Recursion and complexity

Isabel Oitavem

CMA and DM, FCT-UNL

This work was partially supported by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the project UID/MAT/00297/2013 (Centro de Matemática e Aplicações).

centro de matemática e aplicações FCT.UNL

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

NOVA id FCT Associação para a Inovação e Desenvolvimento da FCT

Models of computation: Turing machines

A deterministic Turing machine (TM) with k tapes is a four-tuple

$$M =< Q, \Sigma, \delta, q_0 >$$

where
Q is a finite set of states;
Σ is the tape alphabet;
δ is the transition function,

$$\delta : Q \times \Sigma^k \to Q \times \Sigma^k \times \{L, N, R\}^k;$$

$q_0 \in Q$ is the initial state.

Models of computation: Turing machines

δ is the transition function,

$$\delta : Q \times \Sigma^k \to Q \times \Sigma^k \times \{L, N, R\}^k.$$

- Situations in which the transition function is undefined indicate that the computation must stop;
- Otherwise the result of the transition function is interpreted as follows:
  - The first component is the new state;
  - The second component is the tuple of symbols to be written on the scanned cells of the k tapes;
  - The third component specifies de moves of the tape heads.

Models of computation: Turing machines

- A machine M starts operating on an input word w with:
  - The first tape (input tape) containing w, and a fixed symbol called *blank* (B) in the remaining cells;
  - Every cell of the other tapes also contains B;
  - The internal state is the initial state q_0.
- Then M proceeds by applying the transition function δ as long as possible.
- If after a sequence of steps the machine stops, the output is the word which appears in the k-th tape (output tape) of the machine.

Models of computation: Turing machines

Given a k-tape TM M, a configuration of M, also called an instantaneous description, or a snapshot, is a $k+1$ tuple

$$(q, x_1, \cdots, x_k)$$

where

- q is the current state of M;
- $x_j \in \Sigma^* \# \Sigma^*$, for $1 \leq j \leq k$,
  $\# \notin \Sigma$ marks the position of the tape head (the head scans the symbol immediately at the right of the "$\#$");
- All symbols in the infinite tape not appearing in x_j, for $1 \leq j \leq k$, are assumed to be B.

The initial configuration of a machine M on an input w is $(q_0, \#w, \#, \cdots, \#)$.

Models of computation: Turing machines

Given a k-tape TM M, the computation of M on w is a sequence of configurations of M which

- starts with the initial configuration of M on w;
- each step from a configuration to the next obeys to the transition function of M;
- if it ends, it ends in a configuration in which no more steps can be performed.

A total function f is computed by a TM M iff, for every input w, the computation of M on w is a finite sequence of configurations such that the last configuration corresponds to $f(w)$.

Models of computation

Church's thesis

- Every "efective computation" can be programmed to run on a Turing machine.

Church's thesis cannot be "proven" because concepts such as "efective process" and "algorithms" are not part of any branch of mathematics.

Introduction to complexity theory

The classification of the complexity problems should not depend on a particular computational model but rather should measure the intrinsic difficulty of the problem.

The basic model of computation for our study is the multitape TM, but the measurement mechanisms are essentially machine-independent.

The two most important measures, and the two most common measures, are time, the time it takes a program to execute, and space, the amount of storage used during a computation.

Complexity classes and complexity measures

TIME complexity

- Let M be an (on-line) TM, and let T be a function over $\mathbb{N}$. M is a $T(n)$ time-bounded TM if for every input of length n, M makes at most $T(n)$ moves before halting.
- A function f (over $\mathbb{N}$) that is computed by a deterministic $T(n)$ time-bounded TM M has time complexity $T(n)$.

By convention, the time it takes to read the input is counted (this takes $n+1$ steps). So when we say a computation has time complexity $T(n)$, we really mean $\max(n+1, \lceil T(n) \rceil)$.

- *FDTIME*$(T(n))$ is the set of all functions having time complexity $T(n)$.

Complexity classes and complexity measures

SPACE complexity

- An off-line TM is a multitape TM with a separate read only input tape.
- Let M be an off-line multitape TM and let S be a function over $\mathbb{N}$.
  M is a $S(n)$ space-bounded TM if, for every word of length n, M scans at most $S(n)$ cells over all storage tapes.
- A function f (over $\mathbb{N}$) that is computed by a deterministic $S(n)$ space-bounded TM has space complexity $S(n)$.

Every TM is permitted to use at least one work cell, so by space complexity $S(n)$ we mean $\max(1, \lceil S(n) \rceil)$.

- *FDSPACE*$(S(n))$ is the set of all functions having space complexity $S(n)$.

Complexity classes and complexity measures

We are going to focus on two complexity classes:

- FPtime $= \cup_k FDTIME(n^k)$
- FPspace $= \cup_k FDSPACE(n^k)$

Complexity classes and complexity measures

We are going to focus on two complexity classes:

- FPtime $= \cup_k FDTIME(n^k)$
- FPspace $= \cup_k FDSPACE(n^k)$

Known: FPtime $\subseteq$ FPspace

OPEN: FPtime $=$ FPspace ?

Exercises

1. Describe a TM that works for ever (never terminates) on every input.
2. Describe two different TM computing the function $\pi_1^1 : \{0,1\}^* \to \{0,1\}^*$ such that $\pi_1^1(w) = w$.
3. Consider the function $S_1 : \{0,1\}^* \to \{0,1\}^*$ such that $S_1(x) = x1$.
   3.1 Define a TM for S_1.
   3.2 Is the described machine a $T(n)$ time-bounded TM? Which function T may be considered as time bound?
   3.3 Is the described machine a $S(n)$ space-bounded TM? Which function S may be considered as space bound?