Strategy

- Once we have one NP-complete problem, we can generate more by poly-time reduction.
- Getting the first one requires some work.

Cook-Levin Theorem

We will show that 3SAT is NP complete.

We must show that every problem in NP reduces to 3SAT

We will need some more definitions . . .

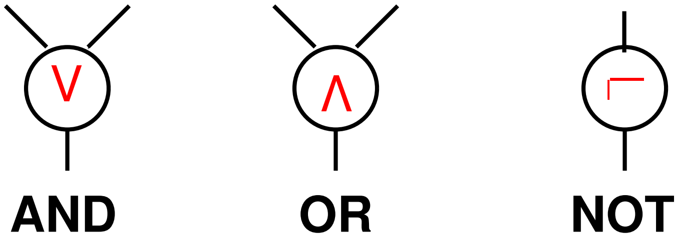
Boolean Circuits

Here is a theoretical analogue of logic circuits.

A **Boolean circuit** is a collection of

- gates
- inputs
- wires
- no cycles

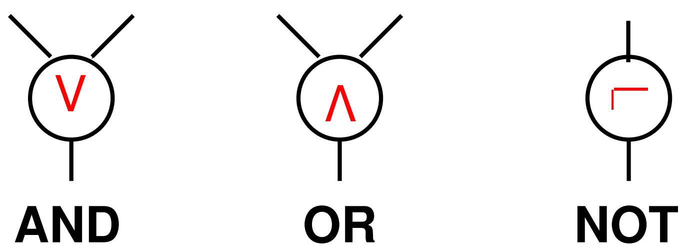
Boolean Circuits



Gates come in three forms

- AND-gates
- OR-gates
- NOT-gates

Boolean Circuits



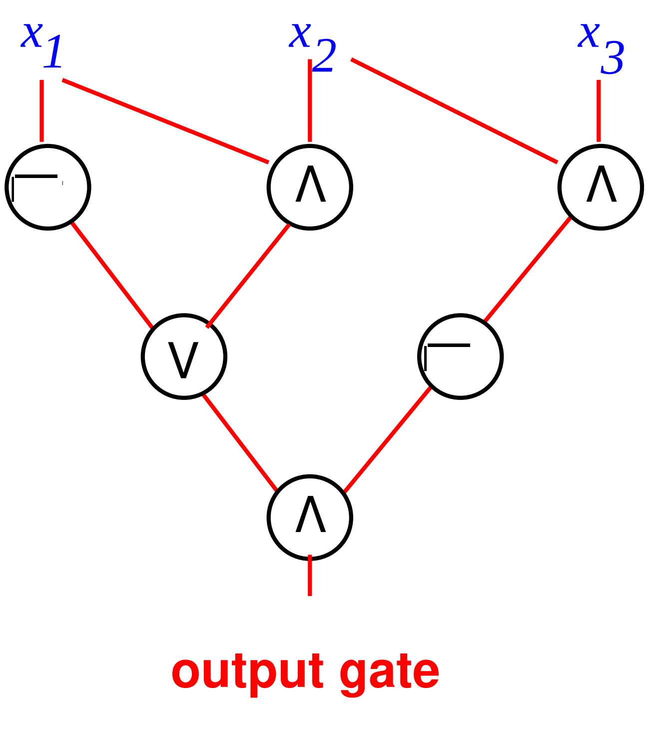
Wires carry 0 or 1.

Gates compute Boolean function

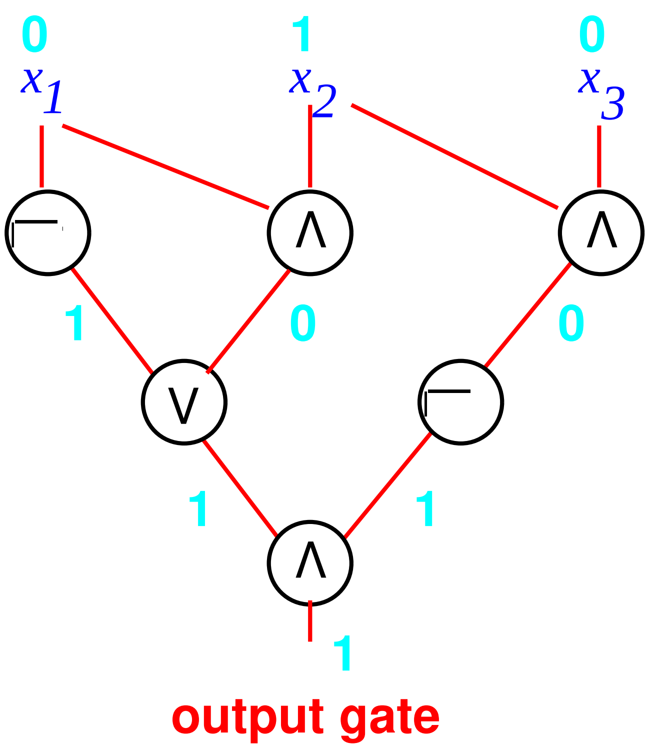
- AND-gates: 1 iff all input wires 1
- OR-gates: 1 iff any input wire 1
- NOT-gates opposite if its input

There is a distinguished *output gate*.

Boolean Circuits



Boolean Circuits



Boolean Circuits

A circuit is *satisfiable* if there is some input setting that produces a 1.

The circuit-satisfiability problem tests whether a circuit is satisfiable.

CIRCUIT-SAT $= \{\langle C\rangle | C$ is a satisfiable Boolean circuit $\}$

Boolean Circuits

Claim: CIRCUIT-SAT is in NP

Pop Quiz: Why?

We will show that circuits can simulate Turing Machines!

Boolean Circuits

Let $t : \mathcal{N} \longrightarrow \mathcal{N}$ be a function, where $t(n) \geq n$.

Let M be a TM that

- takes inputs over alphabet $\{0, 1\}$
- runs in time $t(n)$

Claim: Each n has a Boolean circuit with $O(t^2(n))$ gates that outputs 1 on those inputs of length n accepted by M.

Proof Idea

We design C_n to simulate M on inputs of length n.

Gates of C_n are

- organized in rows
- one for each of the $t(n)$ steps in M's computation
- each row represents configuraiton of M at corresponding step
- each row wired into previous row

Helpful to make modification to M

- accepts or rejects with head at leftmost position

Review: Configurations

Configuration:

$$1011q_70111$$

means:

- state is q_7
- LHS of tape is 1011
- RHS of tape is 0111
- head is on RHS 0

Proof

Let M be TM $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$.

Definition: The **tableau** for M on w is a $t(n)$ by $t(n)$ table.

- rows are configurations of M
- top row is start configuration of M on w
- i^{th} row is configuration at i^{th} step.

Proof

Question: Why $t(n)$ by $t(n)$?

- How much tape can M read in time $t(m)$?
- How many configuations can M assume in time $t(m)$?

Frivolous Aside

What is the plural of *tableau*?

	standard	pedantic/correct	wierd/mistaken
tableau	tableaux		tableaus
millenium	milleniums	millenia	
criterion	criteria		criterions
lemma	lemmas	lemmata	lemmae
schema	schemata		
octopus	octopuses	octopoi	octopi

Singular sensations:

biscotti	biscotti	biscotto	cookie
brocolli	brocolli	brocollo	brocollus
media	media	medium	

Proof

Each entry in the tableau contains either

- a tape symbol (in Γ)
- a state (in Q)

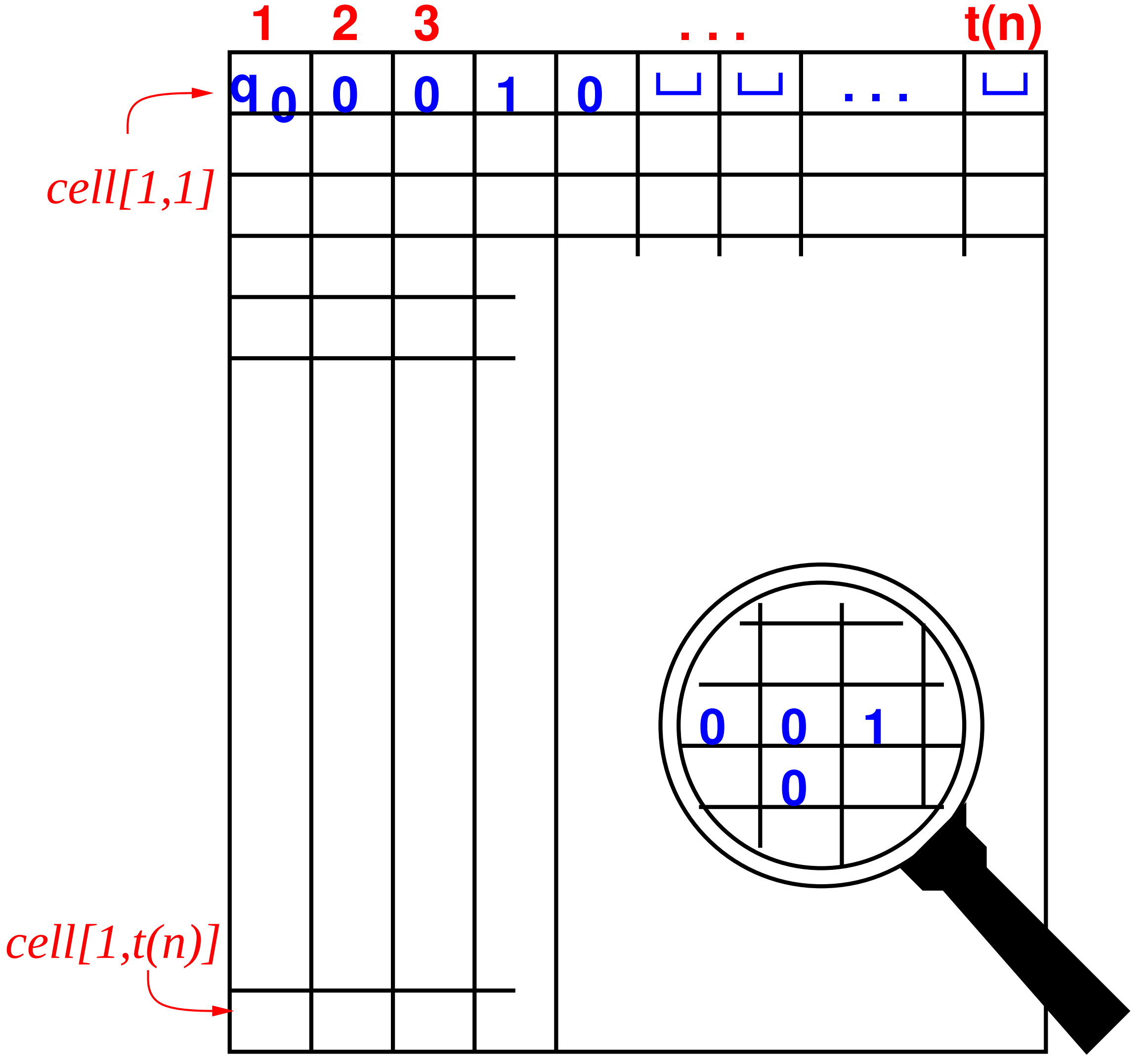
Call entry at row i and column j *cell*$[i, j]$.

- *cell*$[1, 1], \ldots,$ *cell*$[1, t(n)]$ contains starting configuration
- so *cell*$[1, 1] = q_0$, the starting state.

Recall that M is modified so that

- *cell*$[t(n), 1]$ is either q_a or q_r.

Proof



Proof

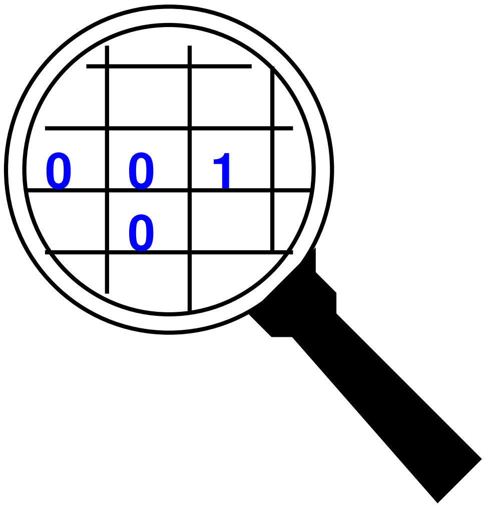
Content of cell is determined by neighbors in preceding row.

Given

- *cell*$[i-1, j-1]$
- *cell*$[i, j-1]$
- *cell*$[i+1, j-1]$

we can determine *cell*$[i, j]$ from M's transition function.

Proof



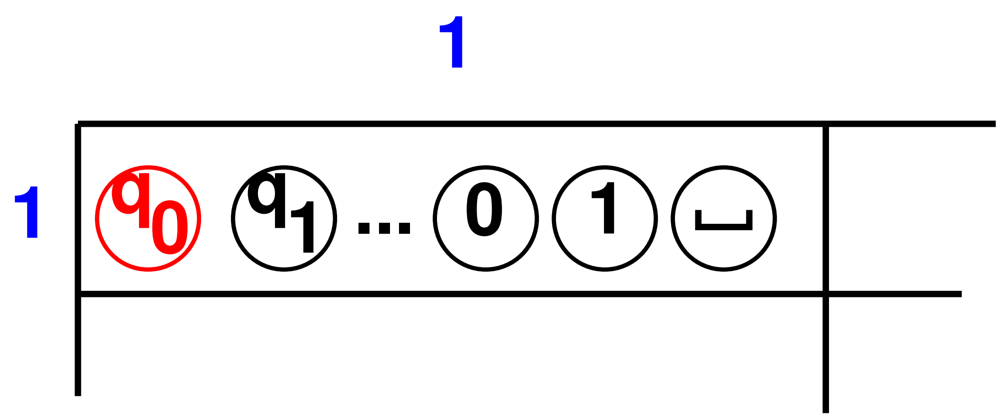
- three top symbols tape symbols
- nowhere near the head
- so middle symbol unchanged on next row

Constructing the Circuit

Circuit C_n

- has several gates for each cell in the tableau
- compute value at a cell from the 3 cells that affect it.

Constructing the Circuit



To make the description easier, we

- add "lights" to show the output
- illustrative purposes only

Let $k =$ number of elements in $\Gamma \cup Q$.

- each cell has k possible lights
- grand total of $k \cdot t^2(n)$ lights
- only one light can be lit,
- that light indicates contents of cell.

Constructing the Circuit

Model each light as *light*$[i, j, s]$, where

- $1 \leq i, j \leq t(n)$
- $s \in \Gamma$
- one and only one light on in each cell
- if *light*$[i, j, s]$ is on, then *cell*$[i, j] = s$.

Constructing the Circuit

Pick *light*$[i, j, s]$ in *cell*$[i, j]$.

This light should be on iff *cell*$[i, j] = s$.

Only three cells can affect this cell:

- *cell*$[i - 1, j - 1]$,
- *cell*$[i, j - 1]$, and
- *cell*$[i + 1, j - 1]$.

How they affect the cell determined by transition function δ.

Constructing the Circuit

Suppose that if

- *cell*$[i-1, j-1] = a_1$,
- *cell*$[i, j-1] = b_1$, and
- *cell*$[i+1, j-1] = c_1$.

then *cell*$[i, j] = 2$, according to δ.

Constructing the Circuit

Suppose that if

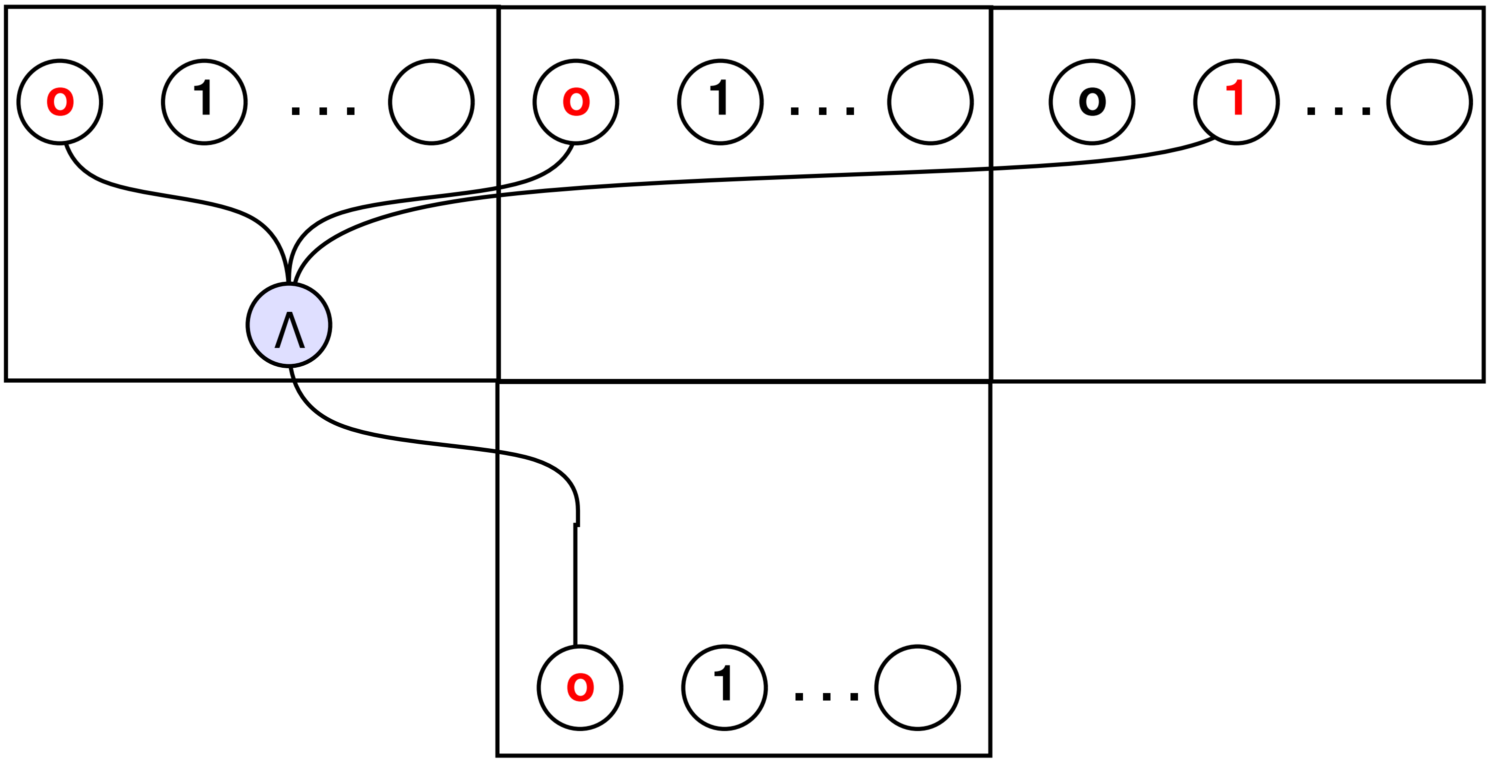
- *cell*$[i-1, j-1] = a_1$,
- *cell*$[i, j-1] = b_1$, and
- *cell*$[i+1, j-1] = c_1$.

then *cell*$[i, j] = s$, according to δ.

Wire the circuit so that

light$[i, j, s] =$ *light*$[i-1, j-1, a_1] \wedge$ *light*$[i-1, j-1, b_1] \wedge$ *light*$[i-1, j-1, c_1]$

Constructing the Circuit



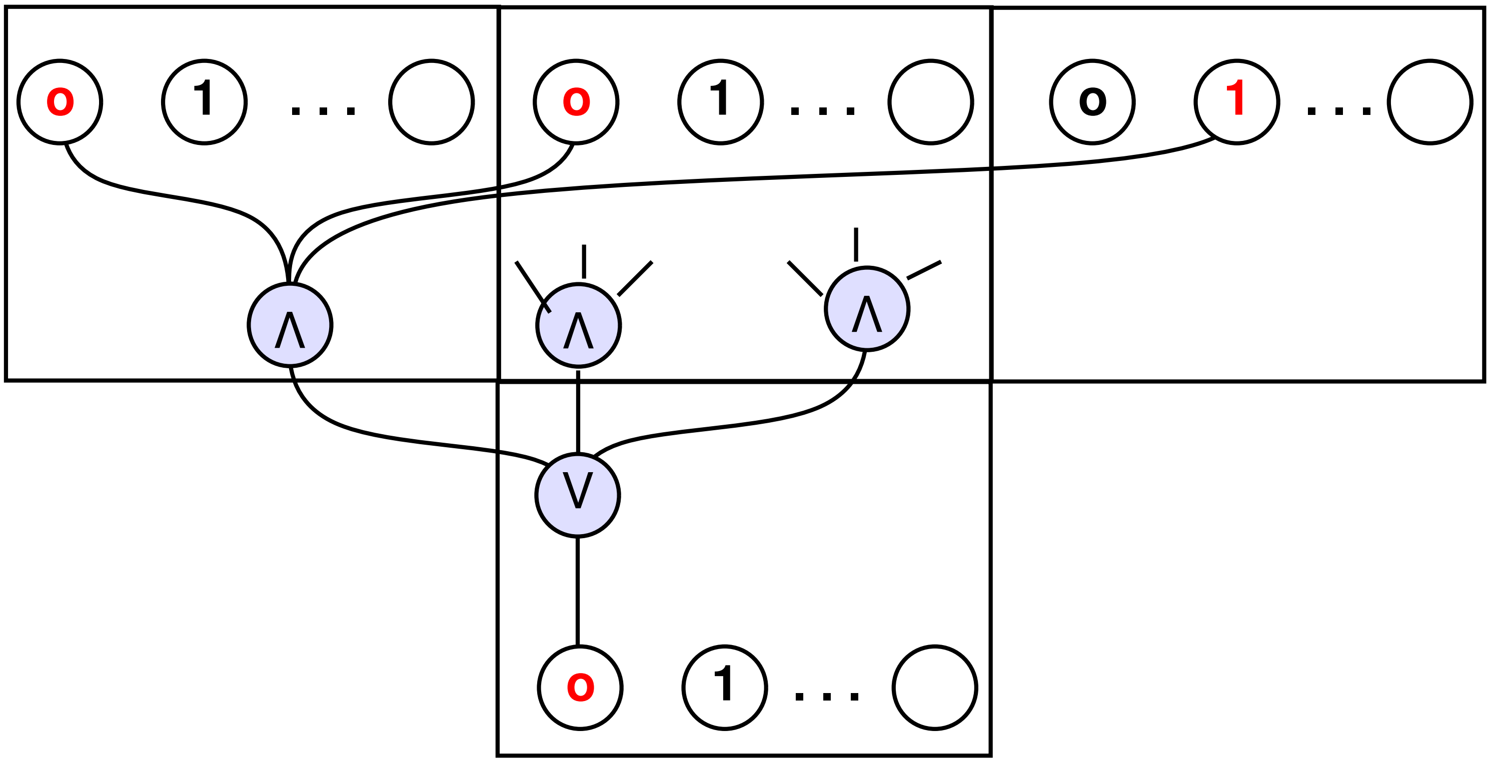
Constructing the Circuit

In general, more than one setting $(a_1, b_1, c_1), \ldots, (a_k, b_k, c_k)$ of *cell*$[i-1, j-1]$.*cell*$[i, j-1]$, *cell*$[i+1, j-1]$ can yield s.

Wire the circuit so that

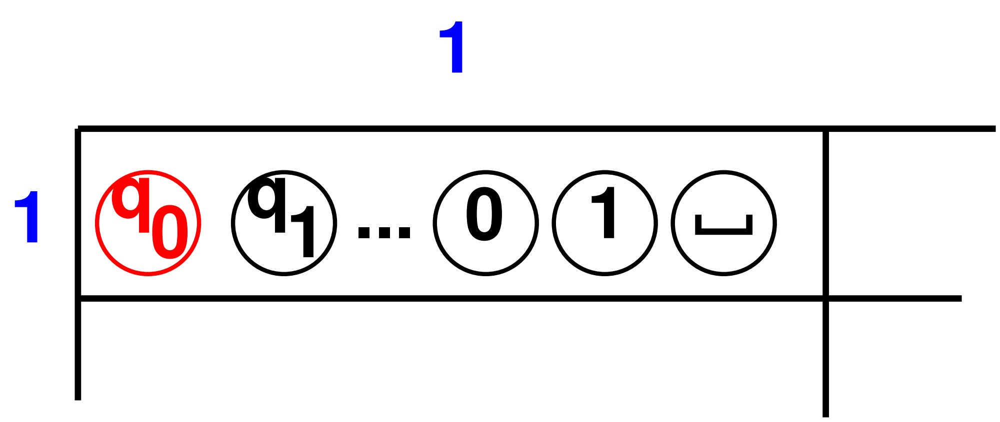
- a_i, b_i, c_i connected by $\wedge$ gates
- different settings connected by $\vee$ gates

Constructing the Circuit



Constructing the Circuit

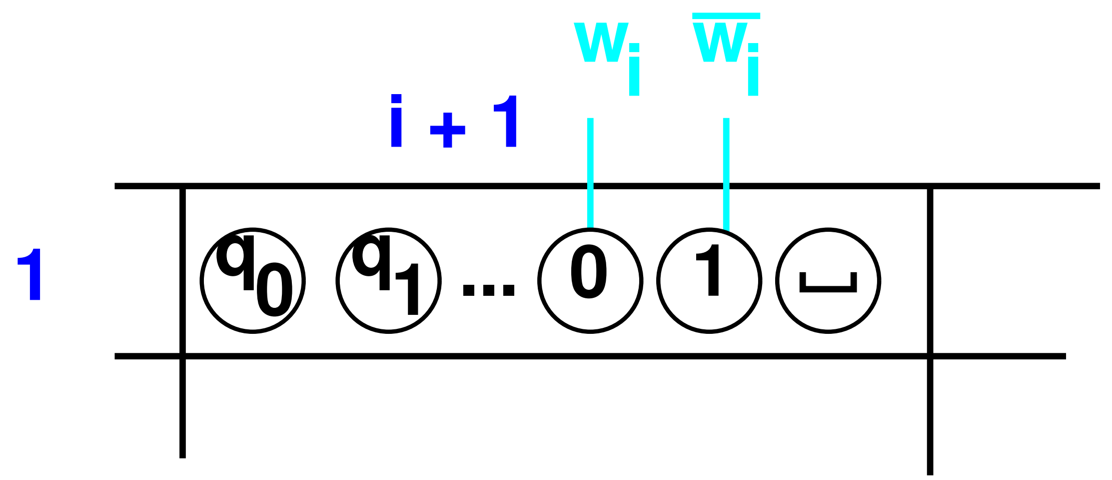
First row hardwired to contain start configuration



light$[1, 1, q_0]$ is initially on!

Constructing the Circuit

First row hardwired to contain start configuration

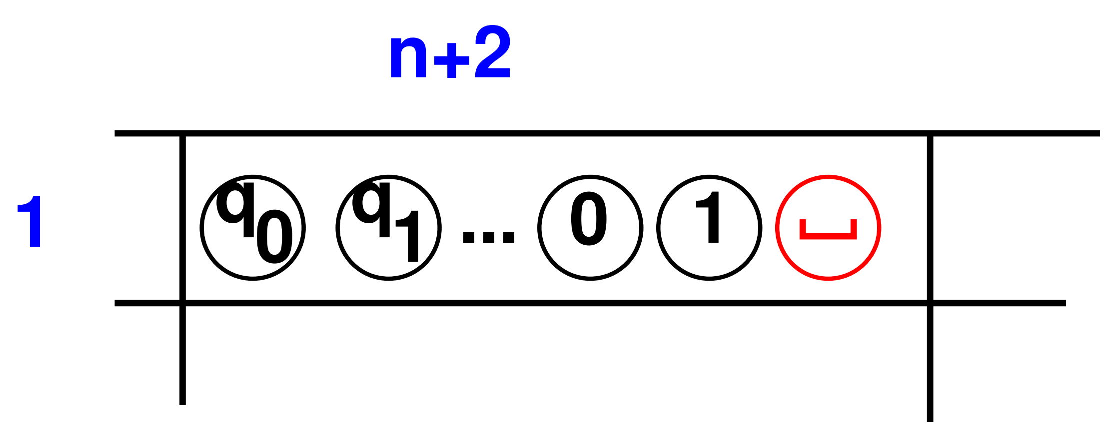


- *light*$[1, 2, 1], \ldots,$ *light*$[1, n+1, 1]$ connected to $w_1, \ldots, w_n$
- *light*$[1, 2, 0], \ldots,$ *light*$[1, n+1, 0]$ connected by NOT gates to $w_1, \ldots, w_n$.

q

Constructing the Circuit

First row hardwired to contain start configuration



light$[1, n+2, \sqcup], \ldots,$ *light*$[1, t(n), \sqcup]$ all on.

Constructing the Circuit

We have constructed a circuit that simulates M through its $t(n)^{th}$ step.

Need one gate to be the output gate of the circuit

M accepts w iff it is

- in q_0, and
- at left-hand end of tape

Choose *light*$[t(n), q, q_a]$.

This completes the proof that M can be simulted by C_n.

NP-Completeness

Theorem: CIRCUIT-SAT is NP-complete.

CIRCUIT-SAT is clearly in NP Why?.

We must show that any language A in NP is reducible to CIRCUIT-SAT.

We must give a polynomial-time reduction, mapping strings to circuits,

$$f(w) = C$$

so tht

$$w \in A \Leftrightarrow C \text{ is satisfiable.}$$

NP-Completeness

Let V be a polynomial-time verifier for A.

- input $\langle x, c \rangle$
- c is certificate for x

Now

- construct circuit simulating V.
- fill in x inputs to circuit with w
- only remaining inputs are for c

Output this circuit, and call it C.

NP-Completeness

Output this circuit, and call it C.

- if C is satisfiable, a certificate exists, and $w \in A$.
- if $w \in A$, certificate exists, and C is satisfiable.

Check that circuit construction is poly time.

- running time of verifier is n^k.
- size of circuit is $O(n^{2k})$
- circuit is simple, and running time of reduction is $O(n^{2k})$

3SAT

Theorem: 3SAT is NP-complete.

3SAT is obviously in NP.

We give poly-time reduction of CIRCUIT-SAT to 3SAT.

Let C be a circuit containing inputs $x_1, \ldots, x_\ell$ and gates $g_1, \ldots, g_m$.

We build a boolean formula ϕ with variables $x_1, \ldots, x_\ell, g_1, \ldots, g_m$.

- the x_1 correspond to input wires
- the g_i correspond to gate output wires
- relable variables $w_1, \ldots, w_{\ell+m}$

3SAT

For each NOT gate in C with input w_i and output w_j, add clauses

$$(w_i \vee w_j) \wedge (\overline{w_i} \vee \overline{w_j})$$

Both clauses satisfied when assignment to w_i and w_j mirrors NOT gate.

3SAT

For each AND gate in C with inputs w_i, w_j and output w_k, add clauses

$$(w_i \vee w_j \vee \overline{w_k}) \wedge (w_i \vee \overline{w_j} \vee \overline{w_k}) \wedge (\overline{w_i} \vee w_j \vee \overline{w_k}) \wedge (\overline{w_i} \vee \overline{w_j} \vee w_k)$$

All clauses satisfied when assignment to w_i, w_j and w_k mirrors correcly functioning AND gate.

3SAT

For each OR gate becomes

$$(w_i \vee w_j \vee \overline{w_k}) \wedge (w_i \vee \overline{w_j} \vee w_k) \wedge (\overline{w_i} \vee w_j \vee w_k) \wedge (\overline{w_i} \vee \overline{w_j} \vee w_k)$$

All clauses satisfied when assignment to w_i, w_j and w_k mirrors correcly functioning OR gate.

3SAT

All clauses have no more than 3 literals.

If they have fewer, then repeat a literal.

- a satisfying assignment for C yields a satisfying assignment for ϕ by assigning the g_i variables according to C.
- a satisfying assignment for ϕ yields a satisfying assignment for C because it describes C's computation with output 1.
- reduction is polynomial time.