PSPACE = set of languages solvable using polynomial number of cells, (input is n cells, the TM decides the language in O(nk) cells for some constant k), on a single tape, deterministic Turing machine

PSPACE = NPSPACE

Savich’s Theorem. NSPACE(f(n)) is in SPACE(f2(n)).

TQBF is PSPACE-complete. (A logical formula with n variables, each variable can be assigned T or F, each variable is bounded by a quantifier.)

A lot of games are PSPACE-complete.

A game strategy is (there exists a move, for all counter moves by my opponent, there exists a move for me, for all counter moves, …. (I win the game)).

Consider the following game on a directed graph. Two players choose where to move a token on the graph. There is a start vertex, and the token has to move to a vertex following the edges. Each player takes a turn moving the token. The token can never revisit a vertex. A player wins if the opponent can’t move the token. (No out edges from that vertex, or all vertices we can get to from that vertex were already visited.)

Is this problem even in PSPACE?

TQBF <=P This game problem

This game problem is PSPACE-complete.

Log space: SPACE(log n)

L = the set of languages decidable by a deterministic Turing machine, input uses n cells, the TM uses log n work cells.

We use a 2-tape TM. Tape 1 stores the input, n cells. It is read only, and the head cannot leave the portion of the tape with the input. Tape 2 is only O(log n) cells, and it is read/write.

Ex: the language {akbk | k >= 0} is in L.

Count the # of a’s. (Represent the number in binary, that is O(log n) cells.)

Count the # of b’s. (Represent the number in binary, that is O(log n) cells.)

Check that the numbers are the same.

Ex: the language {wwR | w in Sigma\*}. Is this in L?

Have index variables i=1 and j=n, represented in binary. Loop incrementing i and decrementing j until they match. Check the input cells at locations i and j. If they do not match, reject. If the loop ends accept. Total space O(log n).

NL = nondeterministic machines that use logspace.

Use Savich’s theorem to show that NSPACE(f(n)) is in SPACE(f2(n)) as long as f(n) >= log n. (Savich’s theorem uses the configuration (state) of a TM. But our TM includes an input string that is n cells long. We can’t use the input string in the configuration.)

NL-complete