PSPACE

PSPACE

PSPACE is the class of all languages decided by polynomial space Turing machines. One can think of NPSPACE, but this is identical to PSPACE by Savitch's Theorem.

NP is included in PSPACE.

PSPACE completeness under polynomial-time reductions: Karp 1972 ($\{\langle G, w\rangle \mid G$ is a CSG that generates $w\}$ is PSPACE complete – even though CSL's are exactly NSPACE(n)).

TQBF

A **quantified formula** is one in which **quantifiers** $\exists$ and $\forall$ may appear. Each quantifier should be followed immediately by a variable, which is **bound** to the quantifier. $\exists x$ means "for some value of x" and $\forall x$ means "for every value of x." A formula may contain constants 0, 1 as well as variables.

The quantification applies to every occurrence of the variable to the right of the point of quantification within the innermost pair of parentheses that contains the quantifier. This is called the **scope** of the quantifier.

A formula is **fully quantified** if all the variables are bound.

A formula is in **prenex normal form** if all of its quantifiers appear at the beginning.

$TQBF = \{\phi \mid \phi$ is a true fully quantified Boolean formula $\}$.

TQBF is PSPACE-complete

Theorem. *TQBF is PSPACE-complete under polynomial time mapping reductions. (Stockmeyer, 1974)*

Proof TQBF is in PSPACE. We can write a recursive procedure TEST to test the membership. Let ϕ be a formula on n variables

- If ϕ contains no quantifier, it is TRUE or FALSE. Output "yes" if it is TRUE and "no" otherwise.
- Let Qy be the leftmost quantifier.
- If $Q = \exists$, output "yes" if TEST outputs "yes" on **either** $\phi(y = 0)$ **or** $\phi(y = 1)$ and "no" otherwise.
- If $Q = \forall$, output "yes" if TEST outputs "yes" on **both** $\phi(y = 0)$ **and** $\phi(y = 1)$ and "no" otherwise.

TQBF is PSPACE-complete (cont'd)

This procedure has the recursion depth of n. A quadratic space Turing machine can implement it.

For the "hardness" let $L \in \mathrm{PSPACE}$ via a Turing machine M that uses $p(n)$ space for a polynomial p. Let x be a string of length n, $n > 0$, whose membership in L we are testing.

Introduce the concept of $CANYIELD$, where each configuration is encoded as a binary string of length $q(n)$ for some polynomial $q(n) = O(p(n))$. Let C_0 and C_f be the unique initial and accepting configuration of M on x. For a vector of boolean variables $X = (X^{(1)}, ..., X^{(N)})$, write $(\exists X)$ for $(\exists X^{(1)})...(\exists X^{(N)})$. Also write $X = Y$ for $\bigwedge_{i=1}^{N}((X^{(i)} \wedge Y^{(i)}) \vee (\overline{X}^{(i)} \wedge \overline{Y}^{(i)}))$; this also works if $X^{(i)}$ or $Y^{(i)}$ is 0 or 1. We will recursively expand the following expression into a QBF:

$$CANYIELD(C_0, C_f, q(n)) \quad \text{(i.e., in } \leq 2^{q(n)} \text{ steps)}$$

TQBF is PSPACE-complete (cont'd)

For every $t > 0$, $C, D, |C| = |D| = q(n)$, $CANYIELD(C, D, t) =$

$$(\exists X, |X| = q(n))\,(\forall Y, Z, |U| = |Z| = q(n))$$
$$[((Y = C \wedge Z = X) \vee (Y = X \wedge Z = D)) \Rightarrow CANYIELD(Y, Z, t-1))]$$

If $t = 0$, then for every $C, D, |C| = |D| = q(n)$, $CANYIELD(C, D, 0)$ amounts to:

C yields D via M's transition function

By combining these we get a big formula for $x \in L$,

$$(\exists X_1, |X_1| = q(n))\,(\forall Y_1, Z_1, |Y_1| = |Z_1| = q(n))$$
$$(\exists X_2, |X_2| = q(n))\,(\forall Y_2, Z_2, |Y_2| = |Z_2| = q(n))$$
$$\cdots\,(\exists X_{q(n)}, |X_2| = q(n))\,(\forall Y_{q(n)}, Z_2, |Y_2| = |Z_2| = q(n))\,\phi,$$

where ϕ is a formula of variables corresponding to the bits of X_i's, Y_i's, and Z_i's, and constants 0, 1 deriving from C_0 and C_f.

The reduction can be computed in polynomial time in n. ∎

Variations of TQBF

In the above proof we can stick in any (within a polynomial bound) number of dummy variables. Also the formula ϕ at the end can be converted to a 3CNF by adding new variables. So, we can argue that adding any of the following restrictions, $TQBF$ is still PSPACE-complete: for PSPACE:

- **The formula is in prenex normal form**
- **The quantifiers are alternating**
- **The quantifiers start from an $\exists$**
- **The quantifiers end with an $\exists$**
- **The formula with the quantifiers eliminated is a 3CNF formula**

Formula Game

Suppose ϕ is a Boolean formula over $x_1, \ldots, x_k$ without quantifiers. Consider the game played by two players in which they take turns, starting from Player 1, in assigning values to the variables $x_1, x_2, \cdots, x_k$ in this order. Player 1 wins if the formula evaluates to TRUE for the assignment and Player 2 wins otherwise.

Define $FORMULA\text{-}GAME = \{\phi \mid \text{Player 1 can always win for } \phi\}$.

Then this problem is PSPACE-complete. Why? The condition $\phi \in FORMULA\text{-}GAME$ is rewritten as:

$$(\exists x_1)(\forall x_2) \cdots (Q_k x_k)\phi,$$

where the quantifiers alternate. This is essentially a variation of $TQBF$.

General Geography

Given a directed graph $G = (V, E)$ with a specified node s, consider the following game played by two players: Initially, $x = s$ and $W = \{s\}$. The players take turns, beginning with the move by Player 1, in selecting a node $u \notin W$ such that $(x, u) \in E$; u is then added to W. A player wins if the other cannot make a move. This is called **General Geography**.

Define $GG = \{(G, s) \mid$ Player 1 has a winning strategy in general geography $\}$.

General Geography is PSPACE-complete

$GG \in \mathrm{PSPACE}$. Let (G, s) be an instance of GG of some n nodes. Consider the following procedure **Search**:

- Input is (V, E, x, W, b), $x \in V$, $W \subseteq V$, $b \in \{1, 2\}$.
  $b = 1, 2$ means that it is the 1st, 2nd player's turn to move.
- Find the nodes $u_1, \ldots, u_k \notin W$ to which there is an edge from x.
- If $k = 0$, output “no” if $b = 1$ and “yes” if $b = 2$
- If $b = 1$, output “yes” if Search outputs “yes” on $(V, E, u_i, W \cup x, 2)$ **for some** i and ”no” otherwise.
- If $b = 2$, output “yes” if Search outputs “yes” on $(V, E, u_i, W \cup x, 1)$ **for all** i and ”no” otherwise.

GG is PSPACE-complete (cont'd)

The procedure outputs "yes" on $(V, E, s, \emptyset, 1)$ if and only if $(G, s) \in GG$. The recursion depth is at most $|V|$, so a polynomial space machine can execute it.

General Geography is PSPACE-Complete

Consider the version of $FORMULA$-$GAME$ with the formula in 3CNF, with an odd number of variables: $\exists x_1 \forall x_2 ... \exists x_{2k-1} \forall x_{2k} \exists x_{2k+1}\, \phi_{3CNF}$. Reduce the formula to an instance of GG as follows: