CS21
Decidability and Tractability

Lecture 23

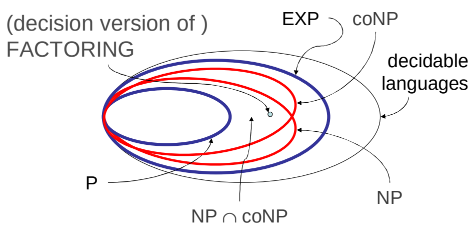
March 3, 2008

Outline

- the class PSPACE
- a PSPACE-complete problem
- PSPACE and 2-player games

Summary

- Picture of the way we believe things are:



Space complexity

Definition: the space complexity of a TM M is a function

$$f:\mathbf{N} \rightarrow \mathbf{N}$$

where f(n) is the maximum number of tape cells M scans on any input of length n.

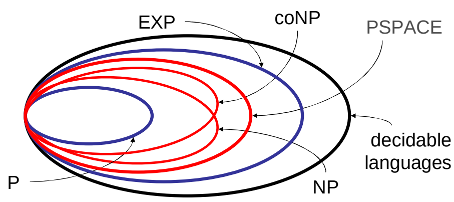
- "M uses space f(n)," "M is a f(n) space TM"

Space complexity

Definition: SPACE(t(n)) = {L : there exists a TM M that decides L in space O(t(n))}

$$\text{PSPACE} = \cup_{k \geq 1} \text{SPACE}(n^k)$$

PSPACE



- NP ⊂ PSPACE, coNP ⊂ PSPACE (proof?)
- PSPACE ⊂ EXP (proof?)
- containments believed to be proper

PSPACE

- A PSPACE-complete problem:
- Quantified Satisfiability:

QSAT = {φ : φ is a 3-CNF, and $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n \varphi(x_1, x_2, x_3, \ldots x_n)$ }

- example: $\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_3)$

$\exists x_1 \forall x_2 \exists x_3 \varphi$?

YES: x_1=T; if x_2=T, set x_3=F; if x_2=F, set x_3=T

PSPACE

- A PSPACE-complete problem:
- Quantified Satisfiability:

QSAT = {φ : φ is a 3-CNF, and $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n \varphi(x_1, x_2, x_3, \ldots x_n)$ }

- example: $\varphi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_2)$

$\exists x_1 \forall x_2 \exists x_3 \varphi$?

NO: x_1=T; if x_2=T…; x_1=F; if x_2=T…

QSAT is **PSPACE**-complete

<u>Theorem</u>: QSAT is **PSPACE**-complete.

- Proof:
  - in **PSPACE:** $\exists x_1 \forall x_2 \exists x_3 \ldots Qx_n \varphi(x_1, x_2, \ldots, x_n)$?
  - "$\exists x_1$": for both $x_1 = 0$, $x_1 = 1$, recursively solve $\forall x_2 \exists x_3 \ldots Qx_n \varphi(x_1, x_2, \ldots, x_n)$?
    - if at least one "yes", return "yes"; else return "no"
  - "$\forall x_1$": for both $x_1 = 0$, $x_1 = 1$, recursively solve $\exists x_2 \forall x_3 \ldots Qx_n \varphi(x_1, x_2, \ldots, x_n)$?
    - if at least one "no", return "no"; else return "yes"
  - base case: evaluating a 3-CNF expression
  - poly(n) recursion depth
  - poly(n) bits of state at each level

QSAT is **PSPACE**-complete

- given TM M deciding L ∈ **PSPACE**; input x
- 2^{n^k} possible configurations
- single START configuration
- assume single ACCEPT configuration

- define:

REACH(X, Y, i) ⇔ configuration Y reachable from configuration X in at most 2^i steps.

QSAT is **PSPACE**-complete

REACH(X, Y, i) ⇔ configuration Y reachable from configuration X in at most 2^i steps.

- Goal: produce 3-CNF $\varphi(w_1,w_2,w_3,\ldots,w_m)$ such that

$\exists w_1 \forall w_2 \ldots \exists w_m \varphi(w_1,\ldots,w_m)$

⇔ REACH(START, ACCEPT, n^k)

QSAT is **PSPACE**-complete

- for i = 0, 1, … n^k produce quantified Boolean **expressions** ψ_i(A, B, W)

$\exists w_1 \forall w_2 \ldots \psi_i(A, B, W)$ ⇔ REACH(A, B, i)

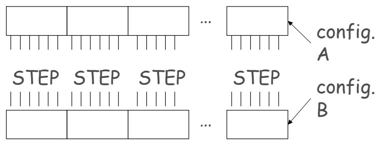
- convert ψ_{n^k} to 3-CNF φ
  - add variables V
- hardwire A = START, B = ACCEPT

$\exists w_1 \forall w_2 \ldots \exists V \varphi(W, V)$ ⇔ x ∈ L

QSAT is **PSPACE**-complete

– $\psi_o(A, B)$ = true iff
  - A = B or
  - A yields B in one step of M

Boolean expression of size $O(n^k)$



QSAT is **PSPACE**-complete

– Key observation:

REACH(A, B, i+1)

$\Leftrightarrow$

$\exists Z$ [REACH(A, Z, i) $\wedge$ REACH(Z, B, i)]

– cannot define $\psi_{i+1}(A; B; Z, W, W')$ to be

$\exists Z\ [\exists w_1 \forall w_2 \ldots\ \psi_i(A, Z, W) \wedge \exists w'_1 \forall w'_2 \ldots\ \psi_i(Z, B, W')\ ]$

(why?)

QSAT is **PSPACE**-complete

– Key idea: use quantifiers

– couldn't do $\psi_{i+1}(A; B; Z, W, W')$ =

$\exists Z\ [\exists w_1 \forall w_2 \ldots\ \psi_i(A, Z, W) \wedge \exists w'_1 \forall w'_2 \ldots\ \psi_i(Z, B, W')\ ]$

– define $\psi_{i+1}(A; B; Z, X, Y, W)$ to be

$\exists Z \forall X \forall Y\ [((X=A \wedge Y=Z) \vee (X=Z \wedge Y=B)) \Rightarrow$

$\exists w_1 \forall w_2 \ldots\ \psi_i(X, Y, W)]$

– $\psi_i(X, Y, W)$ is preceded by quantifiers

– move to front (they don't involve X,Y,Z,A,B)

QSAT is **PSPACE**-complete

$\psi_o(A, B)$ = true iff A = B or A yields B in 1 step

$\psi_{i+1}(A; B; Z, X, Y, W)$ =

$\exists Z \forall X \forall Y\ [((X=A \wedge Y=Z) \vee (X=Z \wedge Y=B)) \Rightarrow$

$\exists w_1 \forall w_2 \ldots\ \psi_i(X, Y, W)]$

– $|\psi_0| = O(n^k)$

– $|\psi_{i+1}| = O(n^k) + |\psi_i|$

– total size of ψ_{nk} is $O(n^k)^2$ = poly(n)

– reduction runs in polynomial time

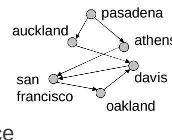
PSPACE and games

QSAT = {φ : φ is a 3-CNF, and $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n\ \varphi(x_1, x_2, x_3, \ldots x_n)$ }

- Think of as 2-player game (player 1 trying to satisfy φ; player 2 adversary):
  – player 1 picks truth value for x_1
  – player 2 picks truth value for x_2
  – player 1 picks truth value for x_3…
- $\varphi \in$ QSAT iff player 1 can win no matter what player 2 does.

PSPACE and games

- General phenomenon: many 2-player games are PSPACE-complete.
  – 2 players I, II
  – alternate picking edges
  – lose when no unvisited choice



- GEOGRAPHY = {(G, s) : G is a directed graph and player I can win from node s}

PSPACE

Theorem: GEOGRAPHY is PSPACE-complete.

Proof:
- in PSPACE (proof?)
- PSPACE-hard. reduction from QSAT.

GEOGRAPHY is PSPACE-complete

- We are reducing from the language:

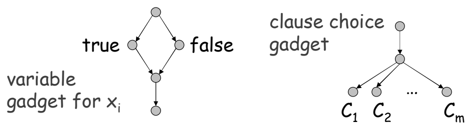
QSAT = {φ : φ is a 3-CNF, and $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n\ \varphi(x_1, x_2, x_3, \ldots x_n)$ }

to the language:

GEOGRAPHY = {(G, s) : G is a directed graph and player I can win from node s}

PSPACE

$\exists x_1 \forall x_2 \exists x_3 \ldots \forall x_n\ \varphi(x_1, x_2, \ldots, x_n)$?



PSPACE

$\exists x_1 \forall x_2 \exists x_3 \ldots (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_3 \vee x_1) \wedge \ldots \wedge (x_1 \vee \neg x_2)$