CPSC 421/501 Nov 26, 2025

Cook-Levin Theorem: Let $L \in NP$, specifically L is decided in time $\leq Cn^k$ by a non-deterministic Turing machine M. Then

$L \leq_p 3SAT$: i.e. there is a

(L is P-time reducible to 3SAT)

polynomial time program, taking $s \in \Sigma^n$ ("input") and producing a formula, $f = f(M, s)$, such that

$s \in L \iff f(M, s) \in 3SAT$.

i – step $1, \ldots, Cn^k$
j – cell $1, \ldots, Cn^k$
$\gamma \in \Gamma$, $q \in Q$

Proof idea: Let

$$X_{ij\gamma} = \begin{cases} T & \text{if at step } i, \text{ cell } j \text{ has symbol } \gamma \in \Gamma \\ F & \text{otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} T & \text{if tape head is over cell } j \text{ at step } i \\ F & \text{otherwise} \end{cases}$$

$$Z_{iq} = \begin{cases} T & \text{if we are in state } q \text{ at step } i \\ F & \text{otherwise} \end{cases}$$

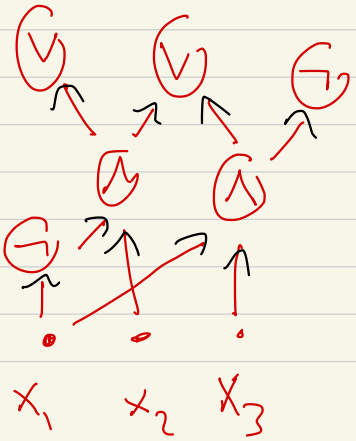
Then — a mess of Boolean algebra

Chap 9 of [Sip]

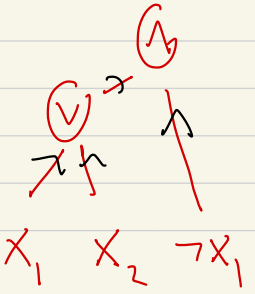
[Hidden strategy to trying to show $P \neq NP$ $P = NP$]

Scholium: If $L \in P$, then for each $n \in \mathbb{N}$, there is a circuit C_n of size poly(n) that on input $s \in \Sigma^n$ computes whether or not $s \in L$.

Circuit: outdegree large



formula: outdegree 1 etc.



M is non-det T.m. We'll run M for t steps ($t = Cn^k$) on an input $s \in \Sigma^n$, we want a formula (massive + redundancy--) $f = f(s) = f(M,s)$ is satisfiable iff s is accepted by M within time t.

Formula (see p. 5):

(1) step 1 must be blah blah blah

AND

(2) step 1 to step 2 must be OK with respect to M

AND

(3) step 2 to step 3 --

some element of $\delta(q_i, \sigma_j)$

AND — 4, 5 — must be (q', σ', R)

⋮

AND

(t) step $t-1$ to step t CK w.r.t. M

AND

$(t+1)$ state at step t = q_{acc}

So ...

cond 1 AND cond 2 AND cond 3 AND ...

3CNF: (3 literals OR-ed together) AND

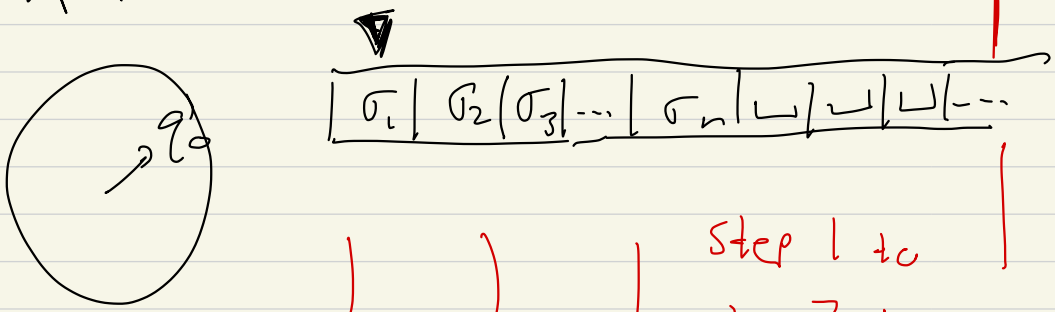
(" " " ") AND

⋮

eventually ⇝ 3CNF formula

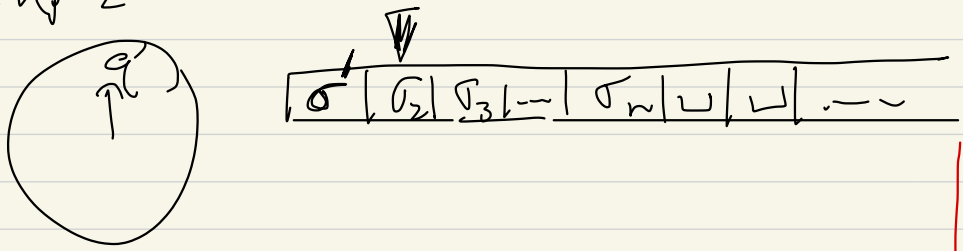
Step 1:

input $s = \sigma_1 \cdots \sigma_n$



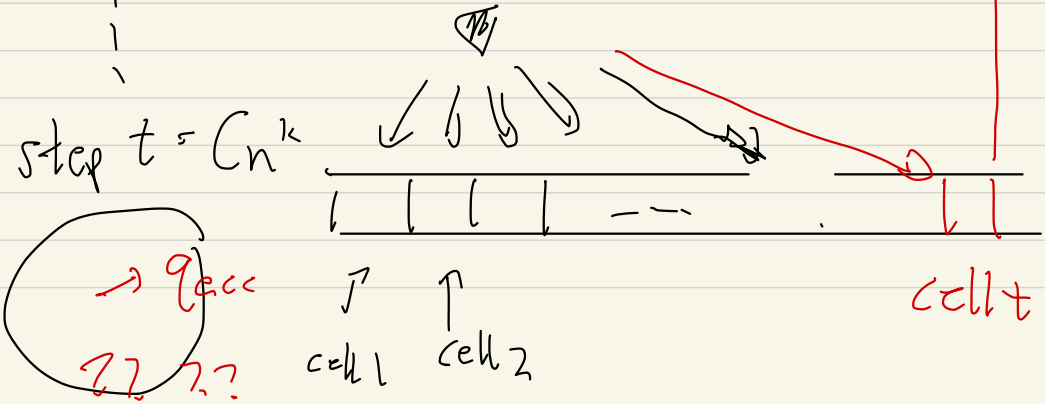
step 1 to step 2 has to be OK with M

Step 2



step 3

step $t = Cn^k$



Proof idea: Let

i - step $1, \ldots, Cn^k$

j - cell $1, \ldots, Cn^k$

$\gamma \in \Gamma$, $q \in Q$

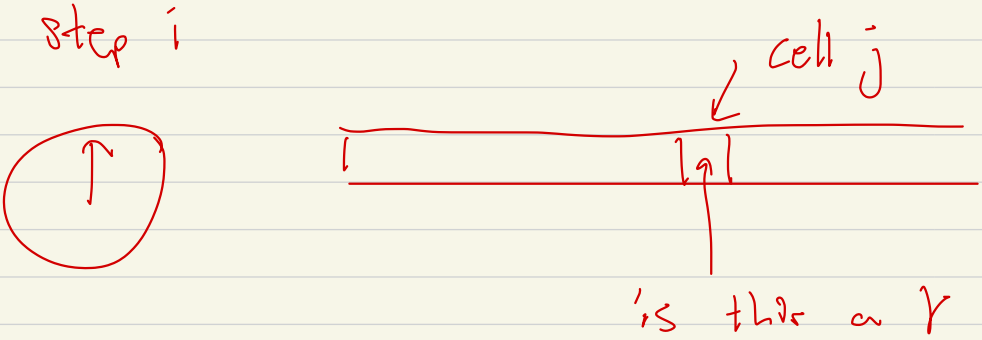
$$X_{ij\gamma} = \begin{cases} T & \text{if at step } i, \text{ cell } j \text{ has symbol } \gamma \in \Gamma \\ F & \text{otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} T & \text{if tape head is over cell } j \text{ at step } i \\ F & \text{otherwise} \end{cases}$$

$$Z_{iq} = \begin{cases} T & \text{if we are in state } q \text{ at step } i \\ F & \text{otherwise} \end{cases}$$

Then — a mess of Boolean algebra

$$X_{ijr} = \begin{cases} T & \text{if at step } i, \text{ cell } j \text{ has symbol } r \in \Gamma \\ F & \text{otherwise} \end{cases}$$



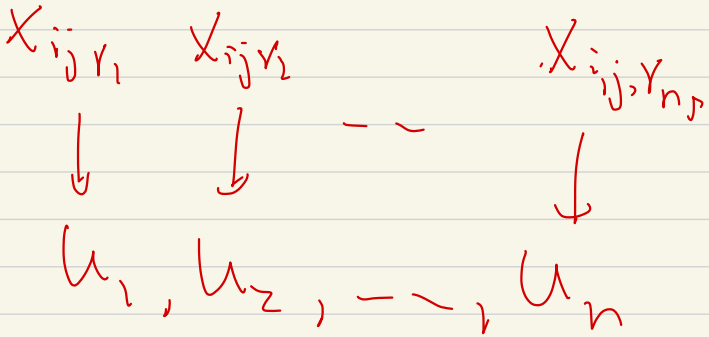
for all $i, j \in [t]$, $r \in \Gamma$

So:

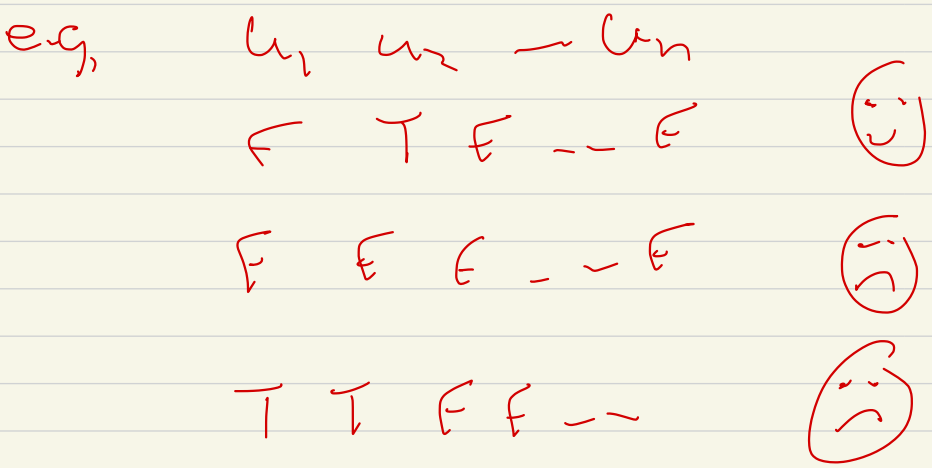
among all X_{ijr}, with i, j fixed

$\Gamma = \{r_1, r_2, \ldots, r_{n_\Gamma}\}$

(exactly one of $X_{ijr_1}, X_{ijr_2}, X_{ijr_3}, \ldots$ $X_{i,j,r_{n_\Gamma}}$ should be true



exactly one of $u_1, \ldots, u_n$ is T, the rest are F



$$\left(u_1 \text{ OR } u_2 \text{ OR } u_3 \text{ OR } \ldots \text{ OR } u_n\right)^2$$

3CNF

can we write in 3CNF

AND

(1) not that U_1 and $U_2 = T$

AND

:: :: U_1 and $U_3 = T$

AND

⋮

AND

not true that U_{n-1} AND $U_n = T$

---

(1) $\leftrightarrow \neg(U_1 \text{ and } U_2)$

$= (\neg U_1) \text{ OR } (\neg U_2)$

$= (\neg U_1) \text{ OR } (\neg U_2) \text{ OR } (\neg U_3)$

(2)

U_1 OR U_2 OR U_3 — OR U_n

$n \leq 3$ ☺ $U_1 = U_1$ OR U_1 OR U_1

What about

U_1 OR U_2 OR U_3 OR U_4 ?

Trick: add W new variable

$U_1 \vee U_2 \vee U_3 \vee U_4 = T$

$\Leftrightarrow$

$(U_1 \vee U_2 \vee W)$ AND $(\neg W \vee U_3 \vee U_4)$

is satisfiable, i.e. via $W = T$ or $W = F$