Chapter 8: Space complexity

Space Complexity

- Let M be a deterministic Turing machine that halts on all inputs. The space complexity of M is the function $f: N \to N$, where $f(n)$ is the maximum number of tape cells that M scans on any input of length n. If the space complexity of M is $f(n)$, we also say that M runs in space $f(n)$.

Some definitions:

- SPACE(f(n))={L| L is a language that is decided by a TM M in tape space O(f(n))}
- NSPACE(f(n))={L| L is a language that is decided by a NTM N in tape space O(f(n))}
- $PSPACE = \bigcup_{k=1}^{\infty} SPACE(n^k)$

SAT $\in$ PSPACE

- The following is a linear space algorithm to decide SAT:
- M= “On input <ϕ>, where ϕ is a boolean formula:

1. For each truth assignment to the variables x1, …, xm of ϕ:
2. Evaluate ϕ on that truth assignment.
3. If ϕ ever evaluated to 1, accept; if not, reject.”

Savitch's Theorem:

$SPACE(f^2(n)) \supseteq NSPACE(f(n))$

Let L be decided by NTM N in NSPACE(f(n))

Define ***possible***(c1, c2, m) to be true iff N can transit from configuration c1 to c2 within *m* steps.

PSPACE completeness

Language L is PSPACE-complete iff:

1. L is in PSPACE
2. Every language *L'* in PSPACE is reducible to *L* via a polynomial time reduction

Theorem

If L is PSPACE-complete and L is in P,
then P = PSPACE

TQBF = {ϕ: ϕ is a true totally quantified boolean formula}

E.g. $\phi = \forall x_1 \exists x_2$ (x_1 **OR** x_2) is in TQBF

$\phi = \forall x_1 \exists x_2$ (x_1 **AND** x_2) is not in TQBF

TQBF is PSPACE-complete

1. <u>TQBF is in PSPACE:</u>

T="On input <ϕ>, a fully quantified boolean formula:

1. If ϕ contains no quantifiers, then it is an expression with only constants, so evaluate ϕ and accept if it is true; otherwise, reject.
2. If ϕ equals $\exists x\ \boldsymbol{\theta}$, recursively call T on $\boldsymbol{\theta}$, first with 0 substituted for x and then with 1 substituted for x. If either result is accept, then accept; otherwise, reject.
3. If ϕ equals $\forall x\ \boldsymbol{\theta}$, recursively call T on $\boldsymbol{\theta}$, first with 0 substituted for x and then with 1 substituted for x. If both results are accept, then accept; otherwise, reject.

L Log space and Non-deterministic Log space

- L is the class of languages that are decidable in logarithmic space on a deterministic Turing machine. In other words, L = SPACE(log(n)) .
- NL is the class languages that are decidable in logarithmic space on a nondeterministic Turing machine. In other words, NL = NSPACE(log(n)).

NL-complete

L is NL-complete, iff:

1. L is in NL
2. Every language L’ in NL is log-space reducible to L

PATH

PATH = {(G, s, t)| G is a directed graph with a path that starts at vertex s and ends at vertex t}

Theorem:

PATH is NL-complete

Reference

- www.cs.unm.edu/~gemmell/500.html by Dr. Peter Gemmell.