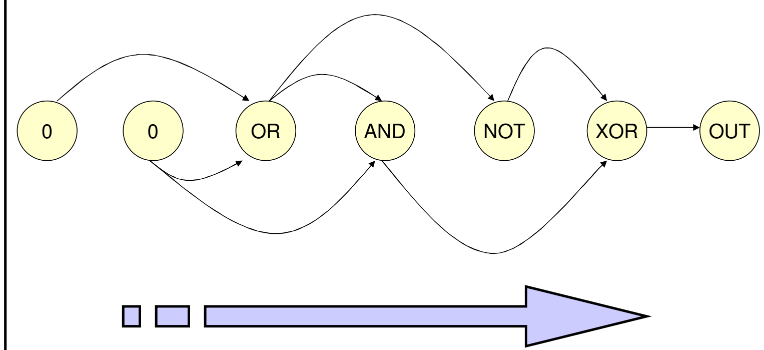
Note: n/2 stages with 2 options each, resulting in $2^{n/2}$ paths.

→ There is no need to check all the paths, only the longest. This takes a linear time.

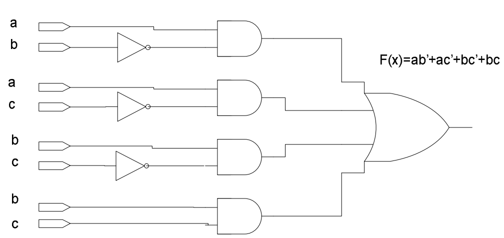
A circuit with 2^n paths

- …cannot be built!
- Why? A combinational circuit is a DAG, therefore we cannot reorder the gates to create different paths. Our only option is to include or exclude gates to create different paths.
- But, having n gates, we only have 2^n such paths. Each gate can be included or excluded, therefore 2^n.

Finding the Maximum Delay



SOP



Definition:

A Boolean function $f:\{0,1\}^n \Rightarrow \{0,1\}$ is *monotone* if $x \geq y \Rightarrow f(x) \geq f(y)$

(where $x \geq y$ means : for every i $x_i \geq y_i$).

Prove the following claim:

$f:\{0,1\}^n \Rightarrow \{0,1\}$ is *monotone* ***iff*** f can be implemented by a combinational circuit that contains only **AND-gates** and **OR-gates**.

Question 3.3

Design a zero-tester defined as follows.
Input: x[n-1 : 0].
Output: y
Functionality:
y = 1 iff x[n-1 : 0] = 0^n:

1. Suggest a design based on an or-tree.
2. Suggest a design based on an and-tree.
3. What do you think about a design based on a tree of nor-gates?

Associative Boolean functions

A Boolean function
$f : \{0,1\}^2 \rightarrow \{0,1\}$ is associative if
$f(f(\sigma_1,\sigma_2),\sigma_3) = f(\sigma_1,f(\sigma_2,\sigma_3))$
for every $\sigma_1, \sigma_2,\sigma_3 \in \{0,1\}$

Question 3.7

Prove that if $n = 2^k$-r, where $0 \le r < 2^{k-1}$,
then any partition in which
$2^{k-1}-r \le a \le 2^{k-1}$ and b = n - a is a balanced partition.

balanced partition

Two positive integers a,b are a balanced partition of n if:

1. a+b = n,
2. $\text{Max}\{\lceil \log_2 a \rceil, \lceil \log_2 b \rceil\} \le \lceil \log_2 n \rceil - 1$

Question 3.8

Consider the following recursive algorithm for constructing a binary tree T_n with $n \geq 2$ leaves. Prove that the depth of T_n is $\lceil \log_2 n \rceil$.

1. The case that $n \leq 2$ is trivial (two leaves connected to a root).
2. If $n > 2$, then let a, b be balanced partition of n.
3. Compute trees T_a and T_b. Connect their roots to a new root to obtain T_n.