Topics covered

- Review of complexity

Turing machines (TMs)

Formally define our notion computability. They consist of:

- an arbitrarily large tape,
- a finite tape alphabet, and
- a finite state automaton (fsa).

If the fsa is non-deterministic, then the TM is called a *non-deterministic TM* (NTM). Otherwise the TM is called a *deterministic TM* (DTM).

Operations:

- reading the current tape symbol,
- overwriting the current tape symbol,
- moving the tape head left or right, and
- changing state (possibly non-deterministically).

Computability, decidability and TMs

A function $\mathcal{F}$ from Σ^* to Σ^* is *computable* iff there exists some TM that can be used to *compute* $\mathcal{F}$ as follows:

1. Place an arbitrary element w of the domain of $\mathcal{F}$ on the tape and position the read head at the first symbol.
2. Start executing the TM.
3. The TM halts with the tape containing $\mathcal{F}(w)$.

A decision problem $\mathcal{P}$ is *decidable* iff it corresponds to a computable Boolean function; that is, if there exists a computable function with a domain consisting of all instances of the problem (suitably encoded) that computes *true* in cases where an instance is true, and *false* otherwise.

Complexity

Complexity theory is a study of the *difficulty* of decidable problems and computable functions, where difficulty is measured by some resource of interest, often the time and space needed by TMs to reason about instances of the problems or to compute functions.

Complexity (cont'd)

Let g be a function on the positive integers. The following are then defined.

- DTIME(g) $\equiv \{\mathcal{F} : \mathcal{F}$ is a computable function and requires at most $g(w)$ steps to compute on a DTM for a given instance $w\}$
- NTIME(g) $\equiv \{\mathcal{F} : \mathcal{F}$ is a computable function and requires at most $g(w)$ steps to compute on a NTM for a given instance $w\}$
- DSPACE(g) $\equiv \{\mathcal{F} : \mathcal{F}$ is a computable function and requires at most $g(w)$ space to compute on a DTM for a given instance $w\}$
- NSPACE(g) $\equiv \{\mathcal{F} : \mathcal{F}$ is a computable function and requires at most $g(w)$ space to compute on a NTM for a given instance $w\}$

Complexity classes

Let $\mathcal{G}$ be a set of functions on positive integers. We write the following to denote *complexity classes*.

- $\mathrm{DTIME}(\mathcal{G}) \equiv \bigcup_{g \in \mathcal{G}} \mathrm{DTIME}(g)$
- $\mathrm{NTIME}(\mathcal{G}) \equiv \bigcup_{g \in \mathcal{G}} \mathrm{NTIME}(g)$
- $\mathrm{DSPACE}(\mathcal{G}) \equiv \bigcup_{g \in \mathcal{G}} \mathrm{DSPACE}(g)$
- $\mathrm{NSPACE}(\mathcal{G}) \equiv \bigcup_{g \in \mathcal{G}} \mathrm{NSPACE}(g)$

Common name for complexity classes

Let *Poly* denote the set of polynomials on positive integers, and let *Exp* denote the set of functions $\{c^n : c \geq 1\}$. Some common names for some complexity classes:

$$\begin{array}{lcl}
\text{DTIME}(\textit{Poly}) & \equiv & \text{PTIME (or simply P)} \\
\text{NTIME}(\textit{Poly}) & \equiv & \text{NP} \\
\text{DSPACE}(\textit{Poly}) & \equiv & \text{PSPACE} \\
\text{NSPACE}(\textit{Poly}) & \equiv & \text{NPSPACE} \\
\text{DTIME}(\textit{Exp}) & \equiv & \text{DEXPTIME} \\
\text{NTIME}(\textit{Exp}) & \equiv & \text{NEXPTIME}
\end{array}$$

Known results on complexity classes

$$\text{PTIME} \subseteq \text{NP} \subseteq \text{PSPACE} \equiv \text{NPSPACE}$$

$$\text{PSPACE} \subseteq \text{DEXPTIME} \subseteq \text{NEXPTIME}$$

$$\text{PTIME} \subset \text{DEXPTIME}$$

$$\text{NP} \subset \text{NEXPTIME}$$

Reducibility and completeness

Let $\mathcal{C}$ and $\mathcal{C}'$ denote complexity classes. A decidable problem $\mathcal{P}$ in $\mathcal{C}$ *is complete in* $\mathcal{C}$ *by* $\mathcal{C}'$ *reductions* iff, for any other problem $\mathcal{P}'$ in $\mathcal{C}$, there is a computable function in $\mathcal{C}'$ that *reduces* any instance of $\mathcal{P}'$ to an instance of $\mathcal{P}$. We call $\mathcal{C}'$ the *type of reduction*. (A reduction problem is the problem of translating instances of one problem to instances of another.)

Some common conventions.

- NP-completeness: $\mathcal{C}$ is NP and $\mathcal{C}'$ is PTIME.
- PSPACE-completeness: $\mathcal{C}$ is PSPACE and $\mathcal{C}'$ is PTIME.
- DEXPTIME-completeness: $\mathcal{C}$ is DEXPTIME and $\mathcal{C}'$ is PTIME.