CS21
Decidability and Tractability

Lecture 23

March 2, 2018

Outline

- The complexity class PSPACE
  - PSPACE and 2-player games
- challenges to the extended Church-Turing Thesis
  - randomized computation
  - quantum computation

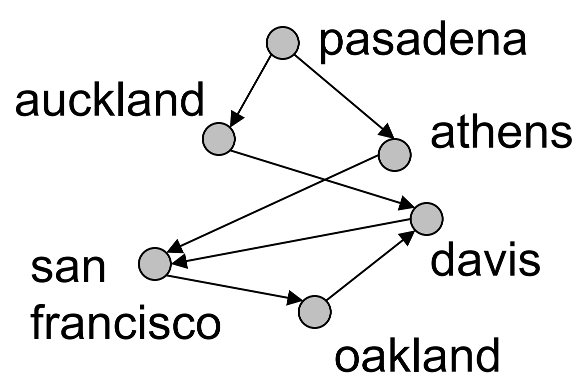
PSPACE and games

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:

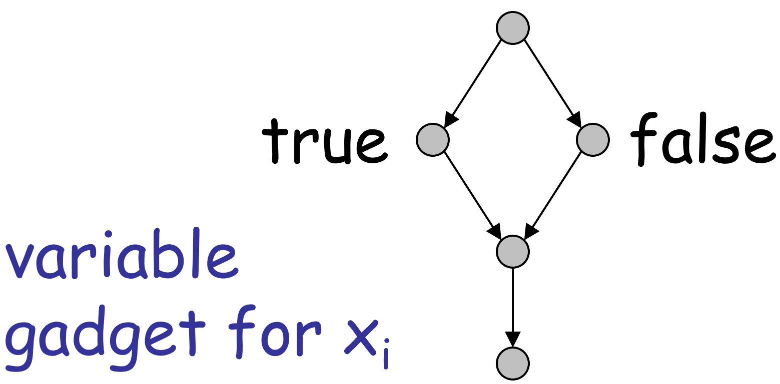
QSAT = {φ : φ is a 3-CNF, and $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n\, \varphi(x_1, x_2, x_3, \ldots x_n)$ }

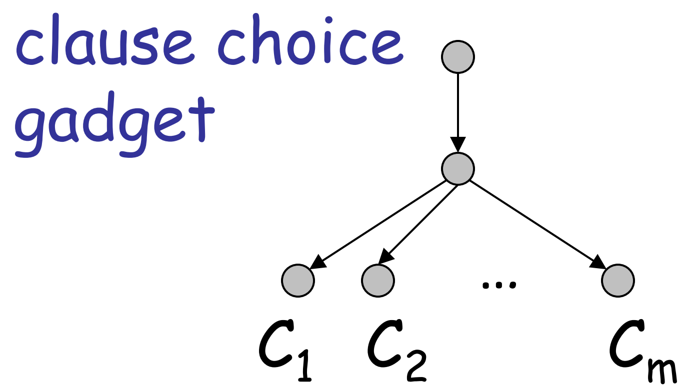
to the language:

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

PSPACE

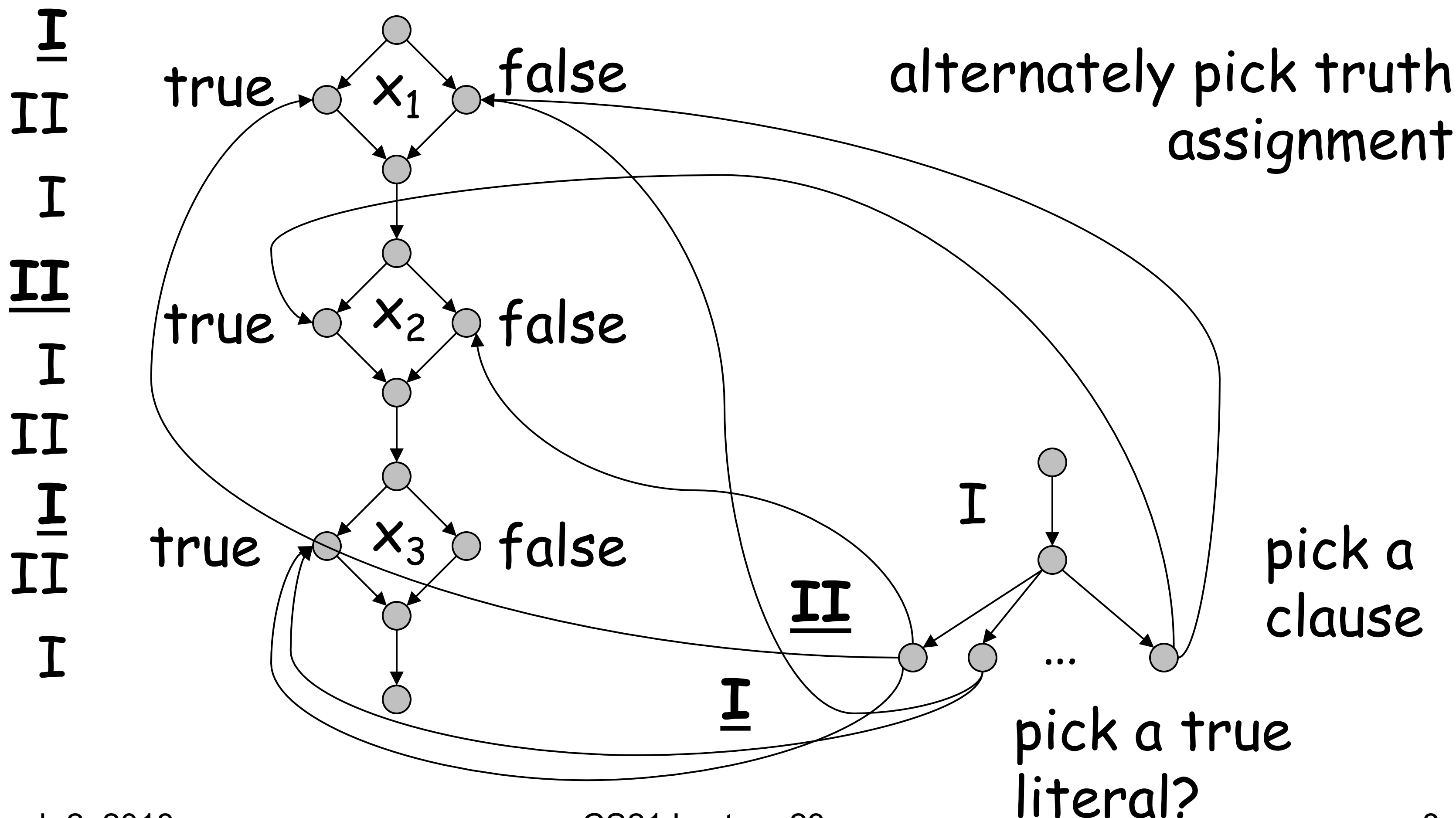
$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \ldots \forall x_n \, \varphi(x_1, x_2, \ldots, x_n)?$$





PSPACE

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



Challenges to the extended Church-Turing thesis

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

two parties: Alice and Bob

function $f:\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$

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)

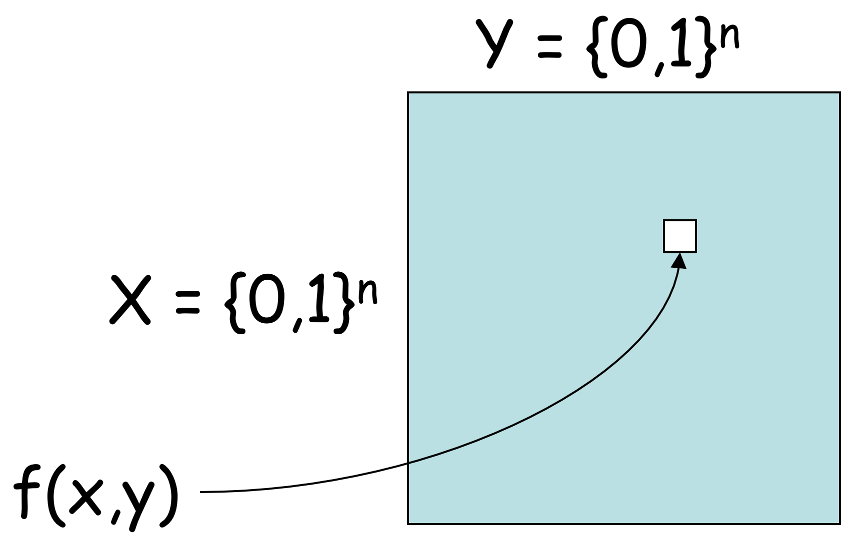
Communication complexity

- 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.

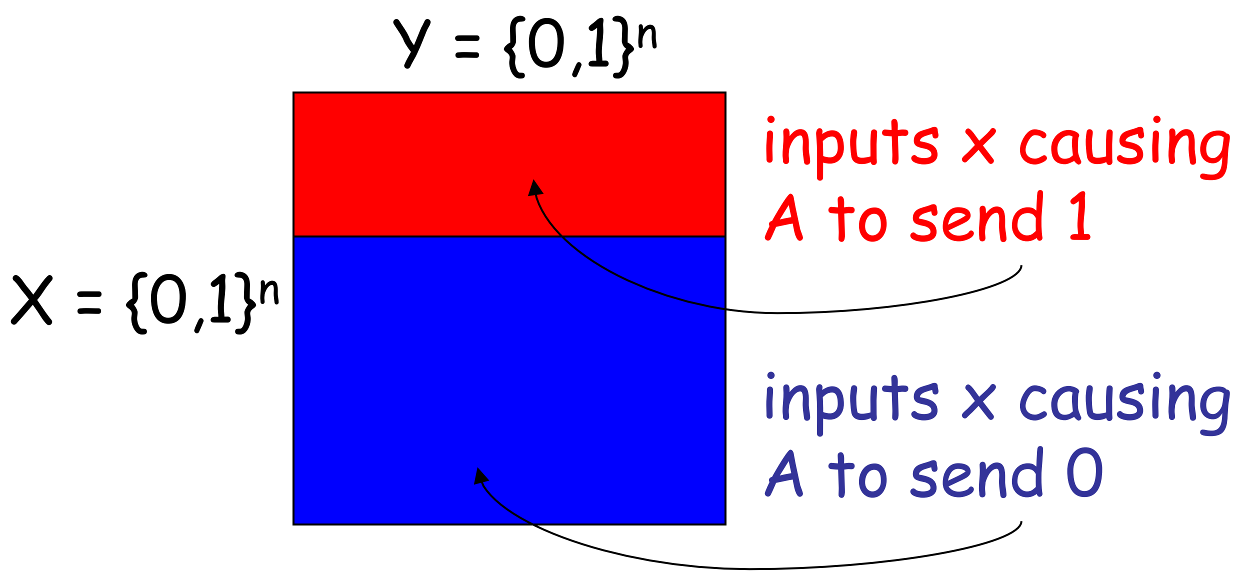
- Proof:
  - "input matrix":



Communication complexity

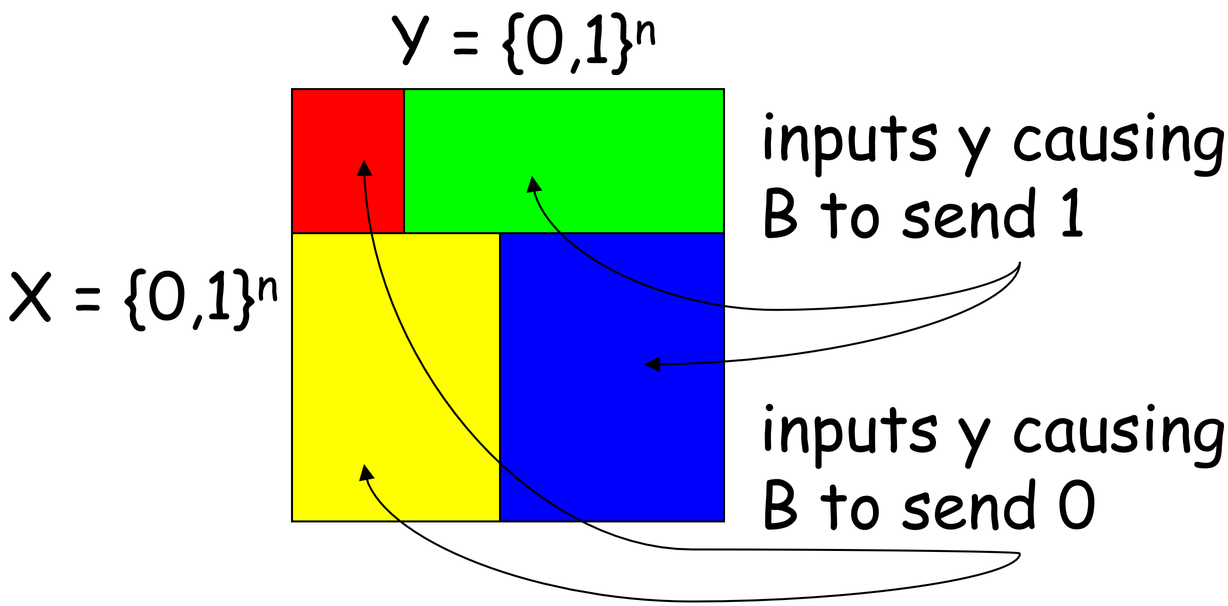
– assume 1 bit sent at a time (but proof works for general case)

– A sends 1 bit depending only on x:



Communication complexity

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



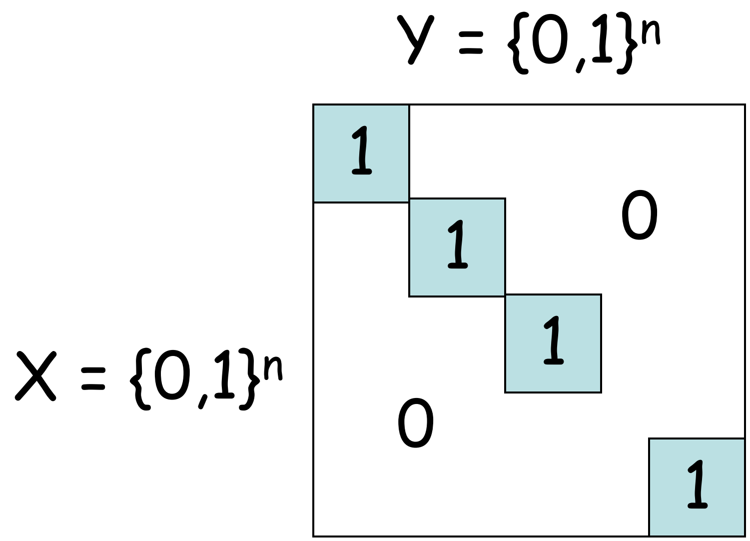
Communication complexity

– 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

Communication complexity

Matrix for EQ:



– 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.

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

- protocol for EQ employing randomness?
  - Alice picks random prime p in $\{1...4n^2\}$, sends:
    - p
    - (x mod p)
  - Bob sends:
    - (y mod p)
  - players output 1 if and only if:

    (x mod p) = (y mod p)

Communication complexity

- $O(\log n)$ bits exchanged
- if $x = y$, always correct
- if $x \neq y$, incorrect if and only if:

  p divides $|x – y|$
- # primes in range is $\geq 2n$
- # primes dividing $|x – y|$ is $\leq n$
- probability incorrect $\leq$ 1/2

Randomness gives an exponential advantage!!