6.841 Advanced Complexity Theory March 7, 2007

Lecture 9

Lecturer: Madhu Sudan *Scribe: Jaime Quinonez*

1 Overview

We will introduce a new model of computation, the debate, that uses alternation. Whereas before alternation gave new insight into time and space complexity of languages, we will now see that allowing alternation is itself an interesting computational phenomenon. We will also introduce the Infinite Hierarchy Assumption (IHA), which informally states that allowing one more alternation allows you to solve more problems. This assumption states that the polynomial hierarchy is infinitely large and that no two classes in the hierarchy are identical. We will show that if you take IHA to be true, you get the following result, known as the Karp-Lipton Theorem:

Theorem 1 *Karp-Lipton Theorem*

$$IHA \Rightarrow NP \not\subseteq P/_{poly}$$

2 Formalizing a Debate

Say that you have some statement x and two parties, Alice (A) and Bob (B) such that A believes x is true and B believes x is false. Each party can try to convince others that their belief is correct by broadcasting a public message to the other party. Unlike Communication Complexity, we allow each message to have arbitrary length, although we will later see that each message should have polynomial (in $|x|$) length in order to be meaningful. Each party is also allowed an unbounded amount of time to compute their messages. We can clearly see that you would never have one party send two consecutive messages without receiving a message in between, since the two messages sent could just be concatenated into one message, unlike Communication Complexity which required 1-bit messages. Thus, in this model, A sends a message a_1 trying to prove that x is true, then B sends a message a_2 in response trying to prove that x is false, then A sends a message a_3 in response trying to prove that x is true, and so on for a predetermined number of rounds.

Someone listening to the debate can then ask themselves, "Is x true?" Initially, we assume the listener is unable to compute this on their own. After listening to the debate, a listener can ask themselves, "Given $(x, a_1, a_2, \ldots, a_i)$, is x true?" The listeners can't compute the a_i on their own, but assuming that both parties A and B are doing the best they can in the debate, and so the a_i are optimal, the listener can use the a_i to prove whether or not x is true. This is similar to the polynomial-length witness for languages in NP. Assuming that $x \in L$ for a language $L \in NP$, there exists a witness w such that a polynomial time verifier can accept x and w as input and prove that x is in L. This definition is exactly a debate where only one message can be sent, and this message is sent by the party trying to prove x is true. Does allowing another round (allowing the other party to respond) change what can be computed? The key question we are considering is how many rounds should be necessary in the debate between A and B to convince a listener V about the truthfulness of x? As we will see, we don't know the answer, but we believe that allowing more messages to be sent in the debate does allow the listener to decide more languages.

Formally, a language L_{debate} that can be decided with a debate of i messages, starting with the party that wants to prove the input x is true, can be defined as:

Definition 2 *A debatable language*

$$\begin{aligned} L_{debate} = \{x \quad | \quad & \textit{there exists a verifier } V \textit{ such that} \\ & x \in L_{debate} \Rightarrow \exists a_1 \forall a_2 \exists a_3 \ldots Q_i a_i \quad s.t. \quad V(x, a_1, a_2, \ldots, a_i) = 1 \\ & x \notin L_{debate} \Rightarrow \exists a_1 \forall a_2 \exists a_3 \ldots Q_i a_i \quad s.t. \quad V(x, a_1, a_2, \ldots, a_i) = 0\} \end{aligned}$$

Here, $Q_i \subset \{\exists, \forall\}$ denotes the i^{th} quantifier.

One can see that since there are only i quantifiers, in an alternating sequence, L_{debate} can be decided by an ATM with i alternations, starting with the $\exists$ quantifier.

3 Classes of debatable languages

We can also define a class for a debatable language as follows:

Definition 3 *A class of debatable languages*

$$\Sigma_i^P = \{L_{debate} \mid L_{debate} \text{ can be debated with } i \text{ rounds of interaction and the party trying to prove } x \in L \text{ sends first message}\}$$

While we don't entirely know whether it makes a difference which party goes first, we can similarly define the class of debatable languages if the party trying to prove $x \notin L$ goes first.

Definition 4 *A class of debatable languages*

$$\Pi_i^P = \{L_{debate} \mid L_{debate} \text{ can be debated with } i \text{ rounds of interaction and the party trying to prove } x \notin L \text{ sends first message}\}$$

From these definitions, it is clear to see some obvious relations to previous complexity classes. Σ_0^P is the class of languages decidable by a polynomial time verifier listening to an empty debate, and thus $\Sigma_0^P = P$. Σ_1^P allows the party trying to prove $x \in L$ one message as input to the verifier, which serves as a witness to the verifier, so $\Sigma_1^P = NP$. Similarly, $\Pi_1^P = coNP$. Also note that since the verifier V has to run in polynomial time, it can only process the messages if each message has polynomial length. Thus, the only meaningful messages in this definition are those that have polynomial length.

There are a few other obvious yet highly useful properties about these classes.

Fact 5 *Facts about Σ_i^P and Π_i^P:*

- *Σ_i^P is the class of languages decidable in $ATIME[i, poly]$ starting with the $\exists$ state. This follows directly from the definition since there are only i quantifiers, starting with $\exists$, and so you only need to have i alternations between quantified states.*
- *$\Sigma_i^P \subseteq \Sigma_{i+1}^P$. If you have $i+1$ rounds in the debate, you can clearly always just ignore the last round and thus simulate only i rounds.*
- *$\Sigma_i^P \subseteq \Pi_{i+1}^P$. Similarly, Π_{i+1}^P starts with an $\forall$ message, then the next i messages start with an $\exists$ message and thus if you ignore the first message you have Σ_i^P.*
- *$\Pi_i^P \subseteq \Sigma_{i+1}^P$. The argument is the same as above.*
- *$PH = \cup_i \Sigma_i^P = \cup_i \Pi_i^P$. This equality just follows from the previous two properties.*

It is also helpful to define a complete problem for these classes, which we can do by simplifying the $PSPACE$-complete language $TQBF$ (which either stands for "True Quantified Boolean Formula" or "Totally Quantified Boolean Formula", depending on the reader's preference).

Definition 6 *$i \cdot \exists TQBF$*

$$i \cdot \exists TQBF = \{\phi | \phi \text{ is a 3-CNF and } \exists a_1 \forall a_2 \exists a_3 \ldots Q_i a_i \text{ s.t. } \phi(a_1, a_2, \ldots, a_i) = true\}$$

It is fairly obvious to verify the following claim:

Claim 7 *$i \cdot \exists TQBF$ is Σ_i^P-complete.*

4 The Infinite Hierarchy Assumption

The Infinite Hierarchy Assumption says that there are infinitely many distinct complexity classes in the polynomial hierarchy.

Definition 8 *The IHA*

$$\begin{aligned} IHA &: \forall i : \Sigma_i^P \neq \Sigma_{i+1}^P \\ \neg IHA &: \exists i : \Sigma_i^P = \Sigma_{i+1}^P \end{aligned}$$

The IHA is a very strong assumption that would have many implications if it were true. Taking $i = 0$ would give $\Sigma_0^P \neq \Sigma_1^P$, which is exactly $P \neq NP$. However, if the IHA is false, it means that at $\Sigma_i^P = \Sigma_{i+1}^P$ for at least one i, not necessarily for $i = 0$, and thus nothing can be said with certainty about P vs. NP.

Lemma 9 *If the IHA is false,*

$$\Sigma_i^P = \Sigma_{i+1}^P \Leftrightarrow \Sigma_i^P = \Pi_i^P$$

Proof

To see that $\Sigma_i^P = \Sigma_{i+1}^P \Rightarrow \Sigma_i^P = \Pi_i^P$, note that $\Pi_i^P \subseteq \Sigma_{i+1}^P = \Sigma_{i+1}^P$. Thus, $\Pi_i^P \subseteq \Sigma_i^P$ and a similar argument would show $\Sigma_i^P \subseteq \Pi_i^P$, implying $\Sigma_i^P = \Pi_i^P$.

To see that $\Sigma_i^P = \Pi_i^P \Rightarrow \Sigma_i^P = \Sigma_{i+1}^P$, look at how you might have a debate to decide a language $L \in \Sigma_{i+1}^P$. You have $L = \{x | \exists a_1 \forall a_2 \exists a_3 \ldots Q_{i+1} a_{i+1} \;\; s.t. \;\; V(x, a_1, a_2, \ldots, a_{i+1}) = 1\}$. To have the debate, you start by sending an $\exists$ message, then an $\forall$ message, and so on. Once you send the exists message a_1, you can view the rest of the debate as deciding a language $L' = \{x, a_1 | \forall a_2 \exists a_3 \ldots Q_{i+1} a_{i+1} \;\; s.t. \;\; V(x, a_1, a_2, \ldots, a_{i+1}) = 1\}$. Therefore you can define the original language as $L = \{x | \exists a_1 \;\; s.t. \;\; (x, a_1) \in L'\}$. Since L' starts with a $\forall$ message and has i messages, $L' \in \Pi_i^P$. Since $\Pi_i^P = \Sigma_i^P$, $L' \in \Sigma_i^P$. Thus, it can be debated with a debate with i messages starting with an $\exists$ message. This means that L can be debated with $i+1$ messages starting with an $\exists$ message, and having the second message also be an $\exists$ message. But you can simply collapse these two messages into one message, meaning you only need i messages, so $L \in \Sigma_i^P$, which completes the lemma. ■

Theorem 10 *If the IHA is false, everything above a certain point in the hierarchy collapses into a single class. Specifically,*

$$\exists i : \Sigma_i^P = \Sigma_{i+1}^P \Rightarrow \forall j > i : \Sigma_i^P = \Sigma_j^P$$

Proof This is fairly easy to see since for $\Sigma_i^P = \Pi_i^P$ implies $\Sigma_i^P = \Sigma_{i+1}^P$, from the previous lemma, and you can just keep propagating this argument through the entire hierarchy above Σ_i^P. ■

5 Proof of Karp-Lipton Theorem

We are now ready to prove the Karp-Lipton Theorem stated earlier, that $IHA \Rightarrow NP \not\subseteq P/_{poly}$. To prove this, we can equivalently prove that $NP \subseteq P/_{poly} \Rightarrow \Sigma_3^P = \Pi_3^P$, violating the IHA.

Proof

The Idea of the proof is that if $NP \subseteq P/_{poly}$, we can guess, for problems of size n, a polynomial-sized circuit C that solves SAT problems. To verify that $C = SAT$, we can say that $\forall \phi : C(\phi) = SAT(\phi)$ iff $\exists y \;\; s.t. \;\; \phi(y) = 1$. While it not might seem helpful to say that we can solve SAT in a three-round debate, since we could solve it in a one-round debate, it actually is helpful in showing $\Sigma_3^P = \Pi_3^P$ under these assumptions.

We can write out the debated language deciding SAT more formally as:

$$SAT = \{x | \exists C \forall \phi, y' \exists y : C(\phi) = 0 \Rightarrow \phi(y') = 0, C(\phi) = 1 \Rightarrow \phi(y) = 1\}$$

Writing these in terms of messages and a polynomial-time verifier V which checks the conditions,

$$SAT = \{x | \exists a_1 \forall a_2 \exists a_3 \;\; V(x, a_1, a_2, a_3) = 1\}$$

The key idea is that $SAT \in \Sigma_3^P$.

Now consider a debatable language $L \in \Pi_3^P$. We wish to use $SAT \in \Sigma_3^P$ to show that $L \in \Sigma_3^P$, which completes the proof. Thus, we have

$$L = \{x | \forall a_1 \exists a_2 \forall a_3 \;\; V(x, a_1, a_2, a_3) = 1\}$$

If we could guess a_1 and a_2, we could use the related language:

$$L' = \{(x, a_1, a_2) | \forall a_3 \;\; V(x, a_1, a_2, a_3) = 1\}$$

It is clear that $L' \in coNP$. If we define $\psi(x, a_1, a_2) = (x, a_1, a_2)$. By assumption that $coNP \in P/_{poly}$, there exists a circuit C deciding the satisfiability of ψ. Thus, we can use the debate for deciding satisfiability to decide L', which gives $L' \in \Sigma_3^P$, which gives $\Sigma_3^P = \Pi_3^P$.

■