

# Computational Complexity Theory

## Lecture 18

November 24, 2006

### Interactive Proofs

Let us first recall some definitions.

**Definition.** An interactive proof system consists of the following components.

1. A DTM  $V$  running in polynomial time  $p(n)$ , called the verifier (or Arthur), which accepts or rejects.
2. An unbounded DTM  $P$ , called the prover (or Merlin).
3. A joint read-only tape containing the input  $x$ .
4. A joint communication tape, through which messages are exchanged.
5. Private work tapes for  $V$  and  $P$ .
6. A private tape for  $V$  containing a random string  $z$ .

A language  $L$  is decided by the system  $(V, P)$  if

$$\begin{aligned}x \in L &\Rightarrow \forall z \in \{0, 1\}^{p(|x|)} : (V, P)(x, z) \text{ accepts} \\x \notin L &\Rightarrow \forall P^* : \exists z \in \{0, 1\}^{p(|x|)} : \Pr[(V, P^*)(x, z) \text{ accepts}] \leq \frac{1}{2}.\end{aligned}$$

$\mathbf{IP} := \{L : L \text{ is decided by some interactive proof system}\}.$

**Remark.** By repetition,  $V$  can ensure an exponentially small error probability.

The aim of this lecture is to show that  $\mathbf{IP} = \mathbf{PSPACE}$ , which was first established by Shamir in 1989. We start with the least difficult inclusion.

**Proposition 1.**  $\mathbf{IP} \subseteq \mathbf{PSPACE}$ .

*Proof.* Suppose that  $L \in \mathbf{IP}$ , as witnessed by the pair  $(V, P)$ . Although we can easily simulate  $V$  in polynomial space, this may not be the case for  $P$ . Therefore, we will ignore  $P$ , and instead simulate the behavior of an *optimal prover*, i.e., one that maximizes the acceptance probability for each input string. A string is in  $L$  iff this probability is greater than  $\frac{1}{2}$ .

For a given input, we consider the corresponding *interaction tree*. A node in the tree is a transcript of the interaction between  $V$  and  $P$  until a certain point, and the node has an outgoing edge for each possible message that may be sent at this point. Leaves are marked according as they lead to acceptance or rejection. The tree is a huge, but finite, Markov decision process. We compute the maximum acceptance probability by a depth-first traversal.

For a node  $n$  where  $V$  is to send the next message, the maximum acceptance probability is a weighted average of the maximum acceptance probabilities of its children. The weight for an edge is the probability that the corresponding message is sent, given that node  $n$  is reached. We can compute this probability by simulating  $V$  on the path from the root to  $n$  for each setting of the random bits, ignoring settings for which  $V$  leaves the path.

At a node where  $P$  is to send the next message, we simply maximize over its children.

All we have to store is the path to the current node and a partially computed maximum probability for each node along that path. Thus, polynomial space suffices.  $\square$

**Remark.** *It follows that restricting the prover to polynomial space would result in an equivalent definition of  $\mathbf{IP}$ .*

To show  $\mathbf{PSPACE} \subseteq \mathbf{IP}$ , it suffices to give an  $\mathbf{IP}$ -protocol for QBF. Instead of working directly with formulas, we will translate them to arithmetic expressions and work with those expressions. This will allow us to exploit useful properties of polynomials, such as the fact that any two univariate polynomials either agree on everything or disagree on “almost everything”.

To avoid size problems, we will consider a restricted class of QBFs.

**Definition.** *We call a closed QBF simple if*

1. *negation is only applied to variables and*
2. *there is at most one “ $\forall$ ” between the quantification and use of a variable.*

*An open QBF is simple if its existential closure is simple.*

It is, however, equivalent to the ordinary QBFs in the following sense.

**Lemma 1.** *Given a QBF, we can construct an equivalent simple QBF in polynomial time.*

*Proof.* We first propagate negations through other connectives and quantifiers until they reach the variables, and then we transform the formula by repeatedly replacing

$$\dots Qx \dots \forall y : f(x, \dots)$$

with

$$\dots Qx \dots \forall y : \exists x' : ((x \wedge x') \vee (\neg x \wedge \neg x')) \wedge f(x', \dots).$$

$\square$

We can now safely define the translation from logic to arithmetic.

**Definition.** The arithmetization  $Ar$  of simple QBFs is defined by

$$\begin{aligned}
x &\xrightarrow{Ar} x \\
\neg x &\xrightarrow{Ar} 1 - x \\
f \vee g &\xrightarrow{Ar} Ar(f) + Ar(g) \\
f \wedge g &\xrightarrow{Ar} Ar(f) \cdot Ar(g) \\
\exists x : f(x) &\xrightarrow{Ar} \sum_{x=0}^1 Ar(f) \\
\forall x : f(x) &\xrightarrow{Ar} \prod_{x=0}^1 Ar(f).
\end{aligned}$$

The value of an arithmetic expression  $A$  is denoted by  $|A|$ .

We have the desired correspondence between formulas and expressions.

**Lemma 2.** A closed simple QBF  $f$  is true iff the value of  $Ar(f)$  is positive.

*Proof.* Structural induction. □

The simplicity constraint makes the degree of resulting polynomials polynomial.

**Lemma 3.** If  $f$  is a simple QBF of length  $n$  with exactly one free variable, then the degree of  $Ar(f)$  is at most  $2n$ .

*Proof.* There is at most one  $\Pi$  acting on the free variable, which can at most double the degree, and each other operator can increase the degree by at most one. □

To ensure a polynomial bitsize of coefficients and values, all arithmetic will be performed modulo a suitably chosen prime, whose existence is guaranteed by the following.

**Lemma 4.** If  $f$  is a closed simple true QBF of length  $n$ , then there is a prime between  $2^n$  and  $2^{3n}$  that does not divide  $|Ar(f)|$ .

*Proof.* We have  $1 \leq |Ar(f)| \leq 2^{2^n}$ . By the Prime Number Theorem, the number of primes between  $2^n$  and  $2^{3n}$  is at least  $2^n$ . Thus, their product is greater than  $|Ar(f)|$ , so by the Chinese Remainder Theorem, at least one of these primes does not divide  $|Ar(f)|$ . □

We are now finally ready to describe the actual protocol. We assume that the input is a simple QBF  $f$ .

1.  $P$  evaluates  $A := Ar(f)$ , chooses  $q$  according to Lemma 4, and sends  $q$  and the value  $a := (|A| \bmod q)$  to  $V$ . (From now on, everything is done modulo  $q$ .)
2.  $V$  verifies that  $a \neq 0$  and that  $q$  is a prime of appropriate size.
3. If  $A$  does not contain  $\Pi$  or  $\Sigma$ , then  $V$  evaluates  $A$  and accepts iff  $|A| = a$ .

4. Otherwise,  $V$  splits  $A$  into  $A_1 + A_2$  or  $A_1 \cdot A_2$ , with  $A_2$  starting with the leftmost  $\Sigma$  or  $\Pi$  in  $A$ , and assigns  $a := a - |A_1|$  or  $a := a/|A_1|$ , respectively. (In the latter case,  $V$  rejects if  $|A_1| = 0$ .)
5.  $V$ , now wanting to be convinced that  $|A_2| = a$ , removes the outermost  $\Sigma$  or  $\Pi$  from  $A_2$ , obtaining  $A_3$ , asks  $P$  for the coefficients of the corresponding polynomial in its standard form  $t$ , and verifies that  $t(0) + t(1) = a$  or  $t(0) \cdot t(1) = a$ , respectively.
6. To be convinced that the polynomial  $t$  is correct,  $V$  chooses a random  $r$ , assigns  $A := A_3(r)$ ,  $a := t(r)$ , and repeats the protocol from step 3.

If  $f$  is true, then an honest prover will convince  $V$ . Otherwise,  $P$  must in some step 5 send an incorrect  $t$ . The only way that  $P$  might get away with this, is if  $V$  in some step 6 picks an  $r$  for which the current fake and real polynomials agree. The probability that this ever happens is less than  $n \cdot \frac{2n}{2^n}$ , where  $n$  is the length of  $f$ .

## Arthur-Merlin Games

We will now consider interactive proof systems that are limited to a single round (and hence not very interactive.) The classes we obtain can be seen as randomized versions of **NP**.

**Definition.** A language  $L$  belongs to **MA** if there is a language  $L' \in \mathbf{P}$  and a polynomial  $p$  such that

$$\begin{aligned} x \in L &\Rightarrow \exists y \in \{0, 1\}^{p(|x|)} : \forall z \in \{0, 1\}^{p(|x|)} : \langle x, y, z \rangle \in L' \\ x \notin L &\Rightarrow \forall y \in \{0, 1\}^{p(|x|)} : \mathfrak{R}z \in \{0, 1\}^{p(|x|)} : \Pr[\langle x, y, z \rangle \in L'] \leq \frac{1}{2}. \end{aligned}$$

A language  $L$  belongs to **AM** if there is a language  $L' \in \mathbf{P}$  and a polynomial  $p$  such that

$$\begin{aligned} x \in L &\Rightarrow \forall z \in \{0, 1\}^{p(|x|)} : \exists y \in \{0, 1\}^{p(|x|)} : \langle x, y, z \rangle \in L' \\ x \notin L &\Rightarrow \mathfrak{R}z \in \{0, 1\}^{p(|x|)} : \Pr[\exists y \in \{0, 1\}^{p(|x|)} : \langle x, y, z \rangle \in L'] \leq \frac{1}{2}. \end{aligned}$$

**Proposition 2.**

$$\begin{array}{c} \Sigma_2^p \\ \cup \\ \mathbf{NP} \subseteq \mathbf{MA} \subseteq \mathbf{AM} \subseteq \Pi_2^p \end{array}$$

*Proof.* The only inclusion that does not follow immediately by definition is  $\mathbf{MA} \subseteq \mathbf{AM}$ . To obtain this inclusion, we first ensure an exponentially small error probability for  $L \in \mathbf{MA}$ , by repeating the work of Arthur, i.e.,  $L'$ , and then note that this allows  $L'$  to also witness that  $L \in \mathbf{AM}$ .  $\square$

**Open problem.**  $\mathbf{AM} \stackrel{?}{\subseteq} \Sigma_2^p$ .

**Conjecture.**  $\mathbf{NP} = \mathbf{MA} = \mathbf{AM}$ .