Chapter 6 - Oracles and the Limits of Diagonalization

- Define oracle Turing machines (OTMs).
- OTMs are a powerful and often used tool in complexity theory.
- Show that that there is an oracle A for which the classes $\mathbf{P}^A$ and $\mathbf{NP}^A$ are identical.
- Show that there is an oracle B for which the classes $\mathbf{P}^B$ and $\mathbf{NP}^B$ are different.
- Diagonalization is oblivious to oracles.
- Conclude that straightforward applications of diagonalization will not yield proofs for $\mathbf{P} \neq \mathbf{NP}$.

Oracle Turing machines (OTMs)

Definition 6.1

- *An oracle Turing machine (OTM) $M^?$ is a TM (deterministic or nondeterministic) with a special tape, called the oracle tape, and three special states $q_?, q_{\mathrm{yes}}, q_{\mathrm{no}}$.*
- *For an arbitrary language $A \subseteq \{0,1\}^*$ we denote by M^A the OTM $M^?$ with access to oracle A.*
- *If M^A is in a state different from $q_?$, then the next step of M^A is defined as for usual TMs.*
- *If M^A is in state $q_?$ and the content of the oracle tape is z (ignoring the start symbol $\triangleright$), then M^A in one step goes into state q_{yes} if $z \in A$ and into state q_{no} if $z \notin A$. The contents of all tapes remain unchanged and the tape heads do not move.*

Running times of OTMs

Definition 6.2

Let $M^?$ be an OTM such that for all languages A the OTM M^A halts on all inputs.

- *The running time or time complexity of $M^?$ is the function $f : \mathbb{N} \rightarrow \mathbb{N}$, where $f(n)$ is the maximum number of steps that any OTM $M^A, A \subseteq \{0,1\}^*$ uses on any input of length n.*
- *The space complexity of $M^?$ is the function $f : \mathbb{N} \rightarrow \mathbb{N}$, where $f(n)$ is the maximum number of tape cells that any OTM $M^A, A \subseteq \{0,1\}^*$ scans on any input of length n.*

Remark

Observe that the oracle's reply on each query is obtained in a single step!

Oracles classes or relativized worlds

Definition 6.3

For every $A \subseteq \{0,1\}^*$ *define*

$$\mathbf{P}^A := \{L \mid L \textit{ can be decided by a deterministic polynomial time OTM } M^? \textit{ with oracle } A.\}$$

$$\mathbf{NP}^A := \{L \mid L \textit{ can be decided by a nondeterministic polynomial time OTM } M^? \textit{ with oracle } A.\}$$

$$\mathbf{PSPACE}^A := \{L \mid L \textit{ can be decided by a deterministic polynomial space OTM } M^? \textit{ with oracle } A.\}$$

Examples

Polynomial time reductions and oracles

Let A, B be languages with $B \leq_p A$. Then

- $B \in \mathbf{P}^A$ and
- $B \in \mathbf{P}^{\overline{A}}$.

SAT oracles

- $\mathbf{NP} \subseteq \mathbf{P}^{SAT}$
- co-$\mathbf{NP} \subseteq \mathbf{P}^{SAT}$

Examples

Equivalence and minimality of Boolean formulas

- We call two Boolean formulas *equivalent*, if
  1. they have the same set of variables and
  2. they are true on the same set of assignments to those variables.
- A Boolean formula is called *minimal* if no shorter Boolean formula is equivalent to it.

Two languages

$\mathrm{MF} := \{\langle\phi\rangle \mid \phi$ is a minimal Boolean formula$\}$
$\overline{\mathrm{MF}} = \{\langle\phi\rangle \mid \phi$ is a not minimal Boolean formula$\}$

Observation

$\overline{\mathrm{MF}} \in \mathbf{NP}^{SAT}$

A nondetermistic oracle TM for $\overline{\mathrm{MF}}$

$N^{SAT}_{\overline{\mathrm{MF}}} =$ "On input ϕ:

1. Nondeterministically guess a Boolean formula ψ with length shorter than the length of ϕ.
2. Compute $\neg(\phi \Leftrightarrow \psi)$, write this formula on the oracle tape, and go to state $q_?$.
3. From state q_{no} go to state q_{accept}, from state q_{yes} go to state q_{reject}."

The limits of diagonalization

Theorem 6.4

1. *An oracle A exists with* $\mathbf{P}^A \neq \mathbf{NP}^A$.
2. *An oracle B exists with* $\mathbf{P}^B = \mathbf{NP}^B$.

Proof for existence of B

- Set $B :=$ TQBF.

$\Rightarrow$ $\mathbf{NP}^{\mathrm{TQBF}} \subseteq \mathbf{NPSPACE} \subseteq \mathbf{PSPACE} \subseteq \mathbf{P}^{\mathrm{TQBF}}$

Proof for existence of A - sketch

- For A arbitrary define

$$L_A := \{1^n \mid n \in \mathbb{N}, \exists x \in A \text{ with } |x| = n\}.$$

- For all $A : L_A \in \mathbf{NP}^A$.
- Construct A, such that for every polynomial time deterministic OTM $M_i^?$ there is a number $n_i \in \mathbb{N}$ with

$$1^{n_i} \in L(M_i^A) \Leftrightarrow 1^{n_i} \notin L_A.$$

$\Rightarrow$ $L_A \notin \mathbf{P}^A$.

Construction of A (1)

- Let $M_1^?, M_2^?, \ldots$ be an enumeration of all polynomial time deterministic OTMs.
- Choose e_i such that the running time of $M_i^?$ is bounded by n^{e_i} (note that the running time of an OTM is defined independently from any specific oracle)
- Inductively construct finite sets $A_0, \tilde{A}_0, A_1, \tilde{A}_1, \ldots$ satisfying
  1. $A_j \subset A_i$ and $\tilde{A}_j \subset \tilde{A}_i$ for all $j < i$, and
  2. $A_i \cap \tilde{A}_i = \emptyset$ for all i.
- Set $A_0 = \tilde{A}_0 = \emptyset$.
- Assume $A_j, \tilde{A}_j, j = 0, \ldots, i-1$ have already been defined properly.
- Set

$$n_i := \min\{n \mid 2^n > n^{e_i} \text{ and } n > |x| \text{ for all } x \in A_{i-1} \cup \tilde{A}_{i-1}\}$$

Construction of A (2)

- To define A_i and $\tilde{A}_i$ we simulate $M_i^?$ with input 1^{n_i}. The oracle queries of $M_i^?$ are answered as follows:
    1. Set $X_i := \emptyset$.
    2. If an oracle query x is in A_{i-1}, go to state q_{yes}.
    3. If an oracle query x is in $\tilde{A}_{i-1}$, go to state q_{no}.
    4. If an oracle query x is neither in A_{i-1} nor in $\tilde{A}_{i-1}$, go to q_{no} and set $X_i := X_i \cup \{x\}$.
- If $M_i^?$ accepts 1^{n_i}, we set

$$A_i := A_{i-1} \text{ and } \tilde{A}_i := \tilde{A}_{i-1} \cup \{0,1\}^{n_i}.$$

- If $M_i^?$ rejects 1^{n_i}, choose $w_i \in \{0,1\}^{n_i} \setminus X_i$ and set

$$A_i := A_{i-1} \cup \{w_i\} \text{ and } \tilde{A}_i := \tilde{A}_{i-1} \cup X_i.$$

- Finally, set

$$A := \bigcup_{i \geq 1} A_i.$$