Math 17
Winter 2015
Wednesday, February 11

The remaining step in proving that all Turing semidecidable sets are Diophantine is proving the functions $AfterP$ and $AfterT$ are Diophantine.

To do this, we defined a k, ℓ *superconfiguration* to be, loosely speaking, a concatenation of $(k+1)$-many configurations of length at most ℓ, in each of which the tape head is reading some cell to the left of cell ℓ.

More precisely, if (p_0, t_0), (p_1, t_1), ..., (p_k, t_k) are configuration codes, where p_i codes a sequence x_i of length ℓ whose last entry is 0, and t_i codes a sequence y_i (whose first element is, necessarily 1, since $*$ is always the first symbol on a Turing machine tape), we let p^* code $x_0^\frown x_1^\frown \cdots^\frown x_k$ and t^* code $y_0^\frown y_1^\frown \cdots^\frown y_k$. The pair (p^*, t^*) is a k, ℓ superconfiguration code. The i^{th} block of the superconfiguration coded by (p^*, t^*) is the configuration coded by (p_i, t_i).

Now $AfterP(k, p, t) = p'$ and $AfterT(k, p, t) = t'$ if and only if there are some large number ℓ and some k, ℓ superconfiguration code (p^*, t^*) with blocks (p_0, t_0), (p_1, t_1), ..., (p_k, t_k) such that $(p_0, t_0) = (p, t)$, $(p_k, t_k) = (p', t')$, and for every $i < k$,

$$(p_{i+1}, t_{i+1}) = (NextP(p_i, t_i), NextT(p_i, t_i)).$$

The number ℓ must be larger than the length of the sequences coded by p and t, and also large enough so that the tape head will remain to the left of cell ℓ when Turing machine M acts on the configuration coded by (p, t) for k-many steps. It will do to take $\ell = t + p + k + 1$.

We are defining functions $NextT^*(k, \ell, p^*, t^*)$ and $NextP^*(k, \ell, p^*, t^*)$ such that if (p^*, t^*) is a k, ℓ superconfiguration code with blocks (p_0, t_0), (p_1, t_1), ..., (p_k, t_k), then

$$(NextP^*(k, \ell, p^*, t^*), NextT^*(k, \ell, p^*, t^*))$$

is a k, ℓ superconfiguration code with blocks $(NextP(p_i, t_i), NextT(p_i, t_i))$.

To define $NextT^*$, in a Diophantine way, we use the following lemma (proof to come later):

> Lemma: For every $u < b$ and $v < b$, there is a Diophantine function $h_{u,v}(N, x, y)$ such that if x codes the sequence $(x_1, \ldots, x_N)$ and y codes the sequence $(y_1, \ldots, y_N)$, then $h_{u,v}(N, x, y)$ codes the sequence $(z_1, \ldots, z_N)$, where
>
> $$z_i = \begin{cases} 1 & \text{if } x_i = u \ \& \ y_i = v; \\ 0 & \text{if not.} \end{cases}$$

We note that if x and y code sequences of length N, then

$$\sum_{u<b,\ v<b} a_{uv} h_{u,v}(N, x, y)$$

codes a sequence $(w_1, \ldots, w_N)$ such that if $x_j = u$ and $y_j = v$ then $w_j = a_{uv}$.

To define $NextT^*(k, \ell, p^*, t^*)$, we choose coefficients a_{uv} such that if p^* and t^* have u and v (respectively) in position j, then $NextT^*(k, \ell, p^*, t^*)$ should have a_{uv} in position j. Then we set

$$NextT^*(k, \ell, p^*, t^*) = \sum_{u<b,\ v<b} a_{uv} h_{u,v}((k+1)\ell, p^*, t^*).$$

Suppose there is an instruction $(q_u\ \alpha_v\ \Rightarrow\ \alpha_r\ D\ q)$ in M (where D is any direction and q any state). Now suppose that p^* has a u in position j and t^* has a v in position j. That indicates a tape head on cell j reading symbol α_v while the machine is state q_u. The machine should then write α_r, so $NextT^*(k, \ell, p^*, t^*)$ should have r in position j. For these pairs we set

$$a_{uv} = r.$$

Suppose there is no such instruction. That could happen because $u = 0$ (indicating that the tape head is in another location), or because q_u is a final state, or because u is not actually the index of a state. (If (p^*, t^*) is actually a superconfiguration, this won't happen, but we must define our coefficient anyway.) In these cases, the machine will not write on this cell, so the symbol will be unchanged. For these pairs we set

$$a_{uv} = v.$$

This finishes the Diophantine definition of $NextT^*$.

Note: If (p^*, t^*) is not actually a superconfiguration code, but its i^{th} block is a configuration code, $NextT^*$ will nevertheless do the appropriate thing to the i^{th} block.

The definition of $NextP^*$ is done in almost exactly the same way. The difference is that the value in the j^{th} position of $NextP^*(k, \ell, p^*, t^*)$ depends not only on the j^{th} positions of p^* and t^*, but on the positions to the left and right.

We can define Diophantine functions L and R such that iff x codes the sequence $(x_1, \ldots, x_N)$ then $L(x, N)$ codes $(0, x_1, \ldots, x_{N-1})$ (the sequence of elements to the left of the elements of x) and $R(x, N)$ codes $(x_2, \ldots, x_N, 0)$ (the sequence of elements to the right of the elements of x):

$$L(x, N) = rem(bx, b^N) \qquad R(x, N) = div(x, b).$$

Then we can define functions $h_{u,v,u',v',u'',v''}$ in the same way as the $h_{u,v}$, and set

$$NextP^*(k, \ell, p^*, t^*) = \sum_{u<b,\ v<b, u'<b,\ v'<b, u''<b,\ v''<b}$$

$$c_{uvu'v'u''v''} h_{u,v,u',v',u'',v''}((k+1)\ell, p^*, t^*, L(p^*, (k+1)\ell), L(t^*, (k+1)\ell), R(p^*, (k+1)\ell), R(t^*, (k+1)\ell)),$$

where the coefficients $c_{uvu'v'u''v''}$ are chosen to give the correct values.

Now we have Diophantine functions $NextP^*$ and $NextT^*$ that, applied to a k, ℓ superconfiguration code, correctly apply $NextP$ and $NextT$ to each block. If we apply them to p^* and t^* that are not superconfiguration codes, but have an i^{th} block that is a configuration code, they still correctly apply $NextP$ and $NextT$ to the i^{th} block, provided two things are true:

1. The configuration coded by the i^{th} block has the tape head to the left of cell ℓ. (Otherwise, the tape head will, loosely speaking, run into the next block.)

2. The symbol in position $\ell(i+1)+1$ of t^* is a 1. (That is, interpreting the $(i+1)^{th}$ block as a configuration code, the first symbol on the tape is $*$. Otherwise, the tape head from the next block could move backward into this block.)

We can say in a Diophantine way that t^* has a 1 in every position $\ell(j)+1$:
The sequence of length $(k+1)\ell$ with a 1 in every position $\ell(j)+1$ is coded by the number

$$\sum_{j=0}^{k} b^j \ell = Repeat(1, b^\ell, k+1).$$

The entries of t^* in every position $\ell(j)+1$ are 1 or greater if

$$PNotGreater(Repeat(1, b^\ell, k+1), t^*, b).$$

The entries of t^* in every position $\ell(j)+1$ are 1 or less if

$$PNotGreater(t^*, Repeat(b-1, b, (k+1)\ell) - (b-2)Repeat(1, b^\ell, k+1), b).$$

We say (p^*, t^*) is a k, ℓ pseudo-superconfiguration (PSC), if p^* and t^* code sequences of length (at most) $(k+1)\ell$ (which we can say in a Diophantine way as $Code(p^*, b, (k+1)\ell)$ and $Code(t^*, b, (k+1)\ell)$), and when we divide them into $(k+1)$-many blocks of length ℓ, each block of t^* begins with 1 (which we can also say in a Diophantine way).

We can now say

$$p' = AfterP(k, p, t) \quad \& \quad t' = AfterT(k, p, t)$$

in a Diophantine way:

There is ℓ such that $\ell = p + t + k + 1$, and there are p^* and t^* forming a k, ℓ PSC, such that:

1. The first block is (p, t):

$$rem(p^*, b^\ell) = p \;\&\; rem(t^*, b^\ell) = t.$$

2. The last block is (p', t'):

$$div(p^*, b^{k\ell}) = p \;\&\; div(t^*, b^{k\ell}) = t.$$

3. Each block is the Next of the previous block:

   p^* and t^* with the first block removed equal $NextP^*(k, \ell, p^*, t^*)$ and $NextT^*(k, \ell, p^*, t^*)$ with the last block removed:

$$div(p^*, b^\ell) = rem(NextP^*(k, \ell, p^*, t^*), b^{k\ell}) \quad \& \quad div(t^*, b^\ell) = rem(NextT^*(k, \ell, p^*, t^*), b^{k\ell}).$$

The one remaining task is to prove the Lemma.

Let w_{uv} be variables for all $u, v < b$. Let $H_{uv}(N, x, y)$ be the statement $w_{uv} = h_{u,v}(N, x, y)$, and $H(N, x, y)$ be the conjunction (connection with "and") of all the $H_{uv}(N, x, y)$. If we can show $H(N, x, y)$ can be written in a Diophantine way, then we can say

$$h_{u,v}(N, x, y) = e$$

in a Diophantine way, as

$$(\exists w_{00})(\exists w_{01}) \cdots (\exists w_{(b-1)(b-1)})[H(N, x, y) \ \& \ e = w_{uv}].$$

We can write $H(N, x, y)$ as the conjunction of the following statements:

For every (u, v),

$$PNotGreater(w_{uv}, Repeat(1, N, b), b).$$

This says that every w_{uv} codes a sequence of 0's and 1's of length N.

For every (u, v) and $(u', v') \neq (u, v)$,

$$PNotGreater(w_{uv} + w_{u'v'}, Repeat(1, N, b), b).$$

This says no two sequences coded by the w_{uv} have 1's in the same position.

$$\sum_{u<b \ v<b} w_{uv} = Repeat(1, N, b).$$

This says, for every position j between 1 and N, exactly one of the w_{uv} sequences has a 1 in position j.

$$\sum_{u<b, \ v<b} uw_{uv} = x \quad \& \quad \sum_{u<b, \ v<b} vw_{uv} = y.$$

This says that, in the positions where the w_{uv} sequence has a 1, x has a u and y has a v.

This completes the proof.