香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF DATA SCIENCE
數據科學學院

CSC 6011 - Theory of Computation

Lecture 13 - Cook-Levin Theorem

Time: Mon & Wed, 4:00 - 5:20 PM

Location: Teaching A, 107

Instructor: Tao LIN (林涛)

TA: Yefan GAO (高叶繁)

Fall 2026

Homework 3 has been posted.

Due: Nov 13 (Fri), 11:59 pm

Take a look before Midterm

Lecture 13 - Cook-Levin Theorem

Last time:

- NP-completeness
- $3\text{SAT} \leq_m^P \text{CLIQUE}$
- $3\text{SAT} \leq_m^P \text{HAMPATH}$

Today:

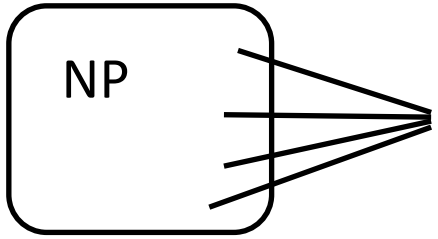
- Cook-Levin Theorem:
  SAT is NP-complete
- 3SAT is NP-complete

Quick Review

Defn: B is NP-complete if

1) $B \in$ NP

2) For all $A \in$ NP, $A \leq_m^P B$ (NP-hard)



today previously

$\leq_m^P$ SAT $\leq_m^P$ 3SAT $\leq_m^P$ CLIQUE

3SAT $\leq_m^P$ HAMPATH

$\leq_m^P$ UHAMPATH

homework

Cook-Levin Theorem:

SAT is NP-complete

So if SAT $\in$ P, then P = NP

Today we prove the starting point:
SAT and 3SAT are NP-complete.

Importance of NP-completeness

1) Evidence that a problem is computationally intractable.

2) A candidate for proving P $\neq$ NP.

To prove C is NP-complete:

Show $C \in$ NP, and reduce a known NP-complete problem to C.

Some Notations

Given Boolean formulas $\varphi_1, \varphi_2, \ldots, \varphi_n$, denote:

Big-AND: $\bigwedge_{1 \le i \le n} \varphi_i = \varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n$ True when all of $\varphi_1, \varphi_2, \ldots, \varphi_n$ are True.

Big-OR: $\bigvee_{1 \le i \le n} \varphi_i = \varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n$ True when some of $\varphi_1, \varphi_2, \ldots, \varphi_n$ is True.

Cook-Levin Theorem (idea)

Theorem: SAT is NP-complete

Proof: 1) $\mathrm{SAT} \in$ NP (done)

2) Show that for each $A \in$ NP we have $A \leq_m^P \mathrm{SAT}$:

Let $A \in$ NP be decided by NTM M in time n^k.

Our goal: Give a polynomial-time reduction f mapping A to SAT.

f: $\Sigma^* \rightarrow$ Boolean formulas

$f(w) = \langle \varphi_{M,w} \rangle$

such that $w \in A$ iff $\varphi_{M,w}$ is satisfiable

Idea: $\varphi_{M,w}$ simulates M on w. Design $\varphi_{M,w}$ to "say" M accepts w.

A satisfying assignment encodes an accepting "computation history" of M on w.

Computation History

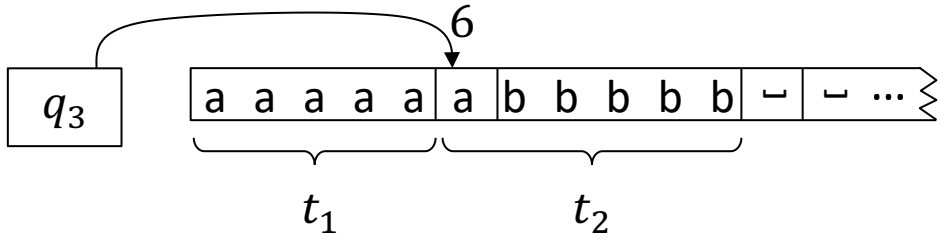
Defn: A <u>configuration</u> of a TM is a triple (q, p, t) where

q = the state,

p = the head position,

t = tape contents

representing a snapshot of the TM at a point in time.



Configuration: $(q_3, 6, \text{aaaaaabbbbb})$

Encoding as a string: $\text{aaaaa}q_3\text{abbbbb}$

Encode configuration (q, p, t) as the string $t_1 q t_2$ where
$t = t_1 t_2$ and the head position is on the first symbol of t_2.

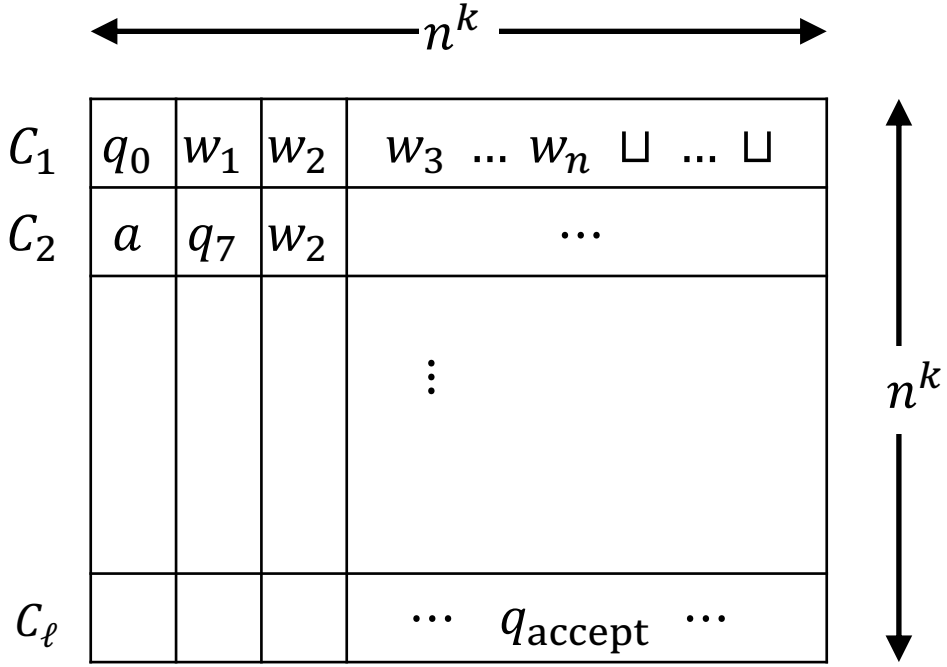
Defn: An (accepting) <u>computation history</u> of M on w is a finite sequence of configurations

$$C_1 \rightarrow C_2 \rightarrow \cdots \rightarrow C_\ell$$

start configuration on w — legal transition of M — an accepting configuration

Computation History in a Table

Write an accepting computation history $C_1 \rightarrow C_2 \rightarrow \cdots \rightarrow C_\ell$ in a table:

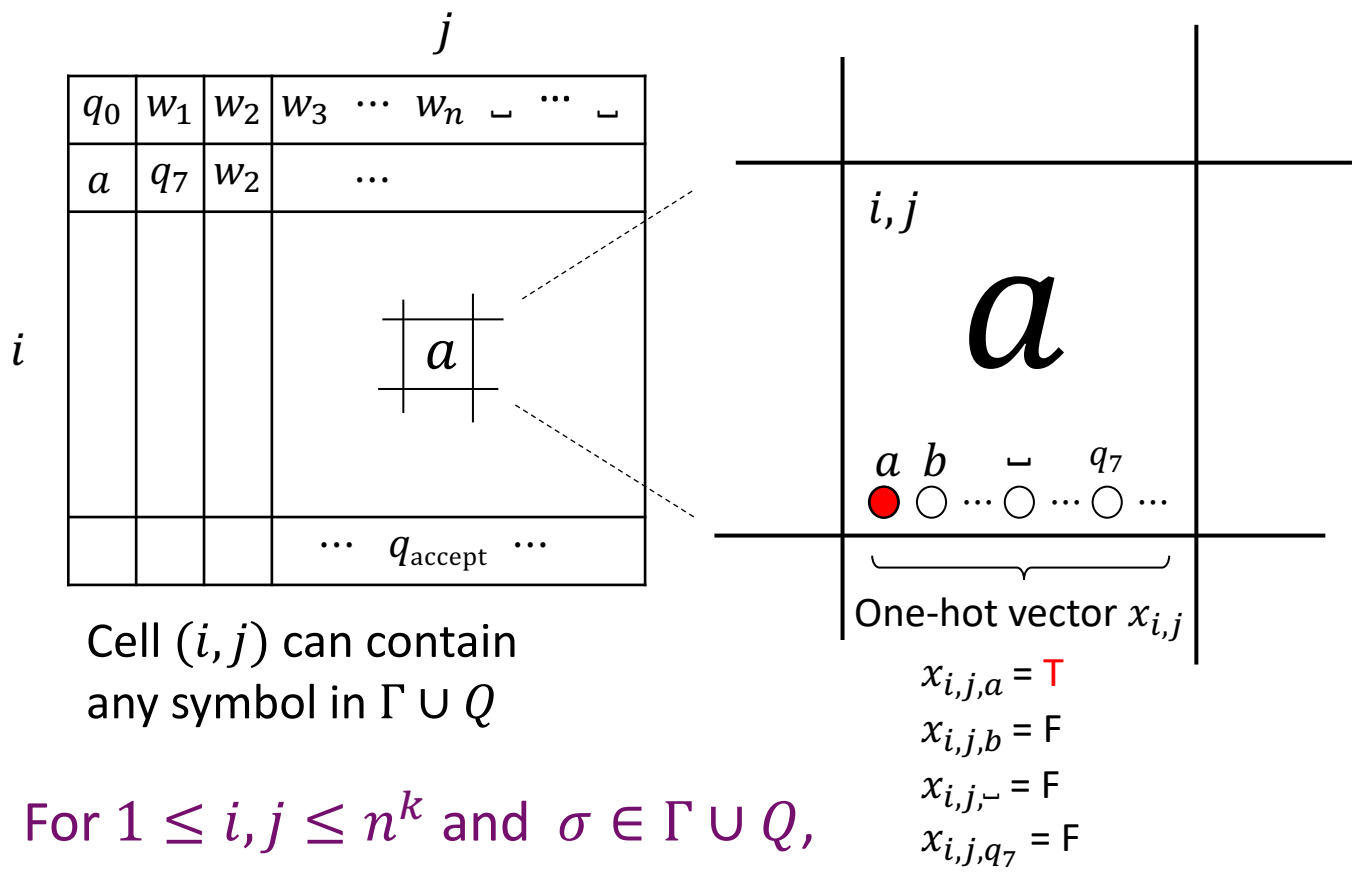


Given M, for any w,
we want to construct $\varphi_{M,w}$ to "say" M accepts w:
"an accepting computation history table exists"

$$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$$

Because the NTM M runs in time n^k,
the size of the computation history table for any branch is at most $n^k \times n^k$

Constructing $\varphi_{M,w}$: φ_{cell}



Cell (i, j) can contain any symbol in $\Gamma \cup Q$

$x_{i,j,a}$ = T
$x_{i,j,b}$ = F
$x_{i,j,␣}$ = F
x_{i,j,q_7} = F

For $1 \leq i, j \leq n^k$ and $\sigma \in \Gamma \cup Q$, introduce Boolean variable $x_{i,j,\sigma}$

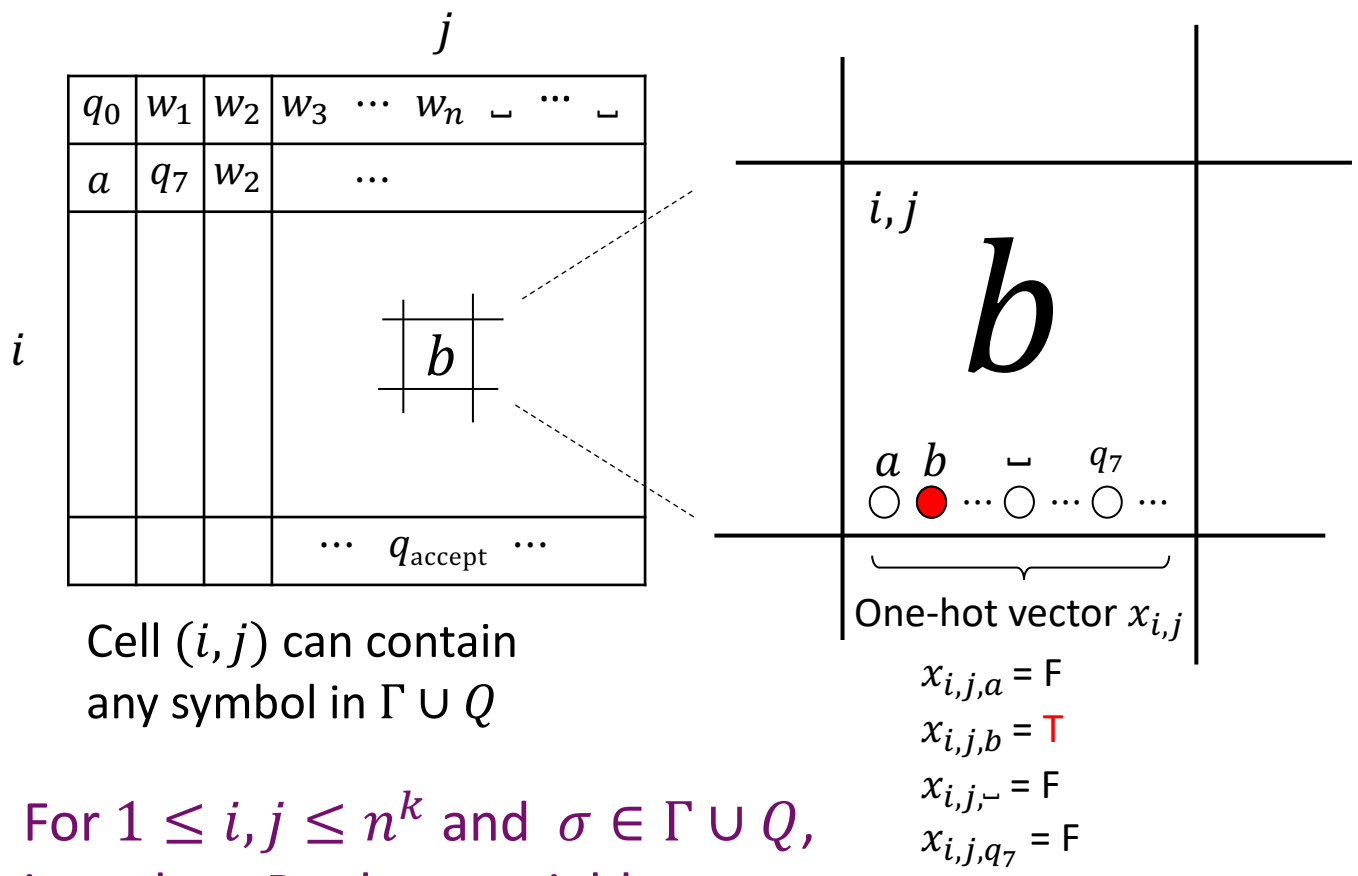
$x_{i,j,\sigma}$ = TRUE means cell (i, j) contains σ.

$\varphi_{M,w}$ "says" that, for M on w, an accepting computation history table exists.

$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$

φ_{cell} "says" that each cell in the table contains a valid symbol.

Constructing $\varphi_{M,w}$: φ_{cell}



Cell (i,j) can contain any symbol in $\Gamma \cup Q$

$x_{i,j,a}$ = F
$x_{i,j,b}$ = T
$x_{i,j,\text{␣}}$ = F
x_{i,j,q_7} = F

For $1 \le i,j \le n^k$ and $\sigma \in \Gamma \cup Q$, introduce Boolean variable $x_{i,j,\sigma}$

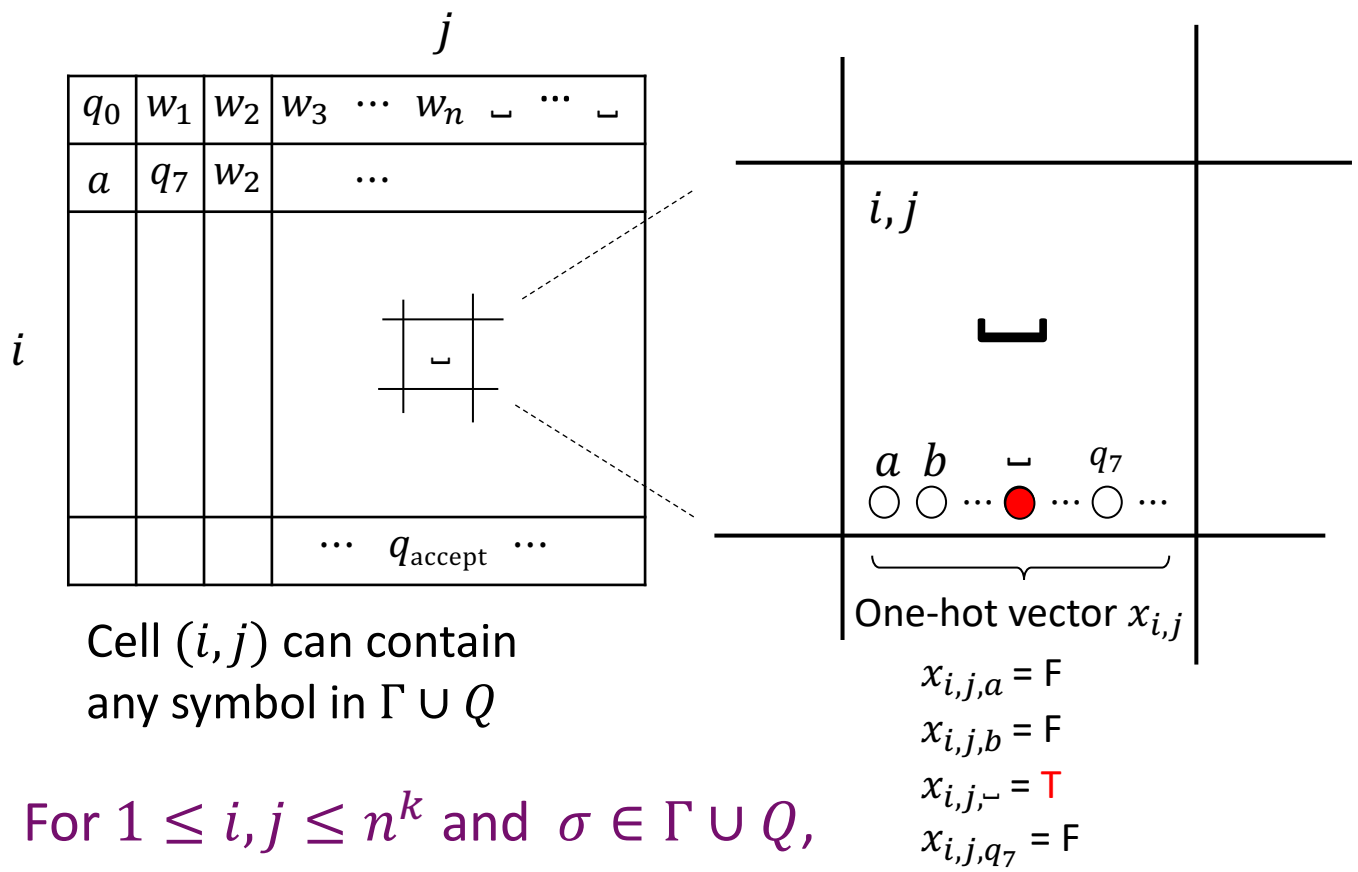
$x_{i,j,\sigma}$ = TRUE means cell (i,j) contains σ.

$\varphi_{M,w}$ "says" that, for M on w, an accepting computation history table exists.

$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$

φ_{cell} "says" that each cell in the table contains a valid symbol.

Constructing $\varphi_{M,w}$: φ_{cell}



Cell (i, j) can contain any symbol in $\Gamma \cup Q$

For $1 \leq i, j \leq n^k$ and $\sigma \in \Gamma \cup Q$, introduce Boolean variable $x_{i,j,\sigma}$

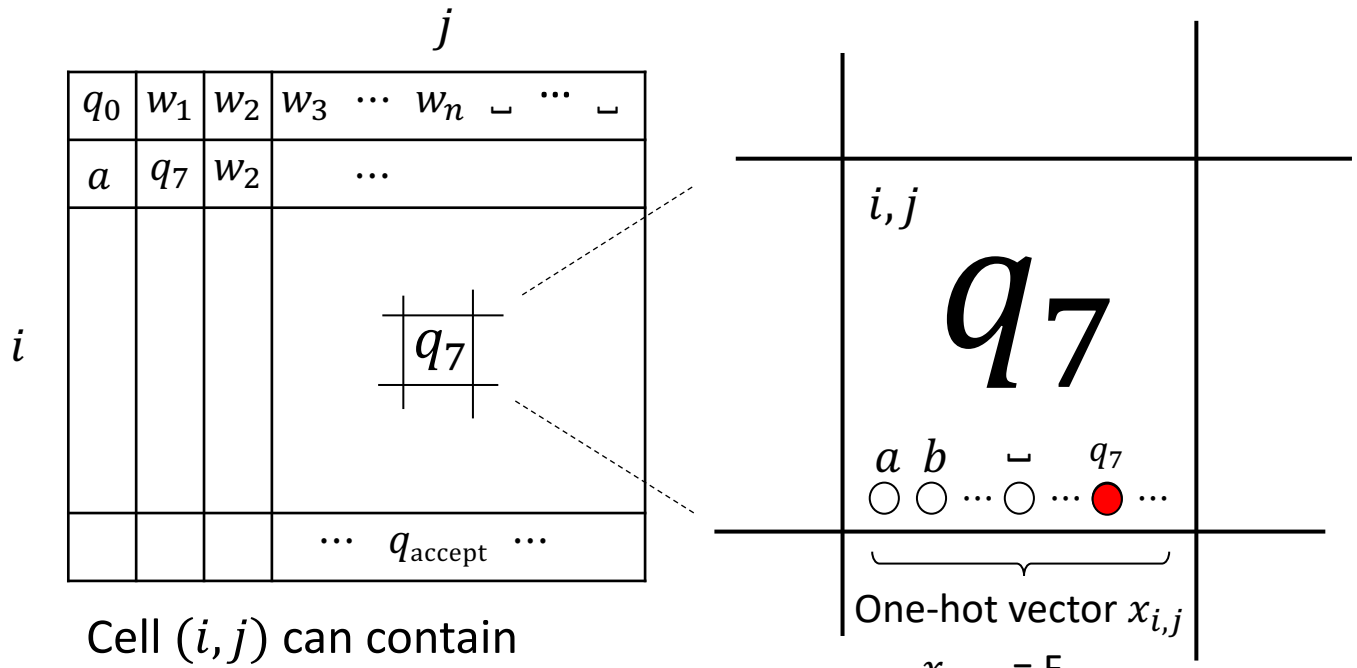
$x_{i,j,\sigma}$ = TRUE means cell (i, j) contains σ.

$\varphi_{M,w}$ "says" that, for M on w, an accepting computation history table exists.

$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$

φ_{cell} "says" that each cell in the table contains a valid symbol.

Constructing $\varphi_{M,w}$: φ_{cell}



Cell (i,j) can contain any symbol in $\Gamma \cup Q$

For $1 \le i,j \le n^k$ and $\sigma \in \Gamma \cup Q$, introduce Boolean variable $x_{i,j,\sigma}$

$x_{i,j,\sigma}$ = TRUE means cell (i,j) contains σ.

$\varphi_{M,w}$ "says" that, for M on w, an accepting computation history table exists.

$$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$$

φ_{cell} "says" that each cell in the table contains a valid symbol.
(i.e., in every cell, exactly one $x_{i,j,\sigma}$ = T)

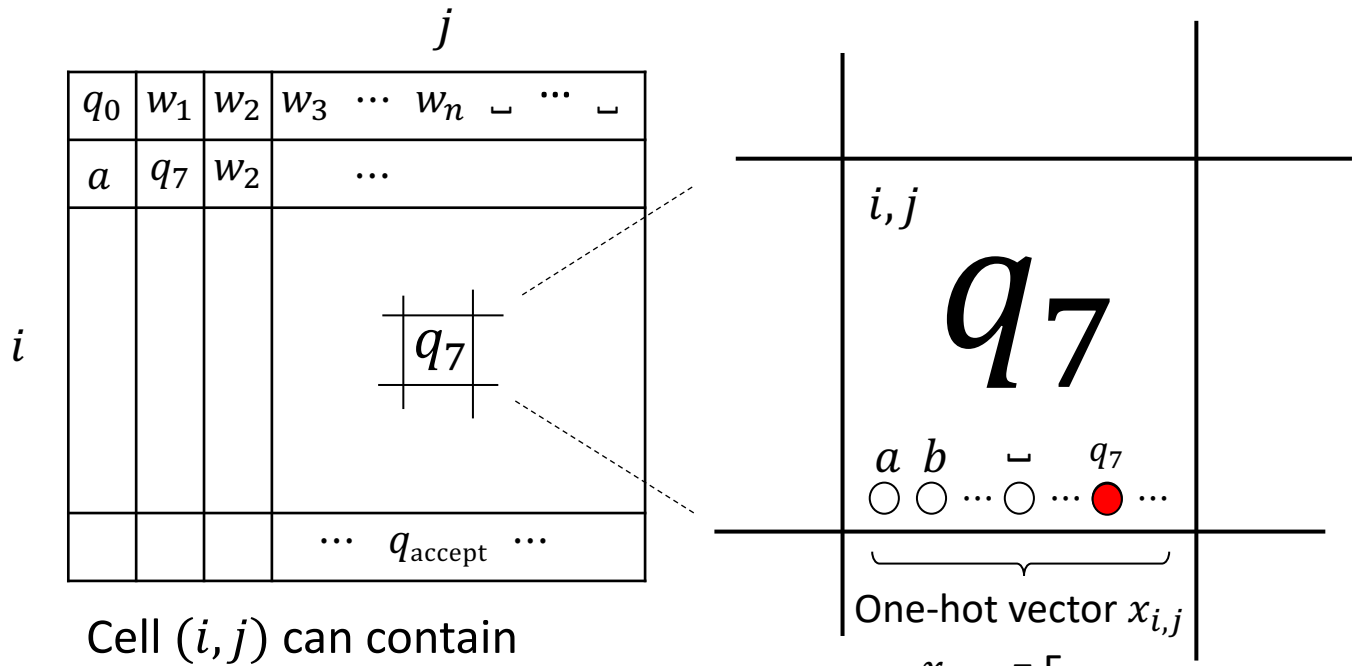
In every cell at least one $x_{i,j,\sigma}$ = T and at most one = T

$$\varphi_{\text{cell}} = \bigwedge_{1 \le i,j \le n^k} \left(\underbrace{\bigvee_{\sigma \in \Gamma \cup Q} x_{i,j,\sigma}}_{x_{i,j,\sigma_1} \vee x_{i,j,\sigma_2} \vee \cdots \vee x_{i,j,\sigma_t}} \wedge \bigwedge_{\substack{\sigma,\tau \in \Gamma \cup Q \\ \sigma \neq \tau}} \left(\overline{x_{i,j,\sigma} \wedge x_{i,j,\tau}} \right) \right)$$

Mini-Quiz 13.1: How many variables does $\varphi_{M,w}$ have? Recall that $n = |w|$.

(a) $O(n^2)$ (b) $O(n^k)$ (c) $O(n^{2k})$ (d) $O(n^{2k+1})$

$n^k \times n^k \times |\Gamma \cup Q|$



Cell (i,j) can contain any symbol in $\Gamma \cup Q$

$\varphi_{M,w}$ "says" that, for M on w, an accepting computation history table exists.

$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$

φ_{cell} "says" that each cell in the table contains a valid symbol.

(i.e., in every cell, exactly one $x_{i,j,\sigma} = \text{T}$)

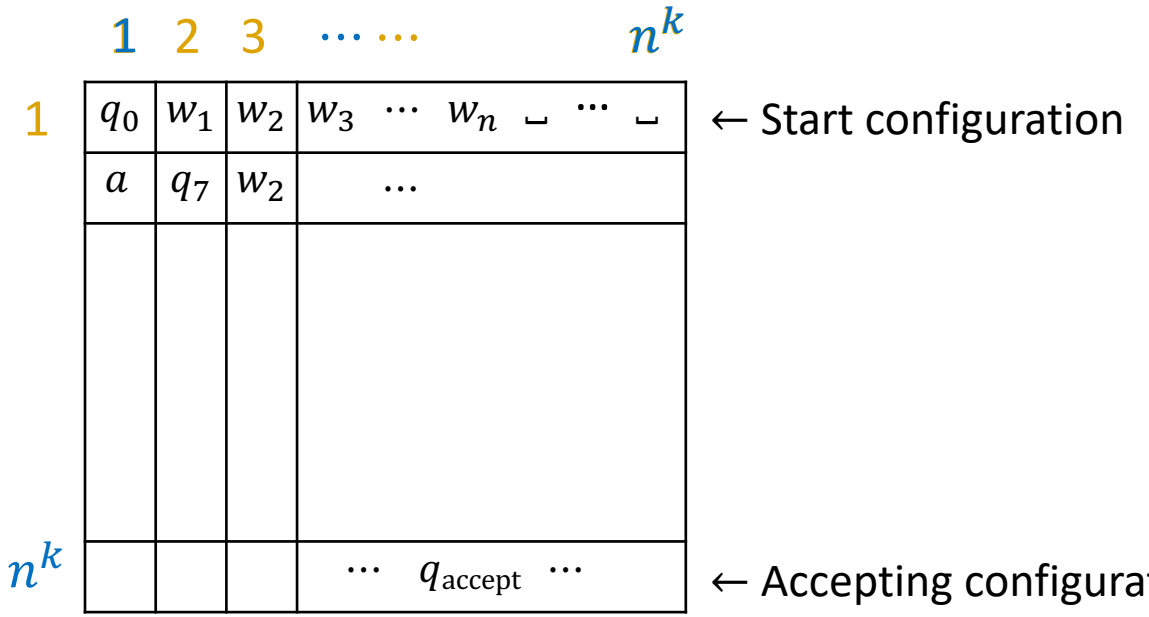
For $1 \le i,j \le n^k$ and $\sigma \in \Gamma \cup Q$, introduce Boolean variable $x_{i,j,\sigma}$

$x_{i,j,\sigma} = \text{TRUE}$ means cell (i,j) contains σ.

In every cell at least one $x_{i,j,\sigma} = \text{T}$ and at most one = T

$$\varphi_{\text{cell}} = \bigwedge_{1 \le i,j \le n^k} \left(\underbrace{x_{i,j,\sigma_1} \vee x_{i,j,\sigma_2} \vee \cdots \vee x_{i,j,\sigma_t}}_{\bigvee_{\sigma \in \Gamma \cup Q} x_{i,j,\sigma}} \wedge \bigwedge_{\substack{\sigma,\tau \in \Gamma \cup Q \\ \sigma \neq \tau}} \left(\overline{x_{i,j,\sigma} \wedge x_{i,j,\tau}} \right) \right)$$

Constructing $\varphi_{M,w}$: φ_{start} and φ_{accept}



← Start configuration

← Accepting configuration

$\varphi_{M,w}$ "says" that, for M on w, an accepting computation history table exists.

$$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$$

φ_{cell} : done ✓

φ_{start} = "The first row is the correct start configuration"

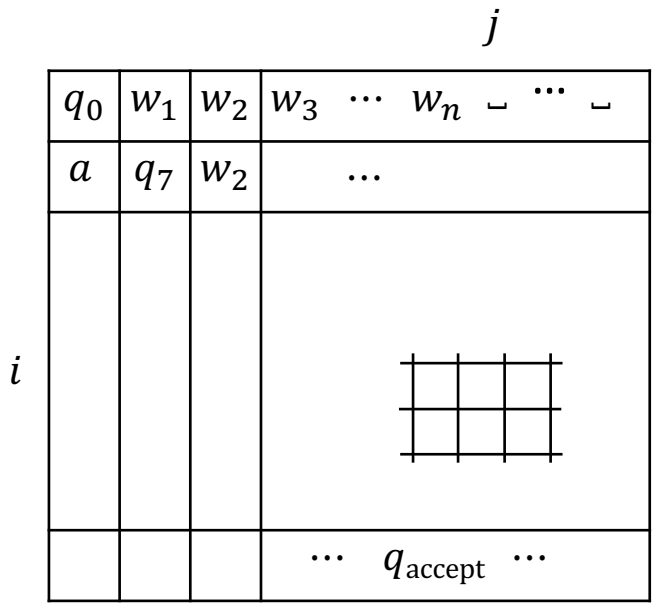
$$= x_{1,1,q_0} \wedge x_{1,2,w_1} \wedge x_{1,3,w_2} \wedge \cdots \wedge x_{1,n^k,␣}$$

φ_{accept} = "The last row is an accepting configuration"

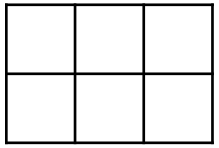
$$= \bigvee_{1 \le j \le n^k} x_{n^k,j,q_{\text{accept}}}$$

(Assume M has a single accepting state q_{accept})

Constructing $\varphi_{M,w}$: φ_{move}



2×3 neighborhood



Legal neighborhoods:

consistent with M's transition function

potential examples:

a	q_7	b
q_3	a	c

a	b	c
a	b	c

a	b	c
a	b	q_5

a	b	c
d	b	c

Illegal neighborhoods:

not consistent with M's transition function

examples:

a	b	c
a	d	c

a	b	c
a	q_2	c

a	q_7	c
a	b	c

a	q_7	c
q_3	d	q_4

$\varphi_{M,w}$ "says" that, for M on w, an accepting computation history table exists.

Claim: If every 2×3 neighborhood is legal, then the table corresponds to a valid computation history.

$$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$$

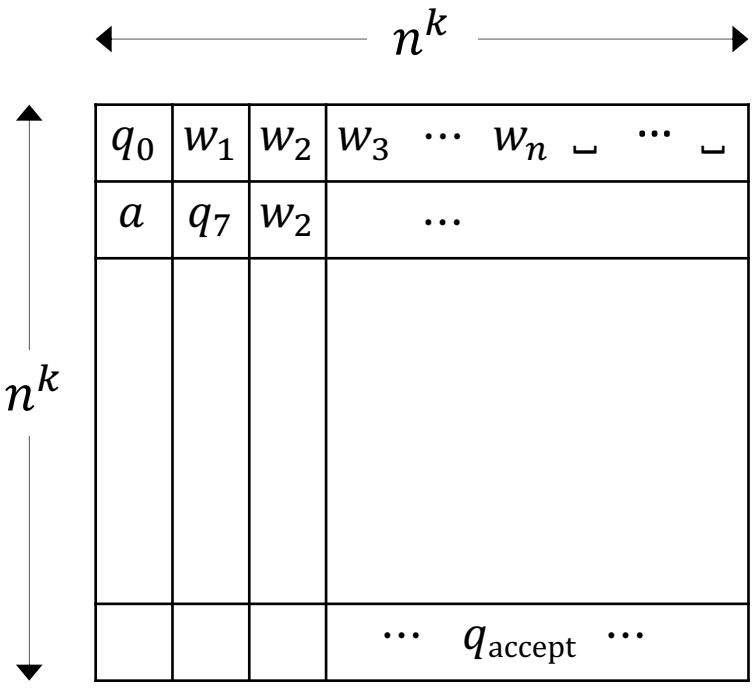
(✓ ✓ ✓)

$$\varphi_{\text{move}} = \bigwedge_{1<i,j<n^k} \left(\bigvee_{\text{Legal}} \left(x_{i,j-1,r} \wedge x_{i,j,s} \wedge x_{i,j+1,t} \wedge x_{i+1,j-1,v} \wedge x_{i+1,j,y} \wedge x_{i+1,j+1,z} \right) \right)$$

r	s	t
v	y	z

Enumerate all legal possibilities (number $\leq |\Gamma \cup Q|^6 = O(1)$) to check whether the neighborhood at i, j is legal

Conclusion: SAT is NP-complete



Summary:

For $A \in$ NP, decided by NTM M,

we gave a reduction f from A to SAT:

$f: \Sigma^* \rightarrow$ Boolean formulas

$f(w) = \langle \varphi_{M,w} \rangle$

$w \in A$ iff $\varphi_{M,w}$ is satisfiable.

$$\varphi_{M,w} = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$$

The size of $\varphi_{M,w}$ is roughly the size of the table:

$$O\left(n^k \times n^k\right) = O\left(n^{2k}\right)$$

Therefore f is computable in polynomial time.

3SAT is NP-complete

Theorem: 3SAT is NP-complete

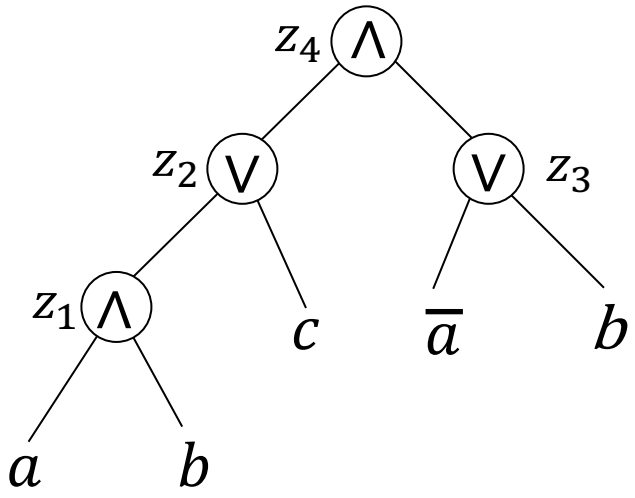
Proof: Show $\text{SAT} \leq_m^P \text{3SAT}$

Give reduction f converting formula φ to 3CNF formula φ',
preserving satisfiability.

Logical equivalence: $(A \to B)$ and $(\overline{A} \vee B)$ $\overline{(A \wedge B)}$ and $(\overline{A} \vee \overline{B})$

Example: Say $\varphi = \big((a \wedge b) \vee c\big) \wedge (\overline{a} \vee b)$

Tree structure for φ:



A	B	$A \to B$	$\overline{A} \vee B$
1	1	1	1
0	1	1	1
1	0	0	0
0	0	1	1

3SAT is NP-complete

a	b	$z = a \wedge b$	
1	1	1	$(a \wedge b) \rightarrow z$
0	1	0	$(\overline{a} \wedge b) \rightarrow \overline{z}$
1	0	0	$(a \wedge \overline{b}) \rightarrow \overline{z}$
0	0	0	$(\overline{a} \wedge \overline{b}) \rightarrow \overline{z}$

Theorem: 3SAT is NP-complete

Proof: Show SAT $\leq_m^P$ 3SAT

Give reduction f converting formula φ to 3CNF formula φ', preserving satisfiability.

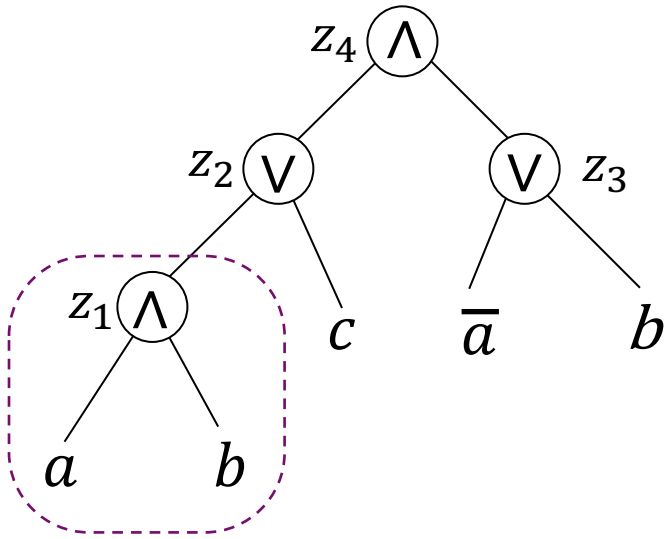
Logical equivalence: $(A \rightarrow B)$ and $(\overline{A} \vee B)$ $\overline{(A \wedge B)}$ and $(\overline{A} \vee \overline{B})$

Example: Say $\varphi = ((a \wedge b) \vee c) \wedge (\overline{a} \vee b)$

$z_1 = a \wedge b$

$$\varphi' = ((a \wedge b) \rightarrow z_1) \wedge ((\overline{a} \wedge b) \rightarrow \overline{z_1}) \wedge ((a \wedge \overline{b}) \rightarrow \overline{z_1}) \wedge ((\overline{a} \wedge \overline{b}) \rightarrow \overline{z_1})$$

Tree structure for φ:



3SAT is NP-complete

a	b	$z = a \vee b$	
1	1	1	$(a \wedge b) \to z$
0	1	1	$(\overline{a} \wedge b) \to z$
1	0	1	$(a \wedge \overline{b}) \to z$
0	0	0	$(\overline{a} \wedge \overline{b}) \to \overline{z}$

Theorem: 3SAT is NP-complete

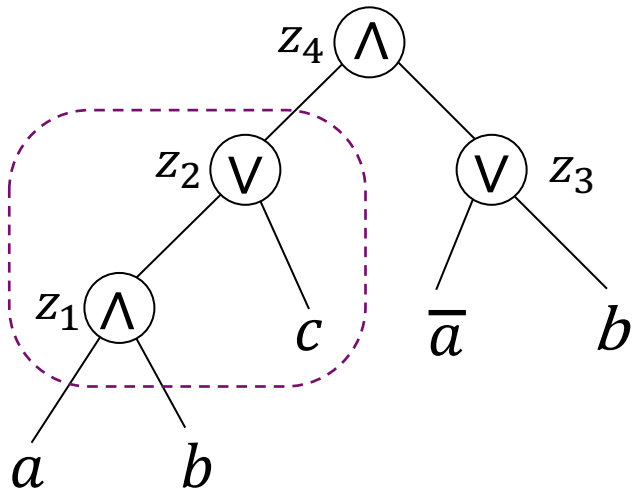
Proof: Show SAT $\leq_m^P$ 3SAT

Give reduction f converting formula φ to 3CNF formula φ', preserving satisfiability.

Logical equivalence: $(A \to B)$ and $(\overline{A} \vee B)$ $\overline{(A \wedge B)}$ and $(\overline{A} \vee \overline{B})$

Example: Say $\varphi = ((a \wedge b) \vee c) \wedge (\overline{a} \vee b)$

Tree structure for φ:



$z_1 = a \wedge b$

$$\varphi' = \big((a \wedge b) \to z_1\big) \wedge \big((\overline{a} \wedge b) \to \overline{z_1}\big) \wedge \big((a \wedge \overline{b}) \to \overline{z_1}\big) \wedge \big((\overline{a} \wedge \overline{b}) \to \overline{z_1}\big)$$

$$\wedge \big((z_1 \wedge c) \to z_2\big) \wedge \big((\overline{z_1} \wedge c) \to z_2\big) \wedge \big((z_1 \wedge \overline{c}) \to z_2\big) \wedge \big((\overline{z_1} \wedge \overline{c}) \to \overline{z_2}\big)$$

$\vdots$ repeat for every z_i

$z_2 = z_1 \vee c$

$\wedge \ (z_4)$

Observe that $((a \wedge b) \to c)$ is logically equivalent to an OR-clause $(\overline{a} \vee \overline{b} \vee c)$

$$((a \wedge b) \to c) = (\overline{(a \wedge b)} \vee c) = ((\overline{a} \vee \overline{b}) \vee c) = (\overline{a} \vee \overline{b} \vee c)$$

3SAT is NP-complete

Theorem: 3SAT is NP-complete

Proof: Show $\text{SAT} \leq_m^P \text{3SAT}$

Mini-Quiz 13.2

If φ has k operations ($\wedge$ and $\vee$), how many clauses has φ'?

(a) k^2 (c) $4k$

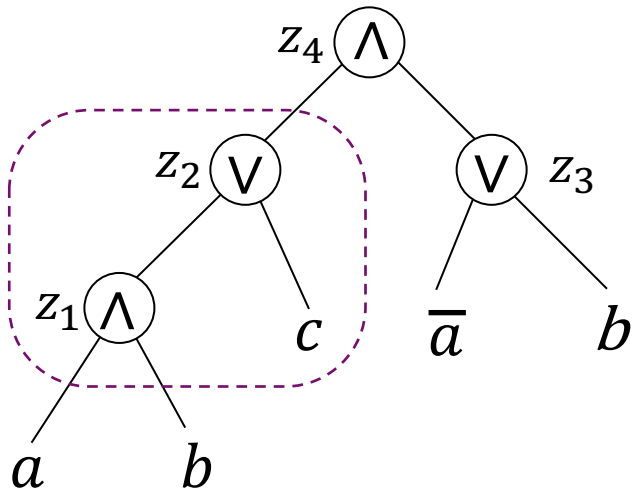
(b) $k+1$ (d) $4k+1$

Give reduction f converting formula φ to 3CNF formula φ', preserving satisfiability.

Logical equivalence: $(A \to B)$ and $(\overline{A} \vee B)$ $\overline{(A \wedge B)}$ and $(\overline{A} \vee \overline{B})$

Example: Say $\varphi = \big((a \wedge b) \vee c\big) \wedge (\overline{a} \vee b)$

Tree structure for φ:



$z_1 = a \wedge b$

$$\varphi' = \big((a \wedge b) \to z_1\big) \wedge \big((\overline{a} \wedge b) \to \overline{z_1}\big) \wedge \big((a \wedge \overline{b}) \to \overline{z_1}\big) \wedge \big((\overline{a} \wedge \overline{b}) \to \overline{z_1}\big)$$

$$\wedge \big((z_1 \wedge c) \to z_2\big) \wedge \big((\overline{z_1} \wedge c) \to z_2\big) \wedge \big((z_1 \wedge \overline{c}) \to z_2\big) \wedge \big((\overline{z_1} \wedge \overline{c}) \to \overline{z_2}\big)$$

$\vdots$ repeat for every z_i

$z_2 = z_1 \vee c$

$\wedge \ (z_4)$

Observe that $\big((a \wedge b) \to c\big)$ is logically equivalent to an OR-clause $(\overline{a} \vee \overline{b} \vee c)$

$$\big((a \wedge b) \to c\big) = \big(\overline{(a \wedge b)} \vee c\big) = \big((\overline{a} \vee \overline{b}) \vee c\big) = (\overline{a} \vee \overline{b} \vee c)$$

Quick review of today

1. SAT is NP-complete
2. $3\mathrm{SAT}$ is NP-complete

These slides are adapted from Michael Sipser, *18.404J Theory of Computation*, Fall 2020. Massachusetts Institute of Technology:

- Source: MIT OpenCourseWare https://ocw.mit.edu

Except for separately identified third-party material, this adapted presentation is licensed under CC BY-NC-SA 4.0.