THE CLASS PSPACE

Thejus Gore

B100090EC

March 12, 2014

$$PSPACE = \bigcup_k SPACE(n^k)$$

$$NPSPACE = PSPACE$$

$$P \subseteq PSPACE$$

$$NP \subseteq NPSPACE$$

so $NP \subseteq PSPACE$

A TM that uses $f(n)$ space can have atmost $f(n)2^{O(f(n))}$ different configurations so $PSPACE \subseteq EXPTIME$

$$P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$$

A language B is $PSPACE - complete$ if

- $B \in PSPACE$ and
- $\forall A \in PSPACE \qquad A \leq_P B$

TQBF

- $\forall$ Universal Quantifier
- $\exists$ Existential Quantifier
- $\forall x \exists y[y > x]$, $x, y \in N$
  Every natural number has another natural number larger than it.
- fully quantified boolean formula
  $\phi = \forall x \exists y[(x \vee y) \wedge (\bar{x} \vee \bar{y})]$ is true
- $TQBF = \{\langle\phi\rangle | \phi$ is a true fully quantified Boolean formula$\}$

TQBF **is** *PSPACE − complete*

Polynomial space algorithm deciding *TQBF*

$T =$ "On input $\langle\phi\rangle$, a fully quantified Boolean formula "

- If ϕ contains no quantifiers, evaluate ϕ
  accept if true; else *reject*
- If $\phi = \exists x\psi$, recursively call T on ψ, first with x =0 and then x =1. If either is accept , *accept*; else *reject*
- If $\phi = \forall x\psi$, recursively call T on ψ, first with x =0 and then x =1. If both are accept , *accept*; else *reject*

Depth is atmost number of variables. Each recursion stores one variable.

T runs in linear space

Polynomial time reduction $A <_P TQBF$

$w \in A, \qquad \phi \in TQBF$

$w \notin A, \qquad \phi \notin TQBF$

$\phi_{c_1,c_2,t}$ is true if and only if M can go from configuration $c_1$to c_2 in atmost t steps.

Each configuration has n^k cells

t=1: c_1= c_2 or $c_1$to c_2 in 1 step

for t¿1

$\phi_{c_1,c_2,t} = \exists m_1[\phi_{c_1,m_1,\lceil t/2 \rceil} \wedge \phi_{m_1,c_2,\lceil t/2 \rceil}]$

m_1 represents a configuration of M.

t is cut in half. Size of formula doubles. $t = 2^{df(n)}$.

Reduce size

$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3, c_4) \in (c_1, m_1), (m1, c_2)[\phi_{c_3,c_4,\lceil t/2 \rceil}]$

c_3	c_1, m_1
c_4	m_1, c_2

$\phi_{c_{start},c_{accept},h}$

Each recursion adds a portion of formula linear in the size of configuration , $O(f(n))$.

Number of levels $log(2^{df(n)})$, $O(f(n))$.

$O(f^2(n))$

Formula Game

$\phi = \exists x_1 \forall x_2 \exists x_3 ... Q x_k [\psi]$, $Q \in \{\forall, \exists\}$

Player A($\forall$) and E($\exists$) select values of $x_1, x_2 .... x_k$

Player E wins if ψ is TRUE

Winning strategy

$FORMULA-GAME = \{\langle\phi\rangle \mid$ Player E has a winning strategy in the formula game associated eith $\phi\}$

$FORMULA-GAME = TQBF$

GENERALIZED GEOGRAPHY

- Simple directed graph with a start node
- The player unable to extend the path fails

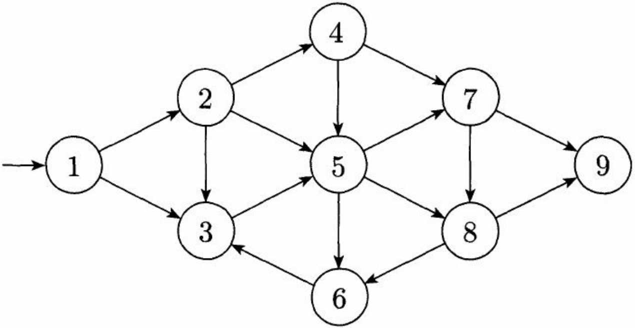


Figure : A sample generalized geography game

$GG = \{\langle\phi\rangle|$ Player 1 has a winning strategy for the generalized geography game played on graph G at node b $\}$

GG is $PSPACE - complete$

$M =$"On input $\langle G, b\rangle$, where G is a directed graph and b is a node in G "

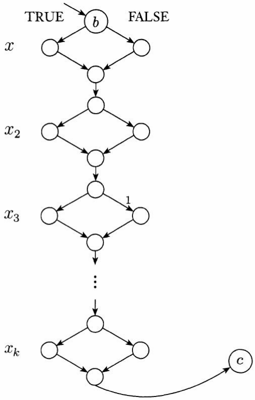
- If b has outdegree 0, *reject*
- Remove node b and all arrows touching it to get a new graph G_1
- For each of the nodes $b_1, b_2, ..., b_k$ that b originally pointed at, recursively call M on $\langle G_i, b_i\rangle$
- If all of these accept Player 2 has a winning strategy, so *reject*. else *accept*"

Each recursion adds a single node to the stack. At most m number of nodes. Linear space.
$FORMULA - GAME$ is polynomial time reducible to GG

$\phi = \exists x_1 \forall x_2 \exists x_3 ... Q x_k [\psi],\qquad Q \in \{\forall, \exists\}$
ϕ is mapped to GG

- Quantifiers begin and end with $\exists$
- Alternate between $\exists$ and $\forall$

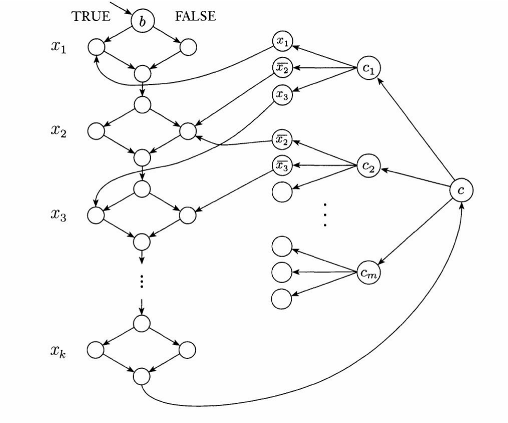
Player 1 - E
Player 2 - A



TRUE
FALSE
b
x
x_2
x_3
1
x_k
c

Player 1 can win if Player E wins
At node c Player 2 can choose a node corresponding to one of the clauses of ψ

If ϕ is FALSE, Player 2 may win by selecting the unsatisfied clause. Any literal that Player 1 may then pick is FALSE and is connected to the side of the diamond that hasn't yet been played.Thus Player 2 may play the node in the diamond,but then Player 1 is unable to move and loses. If ϕ is TRUE,any clause that Player 2 picks contains a TRUE literal. Player 1 selects that literal after Player 2's move. Because the literal is TRUE, it is connected to the side of the diamond that has already been played, so Player 2 is unable to move and loses


TRUE
b
FALSE
x_1
x_2
x_3
x_k
x_1
$\overline{x_2}$
x_3
$\overline{x_2}$
$\overline{x_3}$
c_1
c_2
c_m
c