Lecture 18

Super-Exponential Complexity of Presburger Arithmetic

March 10, 1998
Lecturer: Richard Ladner
Notes:

Recall that Presburger Arithmetic is the theory of the natural numbers with addition. More specifically, let $\mathcal{P} = (N, s, +, <, 0)$. Presburger arithmetic is $\mathrm{Th}(\mathcal{P})$.

We have the following theorem proved by Fischer and Rabin in 1974.

Theorem 18.1: $\mathrm{Th}(\mathcal{P}) \notin \mathrm{NTIME}(2^{2^{cn}})$ for some $c > 0$.

Proof: The idea is to build a formula of linear size that describes the computation of a nondeterministic Turing machine that runs in double exponential time. There are a number of building blocks needed in order to build our formula.

We begin by defining a formula $M_n(x, y, z)$ if length $O(n)$ with the property for all nonnegative integers a, b, and c.

$$\mathcal{P} \models M_n(a, b, c) \text{ if and only if } a < 2^{2^n} \text{ and } ab = c.$$

As can be seen the formula M_n defines a limited form of multiplication. We build M_n inductively on n.

$$M_0(x, y, z) = (x = 0 \wedge z = 0) \vee (x = 1 \wedge z = y).$$

In order to understand the formula M_n for $n > 0$ we use the following fact about nonnegative integers:

$$x < 2^{2^{n+1}} \Leftrightarrow \exists x_1, x_2, x_3, x_4 < 2^{2^n}(x = x_1 x_2 + x_3 + x_4).$$

Hence, $xy = x_1(x_2 y) + x_3 y + x_4 y$. If we let $u_1 = x_2 y$, $u_2 = x_1 u_1$, $u_3 = x_3 y$, $u_4 = x_4 y$ then we can define M_{n+1} as:

$$\begin{aligned} M_{n+1}(x, y, z) \quad = \quad & (\exists u_1 u_2 u_3 u_4 x_1 x_2 x_3 x_4)(\forall w_1 w_2 w_3)[(((w_1 = x_2 \wedge w_2 = y \wedge w_3 = u_1) \\ & \vee (w_1 = x_1 \wedge w_2 = u_1 \wedge w_3 = u_2) \vee (w_1 = x_3 \wedge w_2 = y \wedge w_3 = u_3) \\ & \vee (w_1 = x_4 \wedge w_2 = y \wedge w_3 = u_4)) \Rightarrow M_n(w_1, w_2, w_3)) \wedge z = u_2 + u_3 + u_4] \end{aligned}$$

The key thing to observe is that the length of M_n is linear in n.

Now, M_n is a good formula defining multiplication by numbers less than 2^{2^n}. Unfortunately, this would only let us describe computations of exponential length, not double exponential length. So we have to go a bit further.

We will now describe a formula M'_n with the property

$$\mathcal{P} \models M'_n(a,b,c) \text{ if and only if } a,b,c < g(n) \text{ and } ab = c$$

where g is a function such that $g(n) \geq 2^{2^{2^{n+1}}}$.

Here we need a little number theory. Define $m(n) = 2^{2^{n+2}}$ and

$$g(n) = \prod_{\substack{p<m(n)\\ p \text{ prime}}} p.$$

We define three auxiliary formulas P_n, G_n, and C_n, each of length $O(n)$. We begin with P_n that has the following meaning.

$$\mathcal{P} \models P_n(a) \text{ if and only if } a < m(n) \text{ and } a \text{ is prime.}$$

Recall is that $\mathcal{P} \models M_{n+2}(a,0,0)$ if and only if $a < 2^{2^{n+2}} = m(n)$. Thus, we can define P_n as follows:

$$P_n(x) = M_{n+2}(x,0,0) \wedge \forall x \forall y (M_{n+2}(y,z,x) \Rightarrow (y = 1 \vee y = x)).$$

Next, G_n has the following meaning.

$$\mathcal{P} \models G_n(a) \text{ if and only if } a = g(n).$$

Using the fact that $g(n)$ is the smallest number divisible by all primes less than $m(n)$, we define G_n as follows:

$$G_n(x) = \forall p(P_n(p) \Rightarrow \exists z M_{n+2}(p,z,x)) \wedge \forall y(\forall p(P_n(p) \Rightarrow \exists z M_{n+2}(p,z,y)) \Rightarrow x \leq y)$$

Next, C_n has the following meaning.

$$\mathcal{P} \models C_n(a,b,c,p) \text{ if and only if } p < m(n) \text{ and } ab \equiv c \bmod p.$$

Notice that $xy \equiv z \bmod p$ if and only if there are q_1, q_2, q_3, $r_1 < p$, and $r_2 < p$ such that

$$x = pq_1 + r_1, \quad r_1 y = pq_2 + r_2, \quad z = pq_3 + r_2$$

Each of the multiplications above has its first multiplicand $< m(n)$ so that we can use our previously defined multiplication predicate. Thus we can define C_n as follows using u_1, u_2, u_3, and u_4 to hold the various products.

$$\begin{aligned}
C_n(x,y,z,p) \quad = \quad & M_{n+2}(p,0,0) \wedge \exists u_1 u_2 u_3 u_4 q_1 q_2 q_3 r_1 r_2 [r_1 < p \wedge r_2 < p \\
& \wedge M_{n+2}(p,q_1,u_1) \wedge x = u_1 + r_1 \\
& \wedge M_{n+2}(r_1,y,u_2) \wedge M_{n+2}(p,q_2,u_3) \wedge u_2 = u_3 + r_2 \\
& \wedge M_{n+2}(p,q_3,u_4) \wedge z = u_4 + r_2]
\end{aligned}$$

Given G_n and C_n we can now define M'_n as follows.

$$\begin{aligned} M'_n(x,y,z) &= \exists w(G_n(w) \wedge x < w \wedge y < w \wedge z < w \\ & \quad \forall p(p < w \Rightarrow C_n(x,y,z,p)). \end{aligned}$$

So we can define multiplication for numbers less than $2^{2^{2^n}}$. In order to simulate a Turing machine like we did in Gödel's incompleteness theorm we need bounded exponentiation also. That is, we need a formula Pow_n of length $O(n)$ with the property that

$$\mathcal{P} \models Pow_n(a,b,c) \text{ if and only if } a, b^a, , c < g(n) \text{ and } b^a = c.$$

To do this in the required length we need to define bounded powering (in conjuction with multiplication) in a single formula E_k with the property If $a < 2^{2^k}, b^a, c < 2^{2^n}$

$$\mathcal{P} \models E_k(a,b,c,A,B,C) \text{ if and only if } A < 2^{2^n} AB = C, \text{ and } b^a = c.$$

We can begin with

$$E_0(x,y,z,u,v,w) = [(x = 0 \wedge z = 1) \vee (x = 1 \wedge z = y)] \wedge M_n(u,v,w).$$

To define $E_{k+1}(x,y,z,u,v,w)$ note that $x < 2^{2^{k+1}}$ if and only if $x = x_1 x_2 + x_3 + x_4$ for some $x_1, x_2, x_3, x_4 < 2^{2^k}$. Thus $y^x = (y^{x_1})^{x_2} \cdot y^{x_3} \cdot y^{x_4}$. Thus, E_{k+1} can be defined recursively from E_k.

Now that we have powering defined for numbers less than 2^{2^n} we can using the modular arithmetic technique define powering for numbers less than $g(n)$.

Using Gödel's proof we can, given any nondeterministic Turing machine T that runs in time $2^{2^{cn}}$ and input x of length n, construct a formula F of length $O(n)$ with the property that $\mathcal{P} \models F$ if and only if x is accepted by T. This is sufficient to prove the theorem. $\square$