Turing-Equivalence of $Th(\mathcal{N}, +, \times)$

An interjection not found in the book

Peano Arithmetic

- Axioms of natural arithmetic made by Giuseppe Peano
- The natural numbers with addition and multiplication
  - AKA $Th(\mathcal{N}, +, \times)$
- Later used by Gödel, Church, and Turing to kill logic

Implicit Functions and Iteration in Peano Arithmetic

- We are limited to two relations: $+$ and $\times$
- However, we can use these to construct more!
  - **Even recursive ones**
  - They just have to be *implicit*

Ex:

$$\forall a, b[(Square(a) = b) \leftrightarrow (b = a \cdot a)]$$

Preliminary Functions

- We can add and multiply, but we need a lot more!
- We can implicitly define subtraction, exponentiation, division, modulo
- As well as some special encoding functions Get_2 and Set_2
- These can get and set bits from an arbitrary-length binary integer

Let $\phi_{preliminary}$ be the equation describing these functions.

Preliminary Functions (Formally)

$$\begin{aligned}
\phi_{preliminary} = \forall a, b, c([(a - b = c) \leftrightarrow (a = c + b)] \\
&\wedge \left[\left(\frac{a}{b} = c\right) \leftrightarrow (a = bc)\right] \\
&\wedge [(Rem(a, b) = c) \leftrightarrow \exists d[db + c = a]] \\
&\wedge \left[(a^0 = 1) \wedge (b > 0 \to (a^b = a \cdot a^{b-1}))\right] \\
&\wedge [Log(b^a, b) = a] \\
&\wedge \left[Get_2(a, b) = \frac{Rem(a, 2^{b+1}) - Rem(a, 2^{b-1})}{2^b}\right] \\
&\wedge \left[Set_2(a, b, c) = a + (c - Get_2(a, b)) \cdot 2^b\right])
\end{aligned}$$

- We could bring this to prenex normal form, but it doesn't really matter

Gödel Numberings of TMs

- We previously gave a configuration as $w_l q_i w_r$ for some state q_i and tape content $w_l w_r$
- This will be too clunky for Gödel numbering!
- Let $n = |w_l|$, let w be a binary integer encoding $w_l w_r$

Def: *A* Gödel TM encoding. For configuration $C = w_l q_i w_r$ where $n = |w_l|$, let w be a positive binary integer encoding of $w_l w_r$. Then, a Gödel encoding of C is given by $2^i 3^n 5^w$.

Encoding the Transition Function

- We want to create some implicit function T which takes us from one configuration to the next

T:

- For all state indices i, j, R/W head position n, string content encoding w, binary tape character z, and $\Delta \in \{-1, 1\}$:
- If $\delta(q_i, Get_2(w, n)) = (q_j, z, \Delta)$, **then**
- $T(2^i 3^n 5^w) = 2^j 3^{n+\Delta} 5^{Set_2(w,n,z)}$
- T can be encoded in Peano arithmetic as a large piecewise function

Let ϕ_T be the formula describing T.

Encoding TM Simulation

- Here's where loops come in
- For every configuration C_t:

$$\ldots \wedge \forall t[C_{t+1} = T(C_t)]$$

- We can then specify some initial configuration on the input string encoding in binary as w: $C_0 = 2^0 3^0 5^w$

The Acceptance Problem in Peano Arithmetic

- If we let the index of the accepting state q_A be A, we can ask "Does this TM accept?" as:

$$
\begin{aligned}
\phi_A = & \\
& \wedge \phi_{preliminary} \\
& \wedge \phi_T \\
& \wedge \forall t [C_{t+1} = T(C_t)] \\
& \wedge [C_0 = 2^0 3^0 5^w] \\
& \wedge \exists h, \mu, \omega [C_h = 2^A 3^\mu 5^\omega]
\end{aligned}
$$

$Th(\mathcal{N}, +, \times)$ **is Turing complete!**