Cook-Levin Theorem

Theorem : SAT is NP-complete.

We have to show (1) $SAT \in NP$

(2) $\forall A \in NP,\ A \leq_p SAT$.

We have already seen (1). For showing (2), we use an approach similar to computation histories.

SAT $\in$ NP: Guess TRUE/FALSE for $x_1, x_2, \ldots x_n$. Verify if ϕ is satisfied for the guessed assignment.

The main part is (2). We need to show that any language $A \in NP$ must reduce to SAT. That is, we need f such that

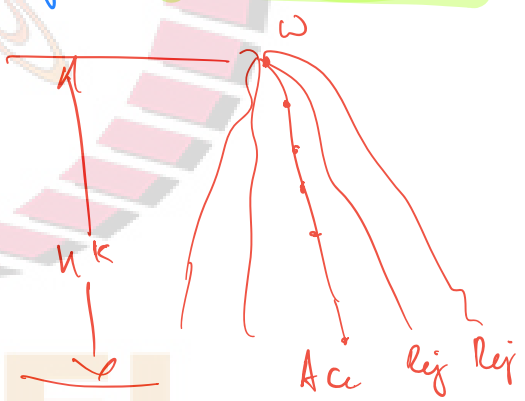
$$w \in A \iff f(w) \in SAT.$$

All we know about A is that $A \in NP$. We cannot assume any other structure of any specific language or problem.

$A \in NP$: This means that A is decided by an NTM N in time n^k. When $w \in A$, there is a sequence of computations that leads to N accepting w. This sequence has $\leq n^k$ steps. When $w \notin A$, no sequence of computations lead to accept.

Let $N = (Q, \Sigma, \Gamma, \delta, q_s, q_a, q_r)$. n denotes $|w|$.

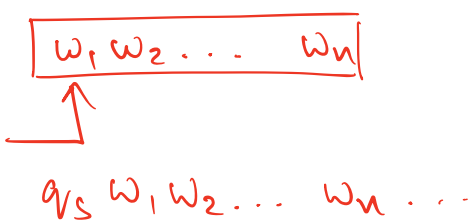
Let N be a 1-tape NTM that runs in time $\leq n^k$. (actually $n^k - 3$). Let us define $\Delta = Q \cup \Gamma \cup \{\#\}$.



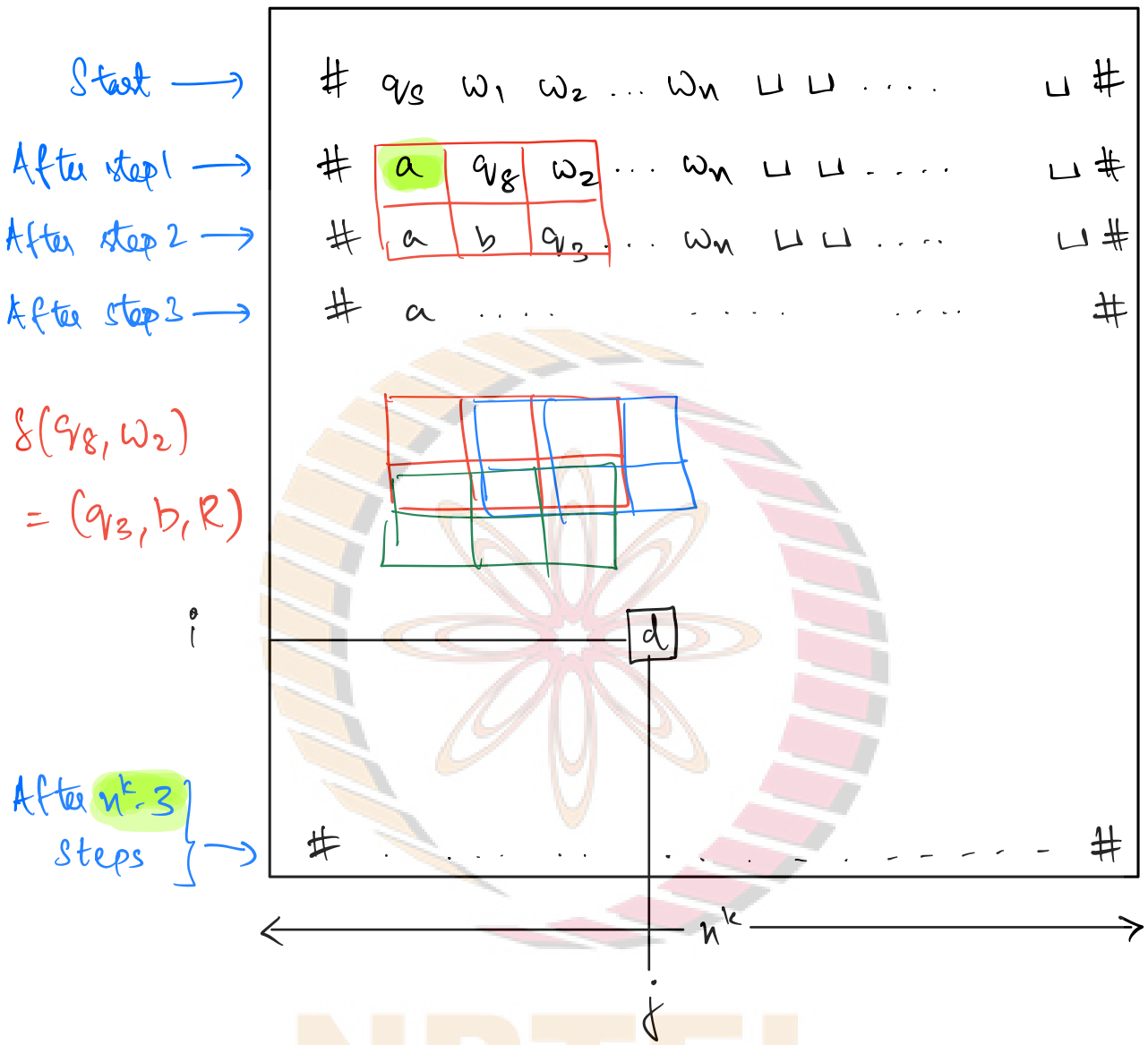
Gadget for reduction : Computation table or tableau

This is a table of successive configurations. Each row is a configuration.

$\delta(q_s, w_1) = \{(q_3, c, R), (q_6, a, R), \ldots\}$ q_s



Computation Table for $w \in L$



$x_{i,j,d}$ = TRUE

$x_{i,j,a}$ = FALSE

$x_{i,j,\#}$ = FALSE

~~$x_{i,j,b}$ = TRUE~~

If TM accepts / rejects before n^k steps, the config remains unchanged after that.

We will create a formula ϕ which uses Boolean logic to check whether the tableau represents a valid accepting computation of N on w.

If there exists a path for N to accept w, then ϕ will have a satisfying assignment. Else ϕ won't have a satisfying assignment.

ϕ checks the following:

(1) Does N start correctly?
(2) Does N move correctly?
(3) Does N end correctly?
(4) Do the variables of ϕ form a "proper encoding" of the table?

$$\phi = \phi_{start} \wedge \phi_{move} \wedge \phi_{accept} \wedge \phi_{cell}$$

We have an $n^k \times n^k$ table. We have $|\Delta|$ symbols that can occupy each cell. The Boolean variables in ϕ are given below.

$x_{i,j,\ell}$ = { TRUE if $(i,j)^{th}$ cell has entry ℓ ; FALSE if $(i,j)^{th}$ cell has entry $\neq \ell$ }

$1 \le i, j \le n^k$

$\ell \in \Delta$

We have $n^k \times n^k \times |\Delta|$ variables overall.

ϕ_{cell} : Is the table properly encoded?

$$\phi_{cell} = \bigwedge_{1 \le i,j \le n^k} \left[\left(\bigvee_{\ell \in \Delta} x_{i,j,\ell} \right) \wedge \left(\bigwedge_{\substack{\ell, \ell' \in \Delta \\ \ell \neq \ell'}} \left[\overline{x_{i,j,\ell}} \vee \overline{x_{i,j,\ell'}} \right] \right) \right]$$

Cell (i,j) contains at least one entry

Cell (i,j) does not contain > 1 entry.

So ϕ_{cell} ensures that each cell (i,j) contains exactly one member of Δ.

ϕ_{start} : First row is a legal starting configuration for N on w.

$$\phi_{start} = x_{1,1,\#} \wedge x_{1,2,q_{start}} \wedge x_{1,3,w_1} \wedge x_{1,4,w_2} \wedge$$
$$\cdots \wedge x_{1,n+2,w_n} \wedge x_{1,n+3,\sqcup} \wedge \cdots$$
$$\cdots \wedge x_{1,n^k-1,\sqcup} \wedge x_{1,n^k,\#}$$

ϕ_{accept} : The table represents an accepting computation.

$$\phi_{accept} = \bigvee_{1 \le i,j \le n^k} \left(x_{i,j,q_{accept}} \right)$$

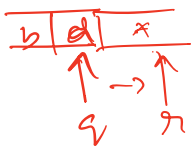
ϕ_{move} : This is the hardest. We have to check that each configuration legally follows from the previous one.

The main idea here is that it is enough to check all the 2×3 windows. We say a window is valid if it can be part of a valid transition.

b	q	a
b	d	r

When $(r, d, R) \in \delta(q, a)$

$\forall b \in \Gamma \cup \{\#\}$



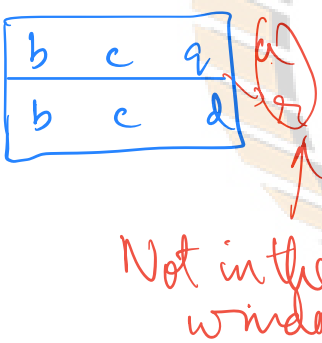
b	q	a
r	b	d

When $(r, d, L) \in \delta(q, a)$

$\forall b \in \Gamma$

#	q	a
#	r	d

When $(r, d, L) \in \delta(q, a)$



If $(r, d, R) \in \delta(q, a)$ for some $a \in \Gamma$, where $b \in \Gamma \cup \{\#\}$, $c \in \Gamma$

q	a	b
d	r	b

If $(r, d, R) \in \delta(q, a)$ where $b \in \Gamma \cup \{\#\}$

a	b	c
a	b	c

Where $a \in \Gamma \cup \{\#\}$, $b, c \in \Gamma$

OR $a, b \in \Gamma$, $c \in \Gamma \cup \{\#\}$.

We can list all the valid windows in this manner. The number of valid windows is finite and depends only on N. It is independent of |w|.

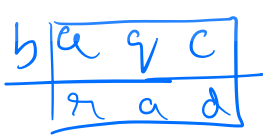
Claim: If all the 2×3 windows are valid, then all C_i are legally followed by C_{i+1}.

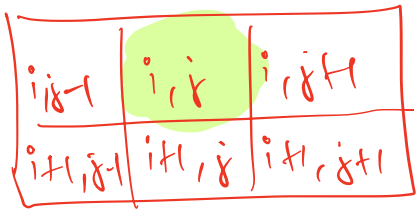
Proof: (1) If a symbol is not adjacent to the state, then it is unchanged.

True because middle symbol is unchanged in windows that do not have state, in the top row of the window.

(2) If a symbol is adjacent to state, it appears in a window where the state is in the top middle.

This window captures a valid move for N.



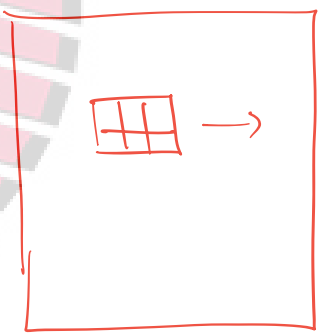


Now we can write ϕ_{mono}.

$$\phi_{mono} = \bigwedge_{1 \le i,j \le n^k} (i,j)^{th} \text{ window is legal}$$

$$\bigvee \begin{array}{|c|c|c|}\hline a_1 & a_2 & a_3 \\ \hline a_4 & a_5 & a_6 \\ \hline \end{array} \text{ is a valid window}$$

$$\left[x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6} \right]$$



We have shown that

$$\phi \in SAT \iff N \text{ accepts } w$$

$$\iff w \in A$$

We also need to show that the reduction, i.e. ϕ, can be constructed in poly time.

There are $n^k * n^k * |\Delta|$ variables $= O(n^{2k})$

↳ independent of $|w|$.

$\phi_{cell} = O(n^{2k})$ length

$\phi_{start} = n^k$ length

$\phi_{accept} = n^{2k}$ length

$\phi_{move} =$ No. of valid windows $\Big\}$ $* \; n^{2k} \; * 6$

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

So ϕ is poly size in n. There is an easy algorithm to generate ϕ, as it has a repetitive structure.

$$(x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_3 \vee \bar{x}_4 \vee x_5) \wedge (x_1 \vee \bar{x}_6 \vee x_7) \wedge \ldots$$

Theorem: 3-SAT is NP-complete.

One way to show is to show $SAT \leq_p 3\text{-}SAT$.

Slightly easier: To modify the above proof so as to generate a 3-CNF SAT instance.

Exercise: Show that the ϕ generated can be modified into a CNF instance. Show that ϕ_{cell}, ϕ_{start}, ϕ_{accept}, ϕ_{move} can be written in CNF form.

Exercise: Show that any CNF formula has an equivalent 3-CNF expression.

$$x_1 \iff (x_1 \vee x_1 \vee x_1)$$

$$(x_1 \vee x_2 \vee x_3 \vee x_4) \iff (x_1 \vee x_2 \vee y_1) \wedge (\bar{y}_1 \vee x_3 \vee x_4)$$

$$(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5) \iff (x_1 \vee x_2 \vee y_1) \wedge (\bar{y}_1 \vee x_3 \vee y_2) \wedge (\bar{y}_2 \vee x_4 \vee x_5)$$

Exercise: Show that 2-SAT $\in$ P.

2-SAT: CNF formula $C_1 \wedge C_2 \wedge C_3 \wedge \ldots$

where each clause is of the form $(x_i \vee x_j)$

Key idea: Every 2-SAT clause can be viewed as an implication.

Distributing OR over AND's.

$(a \wedge b) \vee (c \wedge d) \Leftrightarrow (a \vee c) \wedge (a \vee d) \wedge (b \vee c) \wedge (b \vee d)$