Problem Set 7
Exercises on Oracles & Circuits
CSCI 6114 Fall 2023

Joshua A. Grochow

Released: October 17, 2023
Due: Monday October 23, 2023

Exercises

1. E is the class of languages L such that L is decided by an algorithm running in time $2^{O(n)}$. NE is defined similarly but allowing nondeterministic algorithms. Analogous to NP, NE has an equivalent characterization in terms of verifiers, but here we must be careful about the length of the witness and the runtime: L is in NE iff there is a deterministic verifier V and an integer k such that

$$x \in \text{Ł} \iff (\exists w)[|w| \leq 2^{O(n)} \text{ and } V(x,w) = 1],$$

and $V(x,w)$ runs in time $2^{O(|x|)}$ (if we allowed V to run in time $2^{O(|x|+|w|)}$, then that runtime would be doubly-exponential relative to x ($\sim 2^{2^{|x|}}$), which isn't what we want).

Show that if $\mathsf{NE} \neq \mathsf{E}$, then there is a sparse language in $\mathsf{NP} - \mathsf{P}$. *Hint:* If L is in NE, define $L_{pad} := \{(x, 1^{2^{|x|}}) | x \in L\}$. What complexity class is L_{pad} in? (This is called the *padding* technique.)

2. An N-variable (Boolean) *decision tree* is a binary rooted tree T, with each vertex labeled by an index $i \in \{1, \ldots, N\}$, and the two children edges of a node labeled 0 and 1. Each leaf is labeled with an element of $\{0,1\}$. T compute a Boolean function $\{0,1\}^N \to \{0,1\}$ as follows. On input x, if an internal vertex is labeled i, then the function reads the variable x_i; if $x_i = 0$ it proceeds down the edge labeled 0, and if $x_i = 1$ it proceeds down the edge labeled 1. When it reaches a leaf,

it outputs the label of that leaf. The *depth* of a decision tree is the longest length of a path from the root to a leaf.

(a) Show that if f is a Boolean function computed by a decision tree of depth d, then it can be written as a DNF (OR of ANDs of variables and their negations) where the terms are ANDs of at most d literals (literal=variable or negated variable).

(b) Show that if f is a Boolean function computed by a decision tree of depth d, then it can be written as a CNF (AND of ORs) where the clauses are ORs of at most d literals.
(Foreshadowing: the combination of the two parts of this exercise is analogous to the statement $\mathsf{P} \subseteq \mathsf{NP} \cap \mathsf{coNP}$. We'll make some of this precise in subsequent exercises.)

3. (a) Suppose $M^{\square}$ is a polynomial-time oracle Turing machine. Consider $M^A(1^n)$ as a function $\{0,1\}^{2^{n^q+1}-1} \to \{0,1\}$. If $M^A(1^n)$ queries strings of length at most n^q and runs in time n^t, show that this function is computed by a decision tree of depth n^t. Let $N \sim 2^{n^q}$; show the function we were considering is an N-bit function computed by a decision tree of depth $poly(\log N)$ (thus say that "Relativized P is the exponentially blown-up version of log-depth decision trees").

   (b) Suppose $M^{\square}$ is a *non-deterministic* polynomial-time oracle Turing machine. Consider $M^A(1^n)$ as a Boolean function as in the previous part. Suppose $M^A(1^n)$ uses n^k nondeterministic bits and runs in $O(n^t)$ time. Show that this Boolean function can be computed as an OR of $2^{O(n^k)}$-many decision trees, each of height at most $O(n^t)$. Let $N = 2^{n^t}$; show the Boolean function we're considering is computable as an OR of N decision trees each of height at most $O(\log N)$ (note that we must have $k \leq t$).

4. (a) Prove that the n-bit OR function cannot computed by a decision tree of depth $poly(\log n)$.

   (b) Use part (a) in combination with the previous exercise to give an alternative construction of an oracle A such that $\mathsf{P}^A \neq \mathsf{NP}^A$. (Well, really it is the same construction, but it is an alternative proof.)

5. Analogous to Exercise 3, for each of the following relativizable complexity classes, what is the corresponding circuit class (of which the relativizable class is the exponentially blown-up version): $\Sigma_k\mathsf{P}, \Pi_k\mathsf{P}, \mathsf{PSPACE}$.

What circuit lower bound would give an oracle D such that $\mathsf{PSPACE}^D \neq \mathsf{PH}^D$?

At-home exercises

6. (a) Prove that the N-bit PARITY function (=0 iff the number of input bits that are 1 is even) cannot be computed as an OR of $N^{poly(\log N)}$-many decision trees, each of height at most $poly(\log N)$.

   (b) Use part (a) in combination with Exercise 3 to give a(n alternative proof of the same) construction of an oracle B such that $\mathsf{PSPACE}^B \neq \mathsf{NP}^B$.

7. Now we show a converse to Exercise 3, but for this we need the "right" notion of uniformity (note that decision trees are a nonuniform model of computation, but oracle TMs are still uniform). Let $T = (T_i)_{i=1,2,3,\ldots}$ be a family of decision trees, with T_N computing an N-variable Boolean function for all N. For a decision tree T_N, and define the i-th gate on input x to be either j—if, on input x, after i steps, the next variable to be queried is x_j—or $(\perp, b)$ if, on input x, after i steps, a leaf is reached that outputs $b \in \{0,1\}$. Say T is a *poly(log)-time uniform* family if there is a Turing machine A, such that $A(N, x, i)$ outputs the i-th gate of T_N on input x in time $poly(\log N)$.

   Because we are talking about sub-linear runtimes here, we need to be a little careful about our notion of Turing machine. The machine has a read-only input tape of length n, an "input address" tape of length $\lceil \log n \rceil$, and a constant number of work tapes, and an output tape. The address tape, work tapes, and output tape function as usual (each has its own tape head, etc.). However, there is no head on the input tape; instead, at a given time step, if the input address tape contains the binary expansion of the number i, then the machine has access to the i-th bit of the input.

   Show that if T is a poly(log)-time uniform family of decision trees of poly(log) depth, then there is a polynomial-time oracle Turing machine $M^{\square}$ such that M^A, on inputs of length n, only queries the oracle on strings of length n.

Resources

- Ker-I Ko has an expository survey giving many examples of the connection between circuit lower bounds and oracles.
- Jukna's book on Boolean functions is an excellent resource on constant-depth circuits (Chapters 11 and 12) and decision trees (Chapter 14), among *many* other things!
- In fact, the existence of sparse languages in $\mathsf{NP} - \mathsf{P}$ is *equivalent* to $\mathsf{NE} \neq \mathsf{E}$, due to Hartmanis, Immerman, and Sewelson. In this way, looking at "larger" complexity classes can also be seen as using a microscope to look at more fine-grained structure of smaller complexity classes. They showed analogous results for $\mathsf{PSPACE} - \mathsf{NP}$ and $\mathsf{PSPACE} - \mathsf{P}$, and all these results relativize.
- Dlogtime-uniformity (of which we use the poly(log) version here) was defined and discussed in Barrington, Immerman, and Straubing, *J. Comput. Syst. Sci.*, 1990.