Undergrad Complexity Theory: Notes 2

Ben Chaplin

Contents

1 Background

1.1 Languages

Definition. A **language L** over an alphabet Σ is a subset of strings $L \subseteq \Sigma^*$.

Example. PRIMES $= \{\langle x \rangle : x \in \mathbb{N}, x \text{ is prime}\}$, equivalent to the decision problem $\{0,1\}^* \to \{\text{no, yes}\}$.

1.2 Algorithms

What is an algorithm? We can define the concept using programming languages, or better, models of computation. Turing machines are a good choice as they're easy to formalize. Furthermore:

Church-Turing Thesis. Any real-world algorithm can be simulated by Turing Machines.

However, this is a statement about computability. In terms of complexity theory, it's important to note that an algorithm running in time T in C-like pseudocode, it can compiled to a Turing machine running in time roughly T^4.

2 Turing Machines

2.1 Model

Official model of a Turing Machine:

- one tape
- two-way infinite

Roughly, the tape holds symbols from an alphabet and reads/writes with a "head" pointer. The source code tells us how to move and write.

2.2 Formal definition

Definition. A **Turing machine** $\mathcal{M}$ is a 5-tuple $M = (\Sigma, Q, q_0, F, \delta)$, where

- Σ finite set of symbols, the input alphabet (blank symbol is here)
- Q is a finite set of states
- q_0 is the initial state
- $F \subseteq Q$ is the set of accepting states
- $\delta : (Q) \times \Sigma \nrightarrow Q \times \Sigma \times \{\text{left, right}\}$ is the transition function. If δ is not defined on the current tape symbol, the machine halts.

Conventions: $\Sigma = \{0, 1\}$ where 0 is the blank symbol, $\delta(q, a)$ is undefined for all $q \in F$ and $a \in \Sigma$ (this is how we define halting).

We still need to define how the machine "runs."

Definition. A **configuration** is a word ypx where $x, y \in \Sigma^*$ and $p \in Q$. This means that the TM head is in state p, positioned at x and, the tape inscription is yx.

Next, we describe the following notation for a one-step relation:

$$ypx \vdash^{1}_{\mathcal{M}} y'qx'$$

As the TM $\mathcal{M}$ takes one step, sending configuration ypx to $y'qx'$.

Lemma 1. *The relation $\vdash^{t}_{\mathcal{M}}$ is primitive recursive, uniformly on t.*