CS21
Decidability and Tractability
Lecture 25
March 4, 2022

Outline
• The complexity class PSPACE
  – a PSPACE-complete problem
  – PSPACE and 2-player games

• Challenges to Extended Church-Turing
  – randomized computation
  – quantum computation

PSPACE
• A PSPACE-complete problem:
• Quantified Satisfiability:
  \[ \text{QSAT} = \{ \phi : \phi \text{ is a 3-CNF, and } \exists x_1 \forall x_2 \exists x_3 x_4 \exists x_5 \ldots \forall x_n \phi(x_1, x_2, x_3, \ldots, x_n) \} \]

• example: \( \phi = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_2 \lor x_3) \)
  \[ \exists x_1 \forall x_2 \exists x_3 \phi? \]
  YES: \( x_1 = T; \) if \( x_2 = T, \) set \( x_3 = F; \) if \( x_2 = F, \) set \( x_3 = T \)

QSAT is PSPACE-complete

Theorem: QSAT is PSPACE-complete.

Proof:
– in PSPACE: \( 3x_1 x_2 x_3 x_4 \ldots \exists x_n \phi(x_1, x_2, \ldots, x_n) \)?
– “\( \exists x_1 \)”:
  for both \( x_1 = 0, x_1 = 1, \) recursively solve \( \forall x_2 \exists x_3 x_4 \exists x_5 \ldots \forall x_n \phi(x_1, x_2, x_3, \ldots, x_n) \)?
  • if at least one “yes”, return “yes”; else return “no”
– “\( \forall x_1 \)”:
  for both \( x_1 = 0, x_1 = 1, \) recursively solve \( 3x_2 x_3 x_4 \exists x_5 \phi(x_1, x_2, x_3, \ldots, x_n) \)?
  • if at least one “no”, return “no”; else return “yes”
– base case: evaluating a 3-CNF expression
– \( \text{poly(n)} \) recursion depth
– \( \text{poly(n)} \) bits of state at each level

QSAT is PSPACE-complete

– given TM M deciding \( L \in \text{PSPACE} \); input \( x \)
– \( 2^n \) possible configurations
– single START configuration
– assume single ACCEPT configuration
– define:
  \( \text{REACH(X, Y, i)} \iff \text{configuration Y reachable from configuration X in at most 2 steps.} \)
QSAT is PSPACE-complete

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

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

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

$\iff$ REACH(START, ACCEPT, n^i)

March 4, 2022
CS21 Lecture 25
7

QSAT is PSPACE-complete

– for $i = 0, 1, \ldots, n^k$ produce quantified Boolean expressions $\psi_i(A, B, W)$ such that $\forall A, B$:

$$\exists w_1 \forall w_2 \ldots \psi_i(A, B, W) \iff \text{REACH}(A, B, i)$$

– convert $\psi_n$ to 3-CNF $\varphi$

• add variables $V$

– hardwire $A = \text{START}$, $B = \text{ACCEPT}$

March 4, 2022
CS21 Lecture 25
8

QSAT is PSPACE-complete

– Key observation:

$$\exists Z [\text{REACH}(A, Z, i) \land \text{REACH}(Z, B, i)]$$

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

$$\exists Z \exists w_1 \forall w_2 \ldots \psi_i(A, Z, W) \land \exists w_1' \forall w_2' \ldots \psi_i(Z, B, W')$$

(why?)

March 4, 2022
CS21 Lecture 25
9

QSAT is PSPACE-complete

– Key idea: use quantifiers

– couldn’t do $\psi_i(A; B; Z, W, W') =

$$\exists Z [\exists w_1 \forall w_2 \ldots \psi_i(A, Z, W) \land \exists w_1' \forall w_2' \ldots \psi_i(Z, B, W')]$$

– define $\psi_i(A; B; Z, X, Y, W)$ to be

$$\exists Z \forall X \forall Y [((X = A \land Y = Z) \lor (X = Z \land Y = B)) \implies \exists w_1 \forall w_2 \ldots \psi_i(X, Y, W)]$$

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

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

March 4, 2022
CS21 Lecture 25
10

QSAT is PSPACE-complete

– $\psi_0(A, B) = \text{true iff } A = B \text{ or } A \text{ yields } B \text{ in 1 step}$

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

\exists Z \forall X \forall Y [((X = A \land Y = Z) \lor (X = Z \land Y = B)) \implies \exists w_1 \forall w_2 \ldots \psi_i(X, Y, W)]$$

– $|\psi_0| = O(n^4)$

– $|\psi_{i+1}| = O(n^4) + |\psi_i|$

– total size of $\psi_n$ is $O(n^4)^2 = \text{poly}(n)$

– reduction runs in polynomial time

March 4, 2022
CS21 Lecture 25
11
PSPACE and games

QSAT = \{ \varphi : \varphi \text{ 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) \}\n
- Think of as 2-player game (player 1 trying to satisfy \( \varphi \); 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 \( x_3 \)
  - player 1 picks truth value for \( x_3 \)

- \( \varphi \in \text{QSAT} \) iff player 1 can win no matter what player 2 does.

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:
  \[
  \text{QSAT} = \{ \varphi : \varphi \text{ is a 3-CNF, and } \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n \varphi(x_1, x_2, x_3, \ldots, x_n) \}\n  \]
  to the language:

\[
\text{GEOGRAPHY} = \{(G, s) : G \text{ is a directed graph and player I can win from node } s \}\n\]
Extended Church-Turing Thesis
• the belief that TMs formalize our intuitive notion of an efficient algorithm is:

  **The "extended" Church-Turing Thesis**
  everything we can compute in time \( t(n) \)
  on a physical computer can be computed on a Turing Machine in time \( t(n)^{O(1)} \) (polynomial slowdown)

• randomized computation challenges this belief

Randomness in computation
• Example of the power of randomness
• Randomized complexity classes

Communication complexity
• Goal: compute \( f(x, y) \) while communicating as few bits as possible between Alice and Bob

  two parties: Alice and Bob
  function \( f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\} \)
  Alice holds \( x \in \{0,1\}^n \); Bob holds \( y \in \{0,1\}^n \)

• count number of bits exchanged (computation free)
• at each step: one party sends bits that are a function of held input and received bits so far