1

QSAT is **PSPACE**-complete

Theorem: 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

2

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.

3

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

$$\exists w_1 \forall w_2 \ldots \exists w_m\, \varphi(w_1,\ldots,w_m)$$
$$\Leftrightarrow \text{REACH(START, ACCEPT, } n^k\text{)}$$

4

QSAT is **PSPACE**-complete

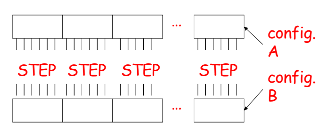
- for i = 0, 1, … n^k produce quantified Boolean **expressions** ψ_i(A, B, W) such that ∀ A,B:
  $\exists w_1 \forall w_2 \ldots \psi_i(A, B, W) \Leftrightarrow$ REACH(A, B, i)
- convert ψ_{n^k} to 3-CNF φ
  - add variables V
- hardwire A = START, B = ACCEPT
  $\exists w_1 \forall w_2 \ldots \exists V\, \varphi(W, V) \Leftrightarrow x \in L$

5

QSAT is **PSPACE**-complete

- ψ_o(A, B) = true iff
  - A = B or
  - A yields B in one step of M

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



6

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?)

7

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)

8

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_{n^k} is $O(n^k)^2$ = poly(n)

– reduction runs in polynomial time

9

PSPACE and games

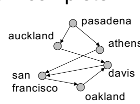
QSAT = {φ : φ is a 3-CNF, and
$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n\ \varphi(x_1, x_2, x_3, \ldots x_n)$ }

- 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.

10

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}

11

PSPACE

Theorem: GEOGRAPHY is PSPACE-complete.

Proof:

– in PSPACE (proof?)

– PSPACE-hard. reduction from QSAT.

12

GEOGRAPHY is PSPACE-complete

- We are reducing from the language:

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

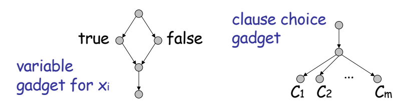
to the language:

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

13

PSPACE

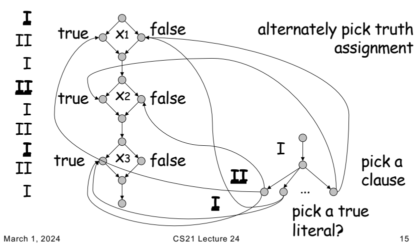
$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n\ \varphi(x_1, x_2, x_3, \ldots x_n)$?



14

PSPACE

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



15

Outline

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

16

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

17

Randomness in computation

- Example of the power of randomness
- Randomized complexity classes

18

Communication complexity

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 $y \in \{0,1\}^n$

- Goal: compute $f(x, y)$ while communicating as few bits as possible between Alice and Bob
- 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

Communication complexity

- simple function (equality):

  $EQ(x, y) = 1$ iff $x = y$

- simple protocol:
  - Alice sends x to Bob (n bits)
  - Bob sends EQ(x, y) to Alice (1 bit)
  - total: n + 1 bits
  - (works for any predicate f)

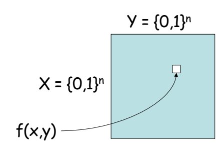
Communication complexity

- Can we do better?
  - deterministic protocol?
  - probabilistic protocol?
    - at each step: one party sends bits that are a function of held input and received bits so far and the result of some coin tosses
    - required to output f(x, y) with high probability over all coin tosses

Communication complexity

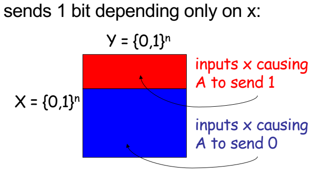
Theorem: no deterministic protocol can compute $EQ(x, y)$ while exchanging fewer than n+1 bits.

- Proof:
  - "input matrix":



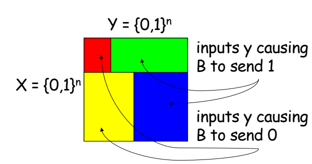
Communication complexity

- assume 1 bit sent at a time (but proof works for general case)
- A sends 1 bit depending only on x:



Communication complexity

- B sends 1 bit depending only on y and received bit:



Communication complexity

- at end of protocol involving k bits of communication, matrix is partitioned into at most 2^k combinatorial rectangles
- bits sent in protocol are the same for every input (x, y) in given rectangle
- conclude: f(x,y) must be constant on each rectangle

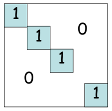
25

Communication complexity

Matrix for EQ:

$Y = \{0,1\}^n$

$X = \{0,1\}^n$



- any partition into combinatorial rectangles with constant f(x,y) must have at least $2^n + 1$ rectangles
- protocol that exchanges ≤ n bits can only create 2^n rectangles, so must exchange at least n+1 bits.

26