Michael Forbes
miforbes@illinois.edu
2019-02-05.4 → 2019-02-07.4
2019-02-07.2c CS 579

ck + 2 p: 10, 8, 10, 10, 10, 10

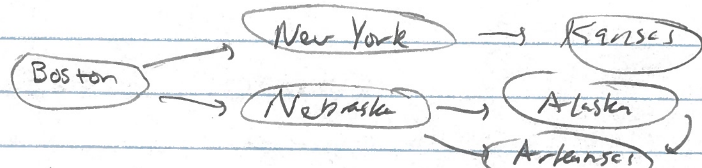
CS 579 Computational Complexity: lecture 8 — K6 21

admin: – ps1 back avg ≈ 55
– ps2 due next week – all info inside page [name, collab]
– send to all course staff [me, TA]

last time: – PSPACE = NPSPACE
– TQBF is PSPACE-complete

today: – generalized geography
– games
– logspace: L, NL [nondeterminism]

geography:

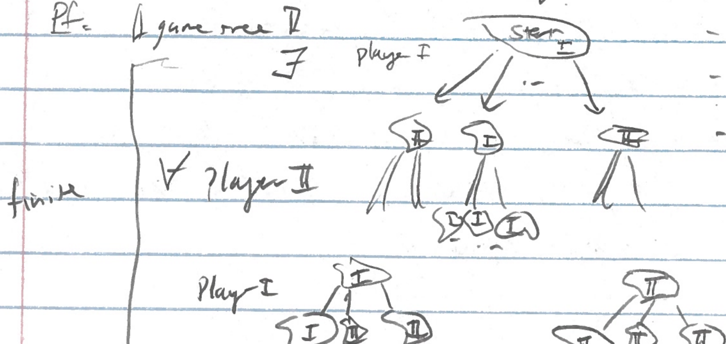


rules: – A → B if last letter of A = first letter of B
– player I starts, players alternate in picking new edge w/ no repeated vertex
– whoever gets stuck loses

generalized geography
directed graph G = (V, E), designated start node s
[rules are the same]
GG = { <G, s> : player I has a winning strategy in G starting at s }
in perfect play player I always wins

thm: in any finite game [finite # configs] [–plays] w/ no ties, exactly one player has a winning strategy

Pf: [game tree]



– each player selects best option available
– inductively define optimal play
– exactly one player has winning strategy at start

thm: GG is PSPACE complete
Pf: – GG ∈ PSPACE: recursively compute optimal play [same also as TQBF] – game tree has ≤ |V| depth
– exp |G| size [reuse space]

– TQBF ≤p PSPACE

Michael Forbes
mforbes@illinois.edu
2019-02-07.2 ⟸ 2019-02-07.1
2019-02-07.3 ⟹
CS574

[[fun!]]

def: formula game on boolean formula $\varphi(x_1, \dots x_n)$
- players alternate in choosing values $x_1, \dots x_n \in \{0,1\}$
- at end: $\varphi(\bar{x}) = 1 \Rightarrow$ player I wins
- $= 0$ II

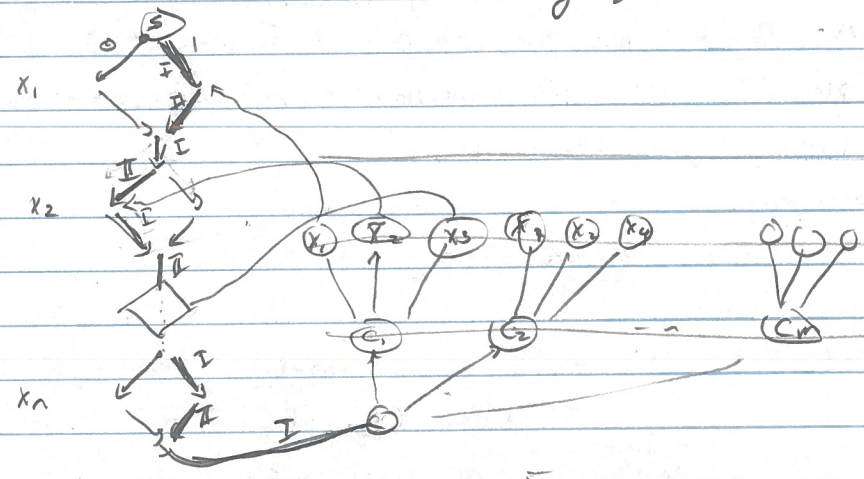
thm: player I has winning strategy in formula game on φ iff $\exists x_1 \forall x_2 \dots : \varphi \in$ TQBF
[[TQBF]]

prog: FORMULAGAME, GG

Pf - Given: $\varphi = \exists x_1 \forall x_2 \dots [(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge \dots]$

assume: - quantifier alternate starting w/ $\exists$, ending w/ $\exists$ (add dummy vars $\exists x \exists y \mapsto \exists x \forall z \exists y$)
- φ is CNF [[psat]]

want: $G = (V, E)$, $S \subseteq V$
- variable gadgets [[by picture]]
- clause gadgets



II win iff literal is false
I win iff some literal true
II iff clause sat
win iff some clause false
I win iff $\varphi(x) = 1$

[[forces players to alternate in choosing assignments]]

hence: - $\varphi \in$ TQBF iff $\langle G, S \rangle \in$ GG [[correctness]]
- $\langle G, S \rangle$ polytime computable [[complexity]]. ∎

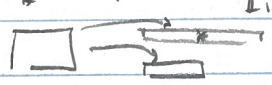
Rmks: - Chess is PSPACE-complete [[actual game limit]]
[[can generalize to $n \times n$ board]]

Questions

Q: computing in sublinear $= o(n)$ space?

$\Rightarrow s(n) \geq n$ always! [[trivial]] [[we are smarter than that]]

def: a TM with read-only input tape
[[two tapes]]
[[input tape]]
- read-only [[transition function cannot overwrite]]
- initialized w/ input
- read/write



Michael Forbes
miforbes@illinois.edu

2019-02-07.2 → 2019-02-07.3
2019-02-07.4 ← CS579

M computes in space $S(n)$ if - always halts
- all x uses $\le S(|x|)$ work tape cells (if input is free in space)

Rmk: - read-only input tape initialized w/ "$\sqcup x \sqcup$"
(allows edge detection, otherwise would require read/write access)

- Why? input can be too big (internet, big data, large hadron collider)

def: $L = SPACE(O(\lg n))$ (logspace) (O(1) pointers to input, natural, also)
$NL = NSPACE(O(\lg n))$ (nondeterministic logspace)

Prop: $A = \{x : x \in \{0,1\}^* \ x = x^R\} \in L$

Pf: "on input x:

for $1 \le i \le n = |x|$
if $x_i \neq x_{n-i}$, reject
"
accept

correctness: clear

complexity: computing n — $O(\lg n)$ bits
iterating over i — $O(\lg n)$ bits
computing x_i — move i cells from start, get x_i, move back to start — $O(\lg n)$ bits
x_{n-i} "

$\Rightarrow O(\lg n)$ bits

$PATH = \{\langle G, s, t\rangle : s \rightsquigarrow t$ path in $G\}$

Prop: $PATH \in NL$

Pf: "on input $\langle G, s, t\rangle$

1) $v \leftarrow s$ — $O(\lg |V|)$ bits
2) for $\le |V|$ steps — $O(\lg |V|)$ bits
   a) if $v = t$, accept
   b) guess u, neighbor of v — $O(\lg |V|)$ bits
   c) $v \leftarrow u$
3) reject

Michael Forbes
miforbes@illinois.edu
2019-02-07.4 ← 2019-02-07.3
↘ 2019-02-12.1

Q: PATH ∈ L? L vs NL? [[open, conj not equal]]

Thm [Savitch]: $s(n) \geq \lg n$. $NSPACE(s) \subseteq SPACE(s^2)$ [[same proof, also in new model]]

Cor:

$$NL \subseteq SPACE(\lg^2 n) \subseteq TIME(n^{O(\lg n)})$$

Prop: $NSPACE(s(n)) \subseteq TIME(2^{O(s(n))})$ $s(n) \geq \lg n$

(Cor: $NL \subseteq NP \subseteq P$)

↳ PF: def configuration of TM: read-only input tape

⇒ current state
- work tape contents
- head locations ~ input tape, ~ work tape

Lem: space $s(n)$, n ⇒ # configs $\leq |Q| \cdot |\Gamma|^{s(n)} \cdot s(n) \cdot n$

$$\leq 2^{O(s(n))} \cdot n \leq 2^{O(s(n))}$$ ← $s(n) \geq \lg n$

space $s(n)$ machine always halts

$$\Rightarrow \leq 2^{O(s(n))} \text{ configs seen} \Rightarrow \text{time} \leq 2^{O(s(n))}$$

Prop: $NL \subseteq P$ [[better than Savitch]]

PF: def: configuration graph $G = (V, E)$

for space $s(n)$ TM M - $V = \{$ all space $s(n)$ configs of $M \}$

- $E = \{ c \to c' : \delta_M(c) \ni c' \}$

Clm: M accepts x iff $c_{init} \rightsquigarrow c_{acc}$ in G (w.l.o.g. unique acc)

algo: - construct G ← size $\leq \text{poly}(2^{O(s(n))}) = \text{poly}(n)$

- solve ⇒ PATH $\in$ P ∎

today: - geography, games, PSPACE completeness

- L, NL

next time: - NL-completeness

- coNL