In this lecture we discuss Shamir's theorem that PSPACE is the set of languages that have interactive proofs with an arbitrary (polynomial) number of rounds.

Theorem 1 (Shamir). IP = PSPACE.

There are two directions here. For IP $\subseteq$ PSPACE, we show that the accepting probability of every interactive protocol with a polynomial-time verifier can be computed by a polynomial space machine. Then this machine can solve every language in IP by checking if the accepting probability exceeds 2/3 or is at most 1/3.

Let us consider an arbitrary interactive protocol specified by a polynomial-time verifier V. Recall that with a loss of at most two rounds of interaction, we can assume that the protocol is public coin. On input x of length n, the verifier sends a random string $r_1 \in \{0,1\}^{p(n)}$, the prover answers with $a_2 \in \{0,1\}^{p(n)}$, the verifier responds with a random $r_3 \in \{0,1\}^{p(n)}$, and so on. After $k(n)$ messages have been exchanged, the verifier makes a decision as to whether $x \in L$. The *accepting probability* of V is the quantity

$$\max_{P^*} \Pr[(P^*, V)(x) \text{ accepts}] = \mathrm{E}_{r_1}[\max_{a_2} \mathrm{E}_{r_3}[\max_{a_4} \ldots \Pr[V(x; r_1, a_2, r_3, a_4, \ldots, r_{k(n)}) \text{ accepts}] \ldots]]$$

where what we mean by "$V(x; y_1, \ldots, y_k)$ accepts" is that V accepts input x when the transcript of the interaction between V and the prover consists of the messages $y_1, \ldots, y_k$. The accepting probability is a quantity that depends entirely on the verifier. If there exists a prover that can make the verifier accept often, this probability is high. If no such prover exists, then it is low.

For every x, the accepting probability can be computed as follows. For every fixed r_1 and a_2 in $\{0,1\}^{p(n)}$, let

$$p(r_1, a_2) = \mathrm{E}_{r_3}[\max_{a_4} \ldots \Pr[V(x; r_1, a_2, r_3, a_4, \ldots, r_{k(n)}) \text{ accepts}] \ldots].$$

The accepting probability of the verifier is then

$$\mathrm{E}_{r_1}[\max_{a_2} p(r_1, a_2)]$$

and this expression can be computed recursively using space polynomial in $|x|$.

We now turn to the more interesting direction: PSPACE $\subseteq$ IP.

1 Boolean encodings for PSPACE

Now we are given a PSPACE language L and we want to design an interactive protocol for it. Let us look at a Turing Machine that decides L in space $s(n)$ on inputs of length n. We know that

such a machine must always halt in time $2^{O(s(n))}$. However it might be the case that the machine takes time much less than $2^{O(s(n))}$ on certain inputs. It will be convenient for us that the machine takes the same time to halt on *all* inputs of length n, and that this time is always a power of two. In fact this can be arranged by carefully making sure that the machine keeps track of its running time and idles for a certain number of steps after it is done with its computation. The details are not so interesting so we summarize this and a few other convenient facts in the following claim.

Claim 2. *For every L in* PSPACE *there exists a polynomial-space Turing Machine M that decides L and a polynomial $t(n)$ such that for every x,*

- *$M(x)$ halts in* exactly $2^{t(|x|)}$ *steps.*
- *$M(x)$ has a unique accepting configuration.*
- *The computation graph of $M(x)$ is acyclic.*

Let M be a machine of this type for L, so that on inputs of length n, M uses space $s(n)$ and runs in time exactly $2^{t(n)}$. Recall that a configuration of M is specified by the state of the control, the position of the head, and the contents of the first $s(n)$ cells of the tape. We can think of this as a string z of symbols $z_1, \ldots, z_{s(n)}$, each taking value in some fixed size alphabet Σ. The value of z_i includes the contents at the ith position of the tape together with information that indicates whether the head is currently at the ith position and, if so, what state the machine is in.

Just as in the proof of the Savitch theorem, given a pair of configurations u and v and a time bound t, we define a formula $\varphi_t(u, v)$ that says whether M goes from configuration u to configuration v in exactly 2^t steps. The formulas φ_t are defined recursively via the equation

$$\varphi_{t+1}(u, v) = \exists w \in \{0,1\}^{s(n)} : \varphi_t(u, w) \wedge \varphi_t(w, v).$$

In the base case, we have

$$\varphi_1(u, v) = \bigwedge_{i=1}^{s(n)} \text{valid}(u_{i-1}, u_i, u_{i+1}, v_i)$$

where $\text{valid}(y_1, y_2, y_3, y)$ says that the transition from (y_1, y_2, y_3) to y is allowed to occur in a computation tableau of M.

Usually, we think of the quantified variables u_i and v_i as indeterminates taking values in Σ, and φ_t as a formula that maps Σ to "true" or "false". The key idea in the proof is to look at an alternative representation of φ_t which we now describe.

2 Low degree extension and arithmetization

For the moment let us forget about φ_t and look at an arbitrary boolean valued formula φ over variables $y_1, \ldots, y_k$ taking values in Σ. We can think of φ as a function mapping Σ^k to $\{0, 1\}$, where 0 stands for "false", and 1 stands for "true".

Let's now think of Σ as a subset of some finite field $\mathbb{F}$, and think of φ as a function mapping Σ^k to $\mathbb{F}$ which happens to always take values 0 and 1. The following claim shows that any such function can be "extended" to a low-degree polynomial over $\mathbb{F}$.

Claim 3. *For every function $\varphi : \Sigma^k \to \mathbb{F}$ there exists a k-variate polynomial q over $\mathbb{F}$ where the degree of each variable is at most $|\Sigma| - 1$ and such that*

$$q(a_1, \ldots, a_k) = \varphi(a_1, \ldots, a_k)$$

for all $(a_1, \ldots, a_k) \in \Sigma^k$.

Proof. By induction on k. Suppose for every $i \in \Sigma$ we have a polynomial $q_i(x_2, \ldots, x_k)$ such that the degree of every variable is at most $|\Sigma| - 1$ and

$$q_i(a_2, \ldots, a_k) = \varphi(i, a_2, \ldots, a_k)$$

for all $(a_2, \ldots, a_k) \in \Sigma^{k-1}$. We now define q by interpolation:

$$q(x_1, \ldots, x_k) = \sum_{i \in \Sigma} \frac{\prod_{j \in \Sigma - \{i\}} (x_1 - j)}{\prod_{j \in \Sigma - \{i\}} (i - j)} q_i(x_2, \ldots, x_k).$$

It is easy to check that q has degree $|\Sigma| - 1$ in all its variables and that $q(a_1, \ldots, a_k) = \varphi(a_1, \ldots, a_k)$ for all $(a_1, \ldots, a_k) \in \Sigma^k$. □

The polynomial q can be thought as a *polynomial extension* of φ. When the inputs fall inside Σ, q says exactly whether φ is true or false. Outside Σ, q outputs some element in $\mathbb{F}$ which is superficially meaningless, but will be crucial in the construction of the interactive protocol for L.

Applying Claim 3 to the predicate valid we obtain its polynomial extension q_{valid} where each variable has degree less than $|\Sigma|$. We now define the *arithmetization* q_t of φ_t by replacing valid with q_{valid}, "exists" with "plus", and "and" with "times".

$$q_{t+1}(u, v) = \sum_{w \in \{0,1\}^{s(n)}} q_t(u, w) \times q_t(w, v)$$

and

$$q_1(u, v) = \prod_{i=1}^{s(n)} q_{\text{valid}}(u_{i-1}, u_i, u_{i+1}, v_i)$$

We argue that $q_t(u, v)$ and $\varphi_t(u, v)$ take the same value when all u_i and v_i are in Σ. For q_1 and φ_1 this is easy to check — products and "and"s take the same value over a $\{0, 1\}$ domain. For q_{t+1} and φ_{t+1} this is not so obvious. In general, when we replace an existential quantifier with a sum, the formula and its corresponding arithmetized polynomial may no longer take the same value. For instance, if the existential quantifier has two satisfying assignments then the value of the corresponding sum will be two.

However, in our setting this will never happen; recall that $\varphi_{t+1}(u, v)$ says whether v is reachable from u in exactly 2^{t+1} steps. If this is the case, there will be a *unique* w such that w is reachable from

u in exactly 2^t steps and v is reachable from w in exactly 2^t steps. (Recall that M is deterministic and never loops.) So for every pair (u, v), there will be either zero or one witnesses w in φ_{t+1}. It follows that the value of the corresponding sum in q_{t+1} either 0 (when there is no w) or 1 (when there is exactly one w).

We need one more technicality: For every t, the *individual* degree of every variable in q_t is at most $4|\Sigma|$. For q_1, this is true because every variable appears in at most four instantiations q_{valid}, and for larger t it can be seen by induction on t that the individual degree of every variable never increases.

Let's now step back and recall what we are trying to do. In the end, given an input x of length n, we want to know if $M(x)$ accepts. This is the same as asking whether

$$q_{t(n)}(start_x, accept) = 1$$

where $start_x$ is the initial configuration of $M(x)$, which we now think of as a string in $\Sigma^{s(n)}$, and *accept* is the (unique) accepting configuration of M.

Now $q_{t(n)}$ is a huge polynomial which we don't even know how to write down explicitly. However we have an *implicit* representation of $q_{t(n)}$ in terms of sums and products of simple polynomials q_1. We next turn to designing interactive protocols for evaluating such implicitly represented polynomials.

3 Sum and product protocols

The approach will be to start with $q_{t(n)}$ and repeatedly "strip" the sums and products until we are left with evaluating q_1, which we can do easily.

To begin with, let's suppose we have an implicit[1] polynomial $q(y)$ in m variables and we want to evaluate $q(a) + q(a')$, where a and a' are points in $\mathbb{F}^m$. We show a simple interactive protocol that reduces the problem to evaluating $q(b)$ for some different b in $\mathbb{F}^m$. Let's see more precisely what this means.

Lemma 4 (Sum protocol). *There is a 1-round interactive protocol (P, V) with the following behavior. The input to the protocol is an implicit polynomial q over $\mathbb{F}$ in m variables of total degree d, two points $a, a' \in \mathbb{F}^m$, and a value $\alpha \in \mathbb{F}$. The output of the protocol is a point $b \in \mathbb{F}^m$ and a value $\beta \in \mathbb{F}$.*

- *If $q(a) + q(a') = \alpha$, then for the prover P, $\Pr[q(b) = \beta] = 1$.*
- *If $q(a) + q(a') \neq \alpha$, then for every prover P^*, $\Pr[q(b) = \beta] \leq d/|\mathbb{F}|$.*

We can now imagine the following interaction. The prover wants to prove that $q(a) + q(a') = \alpha$. If this is true, the verifier has reduced the task to proving the simpler claim $q(b) = \beta$. If this is false, then the prover must trick the verifier into accepting the simpler false claim that $q(b) = \beta$.

[1] Let's not worry too much about what "implicit" means exactly.

Proof. Let $l(t) = ta + (1-t)a'$ be the line going through a and a'. The protocol relies on the simple observation that $r(t) = q(l(t))$ is a univariate polynomial of degree at most d. We now give the protocol.

1. Prover: Send the description of a univariate polynomial $\hat{r}(t) = c_0 + c_1 t + \cdots + c_d t^d$ of degree d by giving its coefficients $c_0, \ldots, c_d$. The honest prover P sends the coefficients of $r(t)$.
2. Verifier: Check that $\hat{r}(0) + \hat{r}(1) = \alpha$. Choose a random $t_0 \in \mathbb{F}$ and output the pair $b = l(t_0)$, $\beta = \hat{r}(t_0)$.

If $q(a) + q(a') = \alpha$, then the prover sends $\hat{r}(t) = r(t)$, and no matter which t_0 the verifier chooses, $\beta = r(t_0) = q(b)$.

Now suppose that $q(a) + q(a') \neq \alpha$ and the prover sends $\hat{r}(t)$. If $\hat{r}(0) + \hat{r}(1) \neq \alpha$, the verifier rejects. Otherwise, $\hat{r}(0) + \hat{r}(1) = \alpha \neq r(0) + r(1)$, so $r(t)$ and $\hat{r}(t)$ must be different polynomials of degree at most d. Then $r(t) - \hat{r}(t)$ is a nonzero polynomial of degree at most d, and it has at most d zeros. It follows that for a random t_0,

$$\Pr_{t_0}[q(b) = \beta] = \Pr_{t_0}[r(t_0) = \hat{r}(t_0)] = \Pr_{t_0}[r(t_0) - \hat{r}(t_0) = 0] \leq d/|\mathbb{F}|. \qquad \square$$

Replacing addition with multiplication, we obtain a completely analogous *product protocol*.

4 Interactive proofs for q_t

Let us now fix the field $\mathbb{F}$ to be of size $\Theta(s(n)^2 t(n))$.

We now apply the sum and product protocols to the polynomials q_t. Suppose the prover claims that $q_{t+1}(u, v) = \alpha$. We can write

$$q_{t+1}(u, v) = h_1(u, v, 0) + h_1(u, v, 1)$$

where

$$h_1(u, v, w_1) = \sum_{w_2, \ldots, w_{s(n)}} q_t(u, w_1, w_2, \ldots, w_{s(n)}) \times q_t(w_1, w_2, \ldots, w_{s(n)}, v).$$

Since in q_t, each variable has individual degree at most $4|\Sigma| = O(1)$, h has total degree at most $O(s(n))$. Applying the sum protocol to h on inputs $(u, v, 0)$, $(u, v, 1)$ and α, we obtain a new pair (u', v', w_1') and β such that

- If $q_{t+1}(u, v) = \alpha$, then $h(u', v', w_1') = \alpha'$
- If $q_{t+1}(u, v) \neq \alpha$, then $h(u', v', w_1') \neq \alpha'$ with probability at least $1 - O(s(n)/|\mathbb{F}|)$.

Continuing in the same manner, we reduce the claim $h_1(u', v', w_1') = \alpha'$ to $h_2(u'', v'', w_1'', w_2'') = \alpha''$, where

$$h_2(u, v, w_1, w_2) = \sum_{w_3, \ldots, w_{s(n)}} q_t(u, w_1, w_2, \ldots, w_{s(n)}) \times q_t(w_1, w_2, \ldots, w_{s(n)}, v)$$

and so on until we are down to the claim $h_{s(n)}(u''', v''', w_1''', \ldots, w_{s(n)}''') = \alpha'''$, where

$$h_{s(n)}(u, v, w_1, \ldots, w_n) = q_t(u, w_1, w_2, \ldots, w_{s(n)}) \times q_t(w_1, w_2, \ldots, w_{s(n)}, v).$$

Adding the errors from the intermediate sum protocols, we have that

- If $q_{t+1}(u, v) = \alpha$, then $h_{s(n)}(u''', v''', w_1''', \ldots, w_{s(n)}''') = \alpha'''$,
- If $q_{t+1}(u, v) = \alpha$, then $h_{s(n)}(u''', v''', w_1''', \ldots, w_{s(n)}''') \neq \alpha'''$ with probability at least $1 - O(s(n)^2/|\mathbb{F}|)$.

We can now apply the product protocol to q_t on inputs (u''', w'''), (w''', v''') and α''' to obtain outputs u'''', v'''' and α'''' such that

- If $q_{t+1}(u, v) = \alpha$, then $q_t(u'''', v'''') = \alpha''''$
- If $q_{t+1}(u, v) \neq \alpha$, then $q_t(u'''', v'''') \neq \alpha''''$ with probability at least $1 - O(s(n)^2/|\mathbb{F}|)$.

Now, using induction on t, we obtain a protocol that, on input $start_x, accept, 1$ outputs u^*, v^* and α^* such that

- If $q_{t(n)}(start_x, accept) = 1$, then $q_1(u^*, v^*) = \alpha^*$
- If $q_{t(n)}(start_x, accept) = 0$, then $q_1(u^*, v^*) \neq \alpha^*$ with probability $1 - O(s(n)^2 t(n)/|\mathbb{F}|) \geq 2/3$.

The verifier can now check the claim $q_1(u^*, v^*) = \alpha^*$ in polynomial-time. Putting everything together, we have that for some prover P (whose job is to send the description of a polynomial in every step)

- If $M(x)$ accepts, then $(P, V)(x)$ accepts with probability 1,
- If $M(x)$ rejects, then for every prover P^*, $(P^*, V)(x)$ rejects with probability at least $2/3$,

so $L \in \mathrm{IP}$.