

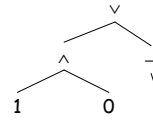
CS151 Complexity Theory

Lecture 2
March 29, 2007

Time and Space

A motivating question:

- Boolean formula with n nodes
- evaluate using $O(\log n)$ space?



- depth-first traversal requires storing intermediate values

- idea: short-circuit ANDs and ORs when possible

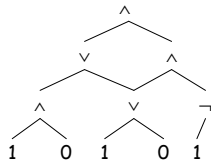
March 29, 2007

CS151 Lecture 2

2

Time and Space

- Can we evaluate an n node Boolean circuit using $O(\log n)$ space?



March 29, 2007

CS151 Lecture 2

3

Time and Space

- Recall:
 - $\text{TIME}(f(n))$, $\text{SPACE}(f(n))$
- Questions:
 - how are these classes related to each other?
 - how do we define **robust** time and space classes?
 - what problems are contained in these classes? complete for these classes?

March 29, 2007

CS151 Lecture 2

4

Outline

- Why big-oh? Linear Speedup Theorem
- Hierarchy Theorems
- Robust Time and Space Classes
- Relationships between classes
- Some complete problems

March 29, 2007

CS151 Lecture 2

5

Linear Speedup

Theorem: Suppose TM M decides language L in time $f(n)$. Then for any $\epsilon > 0$, there exists TM M' that decides L in time

$$\epsilon f(n) + n + 2.$$

- Proof:
 - simple idea: increase "word length"
 - M' will have
 - one more tape than M
 - m -tuples of symbols of M
 - many more states $\Sigma_{\text{new}} = \Sigma_{\text{old}} \cup \Sigma_{\text{old}}^m$

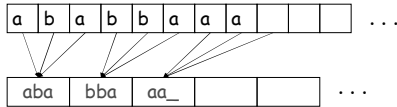
March 29, 2007

CS151 Lecture 2

6

Linear Speedup

- part 1: compress input onto fresh tape



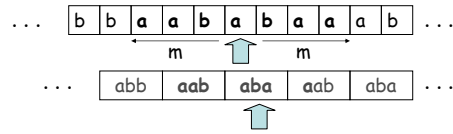
March 29, 2007

CS151 Lecture 2

7

Linear Speedup

- part 2: simulate M, m steps at a time



- 4 (L,R,R,L) steps to read relevant symbols, "remember" in state
- 2 (L,R or R,L) to make M's changes

March 29, 2007

CS151 Lecture 2

8

Linear Speedup

- accounting:
 - part 1 (copying): $n + 2$ steps
 - part 2 (simulation): $6(f(n)/m)$
 - set $m = 6/\epsilon$
 - total: $\epsilon f(n) + n + 2$

Theorem: Suppose TM M decides language L in space $f(n)$. Then for any $\epsilon > 0$, there exists TM M' that decides L in space $\epsilon f(n) + 2$.

- Proof: same.

March 29, 2007

CS151 Lecture 2

9

Time and Space

- Moral: big-oh notation necessary given our model of computation
 - Recall: $f(n) = O(g(n))$ if there exists c such that $f(n) \leq c g(n)$ for all sufficiently large n .
 - TM model incapable of making distinctions between time and space usage that differs by a constant.
- In general: interested in coarse distinctions not affected by model
 - e.g. simulation of k -string TM running in time $f(n)$ by single-string TM running in time $O(f(n)^2)$

March 29, 2007

CS151 Lecture 2

10

Hierarchy Theorems

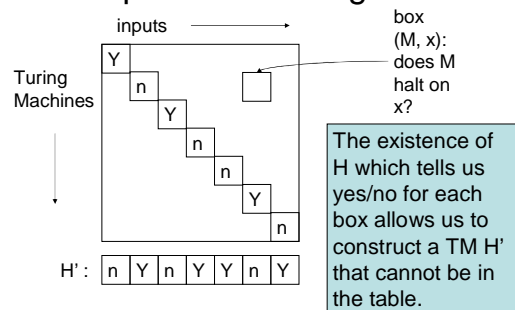
- Does **genuinely** more time permit us to decide new languages?
- how can we construct a language L that is **not** in **TIME(f(n))**...
 - idea: same as "HALT undecidable" diagonalization and simulation

March 29, 2007

CS151 Lecture 2

11

Recall proof for Halting Problem



March 29, 2007

CS151 Lecture 2

12

Time Hierarchy Theorem

Turing Machines

↓

inputs

Y					
	n				
		Y			
			n		
				Y	
					n

D:

n	Y	n	Y	Y	n	Y
---	---	---	---	---	---	---

box (M, x): does M accept x in time f(n)?

- TM SIM tells us yes/no for each box in time $g(n)$
- rows include all of $\text{TIME}(f(n))$
- construct TM D running in time $g(2n)$ that is not in table

March 29, 2007
CS151 Lecture 2
13

Time Hierarchy Theorem

Theorem (Time Hierarchy Theorem): For every proper complexity function $f(n) \geq n$:
 $\text{TIME}(f(n)) \subsetneq \text{TIME}(f(2n)^3)$.

- more on “proper complexity functions” later...

March 29, 2007
CS151 Lecture 2
14

Proof of Time Hierarchy Theorem

- Proof:
 - SIM is TM deciding language $\{ \langle M, x \rangle : M \text{ accepts } x \text{ in } \leq f(|x|) \text{ steps} \}$
 - Claim: SIM runs in time $g(n) = f(n)^3$.
 - define new TM D: on input $\langle M \rangle$
 - if SIM accepts $\langle M, M \rangle$, reject
 - if SIM rejects $\langle M, M \rangle$, accept
 - D runs in time $g(2n)$

March 29, 2007
CS151 Lecture 2
15

Proof of Time Hierarchy Theorem

- Proof (continued):
 - suppose M in $\text{TIME}(f(n))$ decides L(D)
 - $M(\langle M \rangle) = \text{SIM}(\langle M, M \rangle) \neq D(\langle M \rangle)$
 - but $M(\langle M \rangle) = D(\langle M \rangle)$
 - contradiction.

March 29, 2007
CS151 Lecture 2
16

Proof of Time Hierarchy Theorem

- Claim: there is a TM SIM that decides $\{ \langle M, x \rangle : M \text{ accepts } x \text{ in } \leq f(|x|) \text{ steps} \}$ and runs in time $g(n) = f(n)^3$.
- Proof sketch: SIM has 4 work tapes
 - contents and “virtual head” positions for M’s tapes
 - M’s transition function and state
 - $f(|x|)$ “+”s used as a clock
 - scratch space

March 29, 2007
CS151 Lecture 2
17

Proof of Time Hierarchy Theorem

- contents and “virtual head” positions for M’s tapes
- M’s transition function and state
- $f(|x|)$ “+”s used as a clock
- scratch space
- initialize tapes
- simulate step of M, advance head on tape 3; repeat.
- can check running time is as claimed.
- Important detail: need to initialize tape 3 in time $O(f(n))$

March 29, 2007
CS151 Lecture 2
18

Proper Complexity Functions

- Definition: f is a **proper complexity function** if
 - $f(n) \geq f(n-1)$ for all n
 - there exists a TM M that outputs exactly $f(n)$ symbols on input 1^n , and runs in time $O(f(n) + n)$ and space $O(f(n))$.

March 29, 2007

CS151 Lecture 2

19

Proper Complexity Functions

- includes all reasonable functions we will work with
 - $\log n, \sqrt{n}, n^2, 2^n, n!, \dots$
 - if f and g are proper then $f + g, fg, f(g), f^g, 2^g$ are all proper
- can mostly ignore, but be aware it is a genuine concern:
- Theorem: \exists non-proper f such that **$\text{TIME}(f(n)) = \text{TIME}(2^{f(n)})$** .

March 29, 2007

CS151 Lecture 2

20

Hierarchy Theorems

- Does **genuinely** more space permit us to decide new languages?

Theorem (Space Hierarchy Theorem): For every proper complexity function $f(n) \geq \log n$:

$$\text{SPACE}(f(n)) \subsetneq \text{SPACE}(f(n) \log f(n)).$$

- Proof: same ideas.

March 29, 2007

CS151 Lecture 2

21

Robust Time and Space Classes

- What is meant by “robust” class?
 - no formal definition
 - reasonable changes to model of computation shouldn’t change class
 - should allow “modular composition” – calling subroutine in class (for classes closed under complement...)

March 29, 2007

CS151 Lecture 2

22

Robust Time and Space Classes

- Robust time and space classes:

$$L = \text{SPACE}(\log n)$$

$$\text{PSPACE} = \bigcup_k \text{SPACE}(n^k)$$

$$P = \bigcup_k \text{TIME}(n^k)$$

$$\text{EXP} = \bigcup_k \text{TIME}(2^{n^k})$$

March 29, 2007

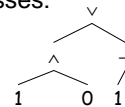
CS151 Lecture 2

23

Time and Space Classes

- Problems in these classes:

L : FVAL, integer multiplication, most reductions...



PSPACE : generalized geography, 2-person games...

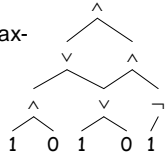
March 29, 2007

CS151 Lecture 2

24

Time and Space Classes

P : CVAL, linear programming, max-flow...



EXP : SAT, all of NP and much more...

March 29, 2007

CS151 Lecture 2

25

Relationships between classes

- How are these four classes related to each other?

- Time Hierarchy Theorem implies

$$P \subsetneq EXP$$

$$- P \subset TIME(2^n) \subsetneq TIME(2^{2n^3}) \subset EXP$$

- Space Hierarchy Theorem implies

$$L \subsetneq PSPACE$$

$$- L = SPACE(\log n) \subsetneq SPACE(\log^2 n) \subset PSPACE$$

March 29, 2007

CS151 Lecture 2

26

Relationships between classes

- Easy: $P \subset PSPACE$
- L vs. P, PSPACE vs. EXP ?

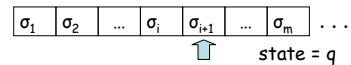
March 29, 2007

CS151 Lecture 2

27

Relationships between classes

- Useful convention: Turing Machine configurations. Any point in computation



represented by string:

$$C = \sigma_1 \sigma_2 \dots \sigma_i q \sigma_{i+1} \sigma_{i+2} \dots \sigma_m$$

- start configuration for single-tape TM on input $x: q_{start} X_1 X_2 \dots X_n$

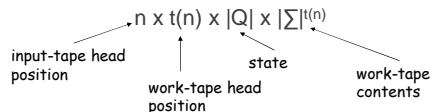
March 29, 2007

CS151 Lecture 2

28

Relationships between classes

- easy to tell if C yields C' in 1 step
- configuration graph: nodes are configurations, edge (C, C') iff C yields C' in one step
- # configurations for a 2-tape TM (work tape + read-only input) that runs in space $t(n)$



March 29, 2007

CS151 Lecture 2

29

Relationships between classes

- if $t(n) = c \log n$, at most $n \times (c \log n) \times c_0 \times c_1^{c \log n} \leq n^k$ configurations.
- can determine if reach q_{accept} or q_{reject} from start configuration by exploring config. graph of size n^k (e.g. by DFS)

- Conclude: $L \subset P$

March 29, 2007

CS151 Lecture 2

30

Relationships between classes

- if $t(n) = n^c$, at most $n \times n^c \times c_0 \times c_1^{n^c} \leq 2^{n^k}$ configurations.
- can determine if reach q_{accept} or q_{reject} from start configuration by exploring config. graph of size 2^{n^k} (e.g. by DFS)
- Conclude: **PSPACE** \subset **EXP**

March 29, 2007

CS151 Lecture 2

31

Relationships between classes

- So far: **L** \subset **P** \subset **PSPACE** \subset **EXP**
- believe all containments strict
- know **L** \subsetneq **PSPACE**, **P** \subsetneq **EXP**
- even before any mention of NP, two **major** unsolved problems:

$$L \stackrel{?}{=} P \quad P \stackrel{?}{=} \text{PSPACE}$$

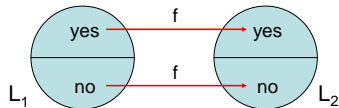
March 29, 2007

CS151 Lecture 2

32

A P-complete problem

- We don't know how to prove **L** \neq **P**
- But, can identify problems in **P** least likely to be in **L** using **P**-completeness.
- need stronger reduction (why?)



March 29, 2007

CS151 Lecture 2

33

A P-complete problem

- logspace reduction**: f computable by TM that uses $O(\log n)$ space
– denoted " $L_1 \leq_L L_2$ "
- If L_2 is **P**-complete, then L_2 in **L** implies **L** = **P** (homework problem)

March 29, 2007

CS151 Lecture 2

34

A P-complete problem

- Circuit Value (CVAL)**: given a variable-free Boolean circuit (gates $\wedge, \vee, \neg, 0, 1$), does it output 1?

Theorem: CVAL is **P**-complete.

- Proof**:
 - already argued in **P**
 - L arbitrary language in **P**, TM M decides L in n^k steps

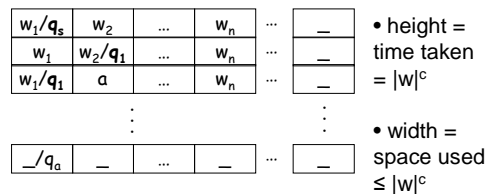
March 29, 2007

CS151 Lecture 2

35

A P-complete problem

- Tableau** (configurations written in an array) for machine M on input w :



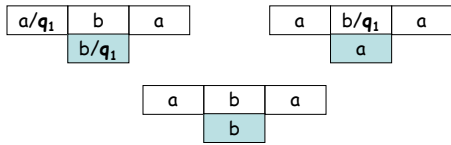
March 29, 2007

CS151 Lecture 2

36

A P-complete problem

- Important observation: contents of cell in tableau determined by 3 others above it:



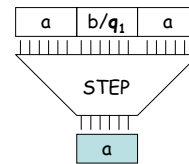
March 29, 2007

CS151 Lecture 2

37

A P-complete problem

- Can build Boolean circuit STEP
 - input (binary encoding of) 3 cells
 - output (binary encoding of) 1 cell



- each output bit is some function of inputs
- can build circuit for each
- size is independent of size of tableau

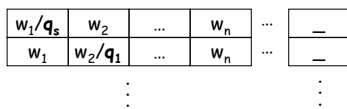
March 29, 2007

CS151 Lecture 2

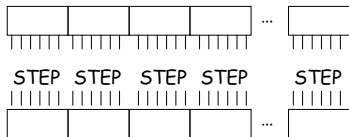
38

A P-complete problem

Tableau for M on input w



- $|w|^c$ copies of STEP compute row i from $i-1$

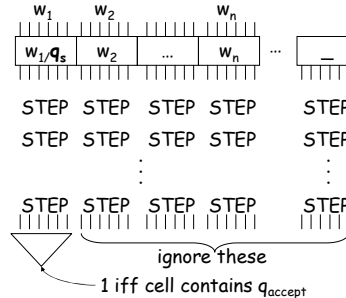


March 29, 2007

CS151 Lecture 2

39

A P-complete problem



This circuit $C_{M,w}$ has inputs $w_1 w_2 \dots w_n$ and $C(w) = 1$ iff M accepts input w.

logspace reduction
Size = $O(|w|^{2c})$

March 29, 2007

CS151 Lecture 2

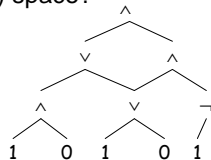
40

Answer to question

- Can we evaluate an n node Boolean circuit using $O(\log n)$ space?

- NO! (probably)

- CVAL in L if and only if $L = P$



March 29, 2007

CS151 Lecture 2

41