CS20a: NP completeness

- Cook's theorem
  - *SAT is an NP-complete problem*

NP-complete definition

- A problem is in NP if it can be solved by a nondeterministic algorithm in poly-time
- A problem Q is NP-hard if every problem in NP can be reduced to Q
- A problem is NP-complete if it is in NP, and it is NP-hard

Related properties

- A problem Q is NP-complete if it is in NP, and some NP-hard problem R can be reduced to Q
  - *For example, assume SAT is NP-complete*
    - *SAT reduces to graph coloring (showed this last time)*
    - *Graph-coloring is in NP (guess a coloring, and check)*
    - *So graph coloring is also NP-complete*
- General method to show a problem Q is NP-complete:
  - *Show Q is in NP (give an algorithm)*
  - *Choose some NP-complete problem R, and reduce R to Q in deterministic poly-time*
- WARNING
  - *Do <u>not</u> get this backward (reducing Q to R)!!!*

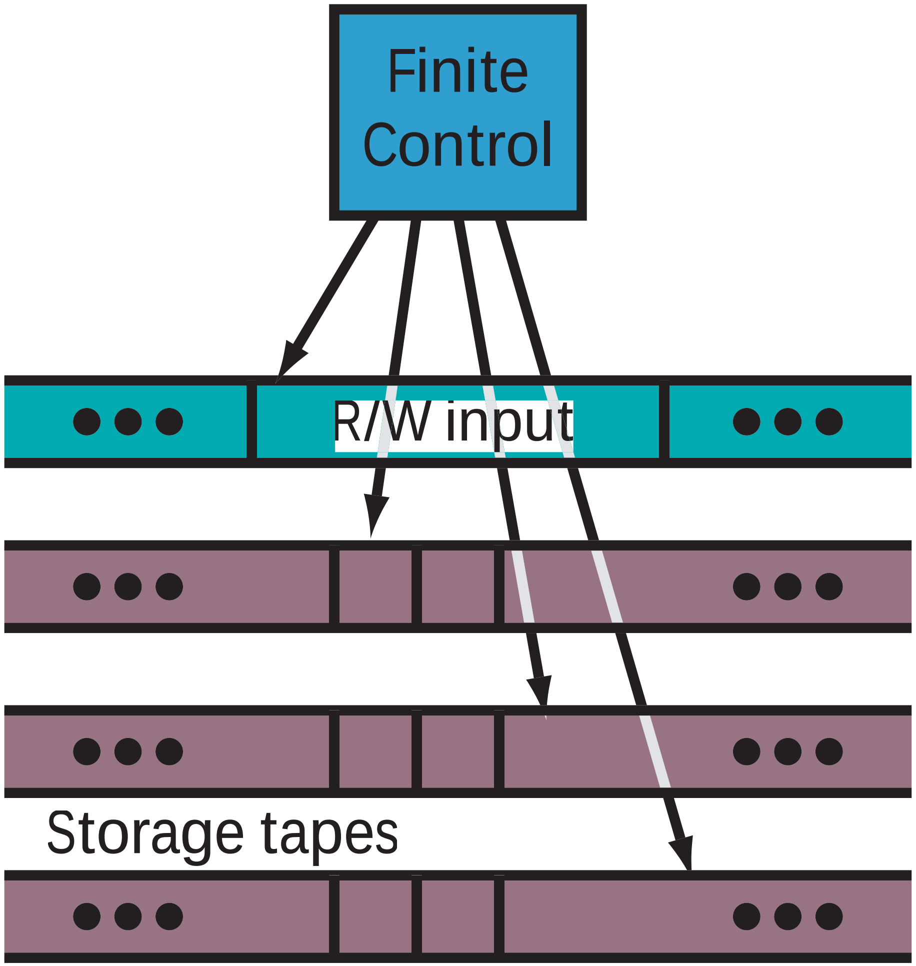
Summary so far

- We have seen a lot of problems in NP
  - *k-CNF SAT*
  - *Graph clique finding*
  - *Graph coloring*
  - *Knapsack, subset-sum, partition, bin-packing*
- We have seen several reductions
  - *k-CNF SAT <p Clique*
  - *k-CNF SAT <p Graph-coloring*
  - *Knapsack =p Subset-sub =p Partition*
- We have not seen an NP-hard problem

Cook’s theorem

- Show that SAT is NP-hard
  - *We have to show that every problem in NP reduces to SAT*
  - *How???*

Time-bounded TMs



- All tapes are 2-way infinite
- M is DTIME(T(n)) if
  - *M is deterministic*
  - *For any input of length n, M takes at most T(n) steps*
- M is NTIME(T(n))
  - *Nondeterministic case*

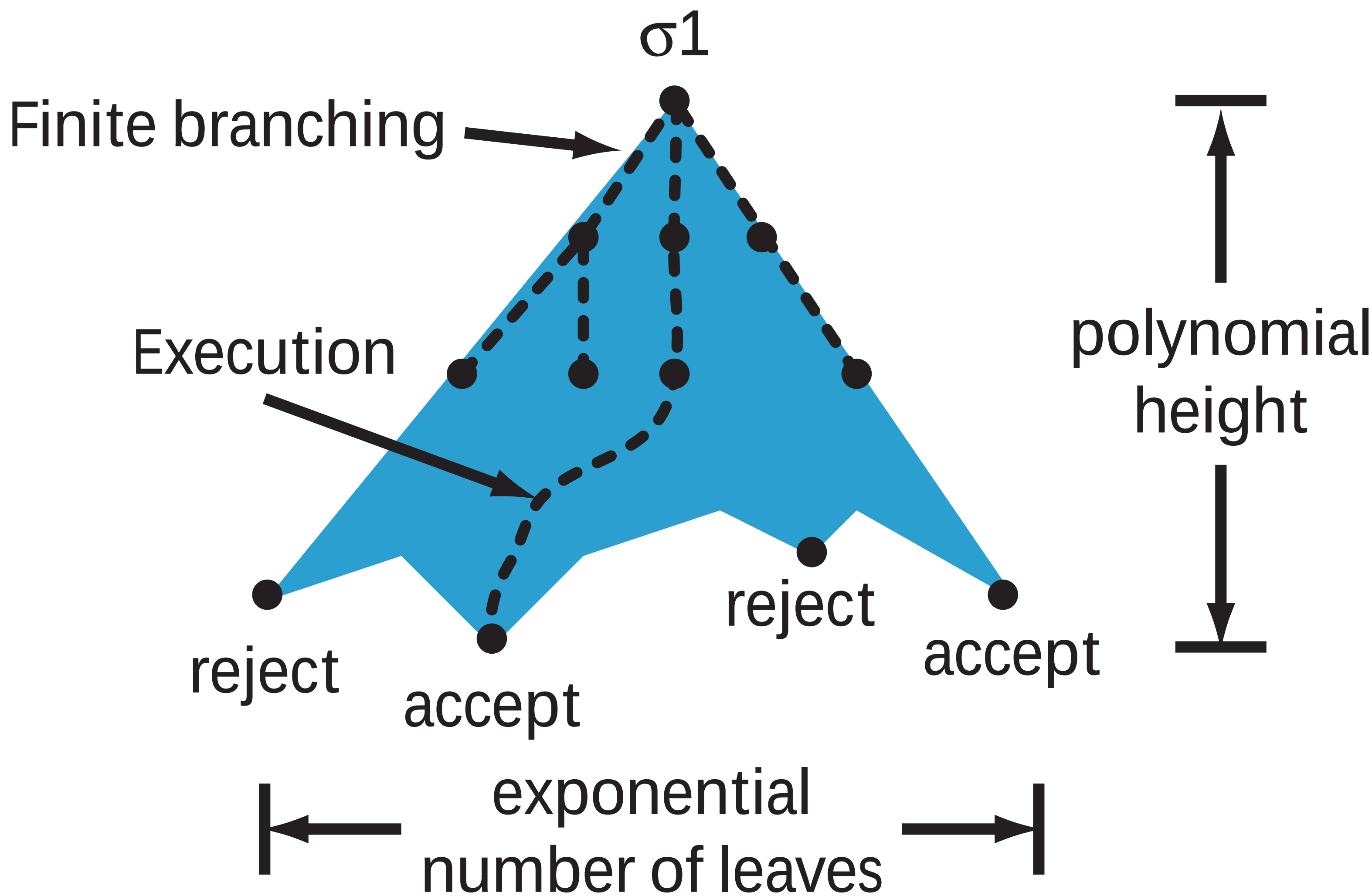
Cook's theorem outline

- Consider an arbitrary problem R in NP
- Then R is accepted by a TM in NTIME($O(n^c)$)
- By definition, R accepts iff there is an accepting execution
- Build a formula that is satisfiable iff there is an accepting execution

Executions

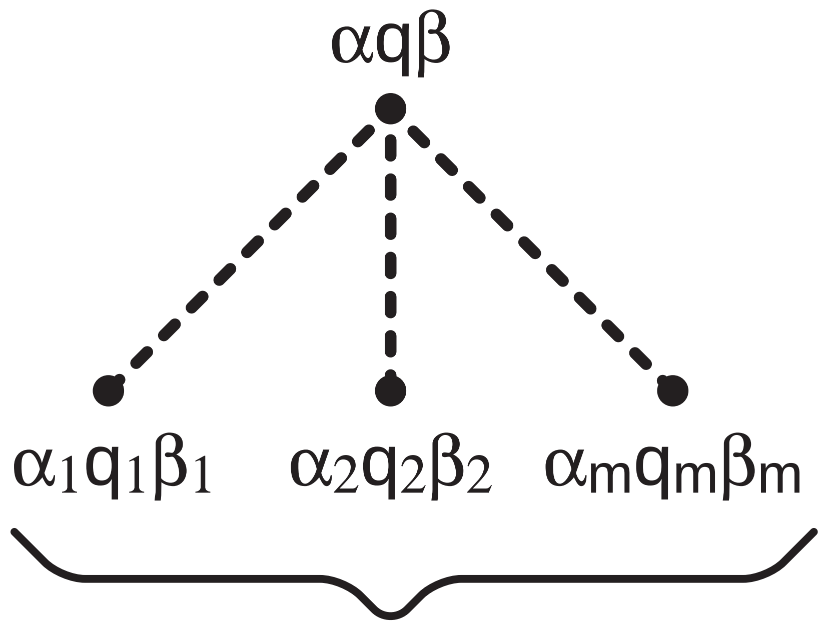
- An *instantaneous description* ID (σ) is $\alpha_1 q \alpha_2$, where
  - q is the current state of the TM,
  - $\alpha_1 \alpha_2 \in \Gamma^*$ is the contents of the tape (to the last non-blank symbol)
  - The current symbol is the first symbol of α_2
- A PTIME *execution* is
  - A sequence $\sigma_1 \sigma_2 \cdots \sigma_m$,
  - where σ_1 has the form $q_0 \alpha$ and $|\alpha| = n$
  - and $\sigma_i \to \sigma_{i+1}$
  - and m is $O(n^c)$

NP executions (configuration trees)



NP configuration tree node

Instantaneous description



Nondeterministic Choices

there are a finite number

determined by the finite control

Plan

- Build formulas for
  - *The initial configuration*
  - *The transitions,*
  - *The final configuration*

Cook's theorem

Theorem If $R \in NP$ then $R \leq_m^p SAT$

Preparation

- Let M be the TM for R
- Assume the depth of the configuration tree is at most $N = |x|^k$ for some constant k
- Assume that M uses a single tape, delimited at the left by a special symbol $\vdash$
- Also, assume when M accepts, it erases its input, and moves all the way left
- Note, M can scan at most N tape cells

Boolean variables

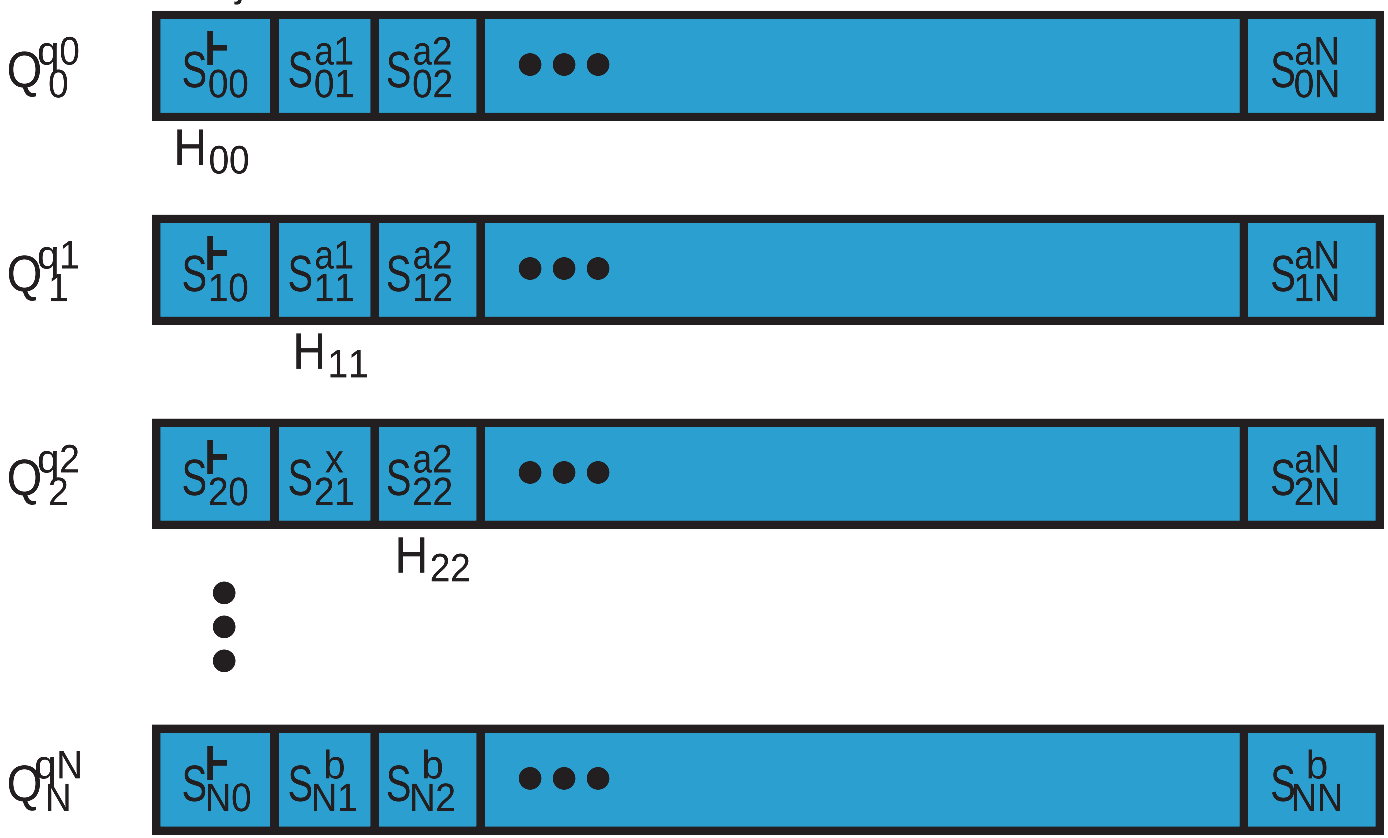
- Q_i^q: at time i, the machine is in state q
- H_{ij}: at time i, the tape head is at position j
- S_{ij}^a: at time i, the symbol in cell j is a

Boolean variables

Q_i^q : at time i, the machine is in state q

S_{ij}^a : at time i, tape cell j contains symbol a

H_{ij} : at time i, the tape head is at cell j



Formula for initial state

- The machine starts in state s with left marker $\vdash$
- The input x fills the first $|x|$ tape cells
- The rest of the tape is filled with blanks

$$Q_0^s \wedge H_{00} \wedge S_{00}^{\vdash} \wedge \bigwedge_{1 \le j \le |x|} S_{0j}^{x_j} \wedge \bigwedge_{|x|+1 \le j \le N} S_{0j}^{\overline{b}}$$

Formula for final state

- When M accepts, it first erases the input, moves left to the $\vdash$ marker, and enters final state t
- It does not move or change state after that

$$Q_N^t \wedge H_{N0} \wedge S_{N0}^{\vdash} \wedge \bigwedge_{1 \le j \le N} S_{0j}^{\overline{b}}$$

State Constraint

- At any time, the machine is in exactly one state

$$\bigwedge_{0 \le i \le N} \left(\bigvee_{q \in Q} Q_i^q \right) \wedge \bigwedge_{0 \le i \le N} \bigwedge_{p,q \in Q \wedge p \ne q} \left(\neg Q_i^p \vee \neg Q_i^q \right)$$

Symbol constraint

- At any time, each tape cell contains exactly one symbol.

$$\bigwedge_{0 \le i,j \le N} \left(\bigvee_{a \in \Sigma} S_{ij}^{a} \right) \wedge \bigwedge_{0 \le i,j \le N} \; \bigwedge_{a,b \in \Sigma \wedge a \ne b} \left(\neg S_{ij}^{a} \vee \neg S_{ij}^{b} \right)$$

Head position constraint

- At any time, the machine is scanning exactly one cell.

$$\bigwedge_{0 \le i \le N} \left(\bigvee_{0 \le j \le N} H_{ij} \right) \wedge \bigwedge_{0 \le i \le N} \bigwedge_{0 \le j < k \le N} \left(\neg H_{ij} \vee \neg H_{ik} \right)$$

Expressing the transition relation

- A deterministic transition function δ has the form

$$\delta : (Q \times \Sigma) \rightarrow (Q \times \Sigma \times \{-1, 0, 1\})$$

- For nondeterministic machines, δ is a relation
- If $((p, a), (q, b, d)) \in \delta$ then
  - When M is in state p, reading symbol a
  - Then it may:
    * Print symbol b
    * Move to state q
    * Move the tape head in direction d

Transition formula

- If M is in state p, reading symbol a
- It must move to some $((p,a),(q,b,d)) \in \delta$

$$\bigwedge_{0 \le i,j \le N, a \in \Sigma, p \in Q}$$

$$Q_i^p \wedge H_{ij} \wedge S_{ij}^a \Rightarrow \bigvee_{((p,a),(q,b,d)) \in \delta} \left(Q_{i+1}^q \wedge H_{i+1,j+d} \wedge S_{i+1,j}^b \right)$$

Preserving the tape state

- If the tape head is not at position i, then the symbol at position i is unchanged

$$\bigwedge_{0 \le i,j \le N, a \in \Sigma} \left(S_{ij}^{a} \wedge \neg H_{ij} \Rightarrow S_{i+1,j}^{a} \right)$$

Cleaning up

- All the formulas are in CNF, except for the transition formulas
  - *Convert the transition formulas to CNF the normal way*
  - *Note that delta is finite, so we get at most a polynomial blowup in size*
- The formulas are poly-size
- The formulas can be constructed in poly-time
- Every satisfying assignment gives rise to an accepting computation