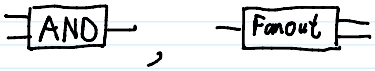
A Brief Introduction to Classical Complexity Theory (circuit version)

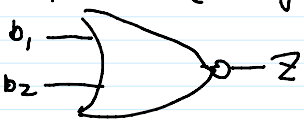
Saturday, June 19, 2021 12:27 AM

## 1. Classical circuit




Fact ∃ a universal gate set U for classical bits compose of gates with 2 bit input, 1 bit output

Example: U={NOR gates}




| $b_1$ | $b_2$ | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Any classical circuit (with bounded gate size) can be converted into gates from U

Original circuit ⟶ Circuit built in U

n Gates poly(n) gates

## 2. Decision problem, P/NP

Language: $L \subseteq \{0,1\}^*$ → all finite string compose of {0,1}

Decision problem($D_L$): Given x, decide if $x \in L$

Example:

$L = \{x \mid x$ is a binary representation of a prime$\}$

$D_L$: Primality testing.

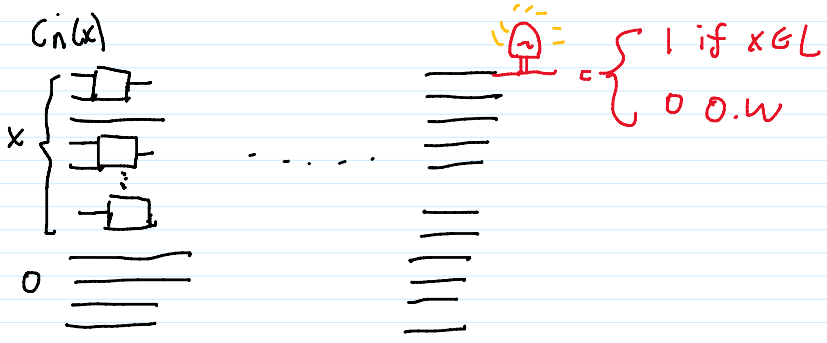
Could be define over $L_{Yes}$, $L_{No}$ with $L_{Yes} \cap L_{No} = \emptyset$

P (Polynomial time): Language $L \in P$, if $D_L$ can be solved in polynomial time.

$\forall n$ ∃ a classical circuit $C_n$, with poly(n) gates

or $C_n(x) = 1$ if $x \in L$

$C_n(x) = 0$ if $x \notin L$




"problem with good algorithms"

NP (Nondeterministic Polynomial time): Language $L \in NP$, if $D_L$ can be "verified" in poly time

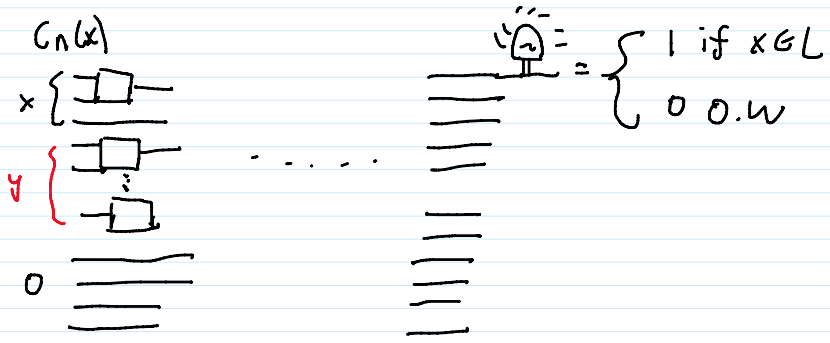
⇒ $\forall n$, ∃ a classical circuit $C_n$, with poly(n) gates

$\forall x \in \{0,1\}^n$: if $x \in L$, $\exists y \in \{0,1\}^{poly(n)}$ ("proof" that $x \in L$) s.t. $C_n(x,y)=1$

$\Rightarrow \forall n$, $\exists$ a classical circuit $C_n$, with poly $(n)$ gates

$\forall x \in \{0,1\}^n$ : if $x \in L$, $\exists y \in \{0,1\}^{poly(n)}$ (= proof that $x \in L$) s.t. $C_n(x,y)=1$

if $x \notin L$, $\forall y \in \{0,1\}^{poly(n)}$ $C_n(x,y)=0$




* Problems with a good validation algorithm's

Example:

$\{(G_1, G_2) \mid G_1, G_2$ graph. $G_1, G_2$ are isomorphic$\}$ (Graph isomorphism problem)

$y$ : isomorphism map between $G_1$ and $G_2$

Reference:

[SIP06]: Introduction to the Theory of Computation