18.404 Recitation 8

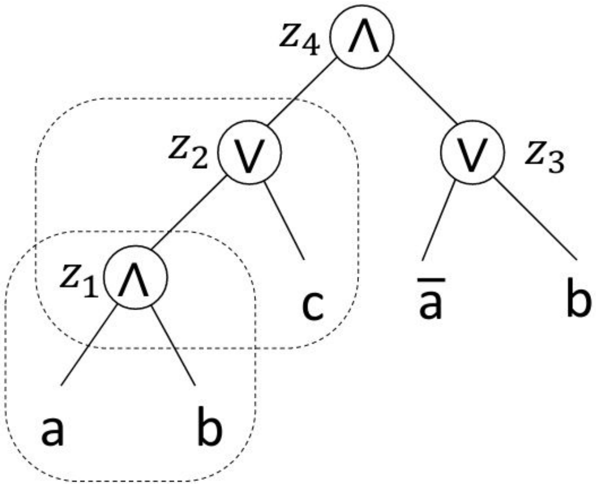
Oct 30, 2020

Today's Topics

- Review: 3-SAT is NP-Complete
- Definition: PSPACE, NPSPACE
- Review: TQBF ∈ PSPACE
- Review: LADDER ∈ NPSPACE
- Review: LADDER ∈ PSPACE
- 3-SAT $\leq_p$ NEQ-SAT
- 3-SAT $\leq_p$ DOMINATING-SET

Review: 3-SAT is NP-Complete

Assume: $\phi = \big((a \land b) \lor c\big) \land (\bar{a} \lor b)$



Logical equivalence: $(A \to B)$ and $(\bar{A} \lor B)$ $\quad \overline{(A \land B)}$ and $(\bar{A} \lor \bar{B})$

$$\phi' = \big((a \land b) \to z_1\big) \land \big((\bar{a} \land b) \to \overline{z_1}\big) \land \big((a \land \bar{b}) \to \overline{z_1}\big) \land \big((\bar{a} \land \bar{b}) \to \overline{z_1}\big)$$

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

⋮ repeat for each z_i

$$\land \; (z_4)$$

Definition: PSPACE

SPACE(f(n)) = { B | some deterministic 1-tape TM decides B in space O(f(n)) }

So:

PSPACE = $\cup_k$SPACE(n^k)

Definition: NPSPACE

NSPACE(f(n)) = { B | some nondet. 1-tape TM decides B in space O(f(n)) }

So:

$\mathrm{NPSPACE} = \bigcup_k \mathrm{NSPACE}(n^k)$

Review: TQBF

Definition: TQBF (True Qualified Boolean Formula) is a Boolean formula where every variable has either an exists (∃) or forall (∀) qualifier.

ex) $\varphi = \forall a \forall b \exists c. (a \lor c) \land ((b \land c) \lor \neg a)$

See if formula is satisfiable?

Review: TQBF $\in$ PSPACE

Proof: “On input <φ>

1. If φ has no qualifiers, has no variables. So either φ=True or φ=False. Output accordingly
2. If $\varphi = \exists x.\ \psi$, then evaluate ψ with x set to True/False. Accept if <u>either</u> evaluates to True. Reject if not
3. If $\varphi = \forall x.\ \psi$, then evaluate ψ with x set to True/False. Accept if <u>both</u> evaluates to True. Reject if not”

- On every recursive level, space used is constant. --- Only has to remember setting for x
- The number of recursive levels is linear. --- Each level removes one variable and there are a linear number of variables.

Review: LADDER

A LADDER is a list of words such that:

- List starts with a word **U** and ending with target word **V**
- Every immediately adjacent pair of words differ in only one symbol

$LADDER_{DFA}$ = { <B,u,v> | B is a DFA and L(B) contains a ladder $y_1, y_2, \ldots, y_k$ where y_1=u and y_k=v }

WORK
PORK
PORT
SORT
SOOT
SLOT
PLOT
PLOY
PLAY

Review: $LADDER_{DFA} \in NPSPACE$

Note:

- Cannot store sequence of guesses strings
- Cannot run indefinitely

Space usage: linear relative to the input
B/c only have to store one word at a time.

So, we are using polynomial space with nondeterminism, hence NPSPACE

Proof: “On input <B, u, v>

1. Let y=u and let m=|u|
2. Repeat at most t times where $t = |\Sigma|^m$
   a. Nondeterministically change one symbol of y at a time
   b. Reject if $y \notin L(B)$
   c. Accept if y=v
3. Reject if exceeds t steps”

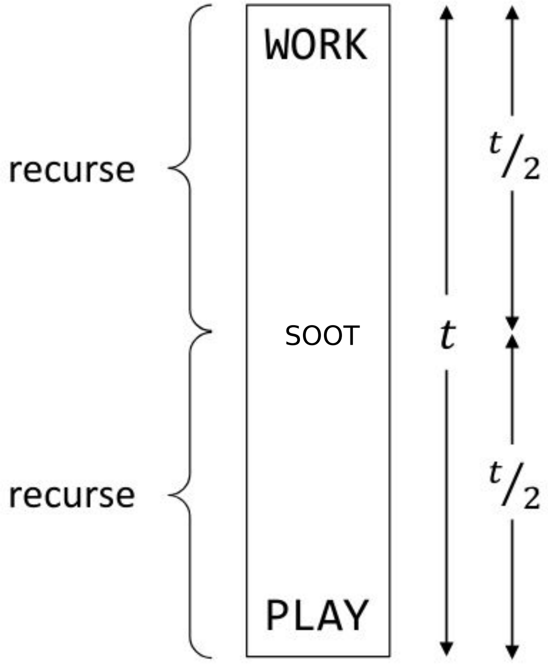
Review: $LADDER_{DFA} \in PSPACE$

Idea: Use recursion and implicit memoization

Proof: "On input <B,u,v,t>   Let $|m|=|u|=|v|$

1. For t=1, accept if $u,v \in L(B)$ and differ by at most 1 character
2. For t>1, repeat for each w of length $|u|$
   a. Recursively test $u \to_{t/2} w$, $w \to_{t/2} v$
   b. Accept if both accept
3. Reject if all fail"

Test <B,u,v> by running <B,u,v,t> where $t=|\Sigma|^m$



Review: $LADDER_{DFA} \in$ PSPACE (Space Analysis)

Recall: "On input <B,u,v,t>   Let $|m|=|u|=|v|$

1. For t=1, accept if u,v $\in$ L(B) and differ by at most 1 character
2. For t>1, repeat for each w of length $|u|$
   a. Recursively test $u \rightarrow_{t/2} w$, $w \rightarrow_{t/2} v$
   b. Accept if both accept
3. Reject if all fail"

Test <B,u,v> by running <B,u,v,t> where $t=|\Sigma|^m$

Space usage per recursive level is linear relative to input: O(m)
Number of recursive levels: log(t) = log(|Sigma|^m) = O(m)

Total space usage: O(m) * O(m) = O(m^2) which is polynomial relative to the input

3-SAT ⩽p NEQ-SAT

Definition: NEQ-SAT

Let φ be a 3cnf-formula and

The NEQ-SAT problem states that every clause in φ has literals with unequal truth values. In other words, at least one True and one False value.

Firstly: NEQ-SAT ∈ NP

3-SAT ≤p NEQ-SAT (cont.)

Part a)

Show that the negation of any satisfying NEQ-SAT formula is also a satisfying NEQ-SAT formula

3-SAT ≤p NEQ-SAT (cont.)

Part b)

$$(y_1 \vee y_2 \vee y_3)$$

$$(y_1 \vee y_2 \vee z_i) \quad \text{and} \quad (\overline{z_i} \vee y_3 \vee b)$$

where z_i is a new variable for each clause c_i, and b is a single additional new variable.

The reduction:

$(y_1 \vee y_2 \vee y_3) \rightarrow (y_1 \vee y_2 \vee z_i) \wedge (\neg z_i \vee y_3 \vee b)$

$(y_1 \vee y_2 \vee z_i) \wedge (\neg z_i \vee y_3 \vee b) \rightarrow (y_1 \vee y_2 \vee y_3)$