Exercise 2: Halting Problem

Consider some TM M with input w. We can then create a second TM M_0 that works as follows:

Given any input w_0, M_0 ignores w_0 and simulates running w on M (M is hard-coded into M_0). M_0 halts on all inputs iff M halts on w. We know that the problem of determining whether a TM halts on a single input is undecidable and thus, in general, we cannot decide whether M halts on w. It follows that it is undecidable whether M_0 halts on all inputs.

Exercise 3: Primitive Recursion

Part 1

We first show by induction on n that for any fixed n, $f(x,y) = A(n,x,y)$ is primitive recursive.

Base case: Here, $n = 0$ and $f(x,y) = A(0,x,y) = s(y)$ is clearly primitive recursive. Formally, we can define $f(x,y) = s(\pi_2^2(x,y))$.

Inductive step: Assume that for some $n \geq 0$, $f(x,y) = A(n,x,y)$ is primitive recursive. We show that $f'(x,y) = A(n+1,x,y)$ is primitive recursive by construction. We first construct a function $g(y,x)$ by primitive recursion as follows. Intuitively, we'd like to have $g(y,x) = A(n+1,x,y)$.

$$\begin{aligned}
g(0,x) &= A(n+1,x,0) = \begin{cases} s(z(x)) & \text{if } n+1 \geq 3 \\ z(x) & \text{if } n+1 = 2 \\ \pi_1^1(x) & \text{if } n+1 = 1 \end{cases} \\
g(s(y),x) &= A(n+1,x,s(y)) \\
&= A(n,x,A(n+1,x,y)) \\
&= A(n,x,g(y,x)) \\
&= f(\pi_2^3(y,x,g(y,x)), \pi_3^3(y,x,g(y,x)))
\end{aligned}$$

Since $f(x,y)$ is primitive recursive by the inductive hypothesis, we can conclude that g is primitive recursive. To complete the inductive step, we need to define $f'(x,y) = g(y,x) = g(\pi_2^2(x,y), \pi_1^2(x,y))$

This completes the proof for the first part of the problem.

Part 2

The proof for the second part of the problem is more involved. The intuition is that Ackermann's function grows much more quickly than any primitive recursive function. Formally, we can prove the following. (From here on, we'll use the alternate definition of Ackermann's function that was posted on the web.)

Theorem: Let $f(x_1, \ldots, x_n)$ be a primitive recursive function from $\mathbb{N}^n$ to $\mathbb{N}$. Then there exists some $M \in \mathbb{N}$ such that $f(x_1, \ldots, x_n) < A_M(\max(x_1, \ldots, x_n))$ for all $x_1, \ldots, x_n \in \mathbb{N}$.

For the moment, we assume the theorem to be true and show how it can be used to prove that Ackermann's function is not primitive recursive. The proof will be by contradiction. So we assume that $B(x) = A(x, x)$ is primitive recursive. Then $B(x) + 1$ is also primitive recursive. By the theorem, there exists some $M \in \mathbb{N}$ such that $B(x) + 1 < A_M(x) = A(M, x)$. But note that

$$A(M, M) + 1 = B(M) + 1 < A_M(M) = A(M, M)$$

This is a clear contradiction, so Ackermann's function cannot be primitive recursive.

We now return to proving the above theorem. We'll state and prove four lemmas first.

Lemma 1

Each A_n is strictly increasing, and for all x, $A_n(x) > x$.

Proof: By induction on n. The base case $n = 0$ follows directly from the definition of A_0. So suppose the lemma is true for some A_n. Then we have

$$A_{n+1}(x+1) = A_n^{(x+2)}(1) = A_n(A_n^{(x+1)}(1)) > A_n^{(x+1)}(1) = A_{n+1}(x)$$

which shows that A_{n+1} is strictly increasing. To see that $A_{n+1}(x) > x$, it suffices to note that

$$A_{n+1}(0) = A_n(1) > 1 > 0$$

by the inductive hypothesis. This completes the induction.

Lemma 2

If $n < n'$, then $A_n(x) < A_{n'}(x)$.

Proof: By induction on n; for each n, we will show that for all x, $A_n(x) < A_{n+1}(x)$. For $n = 0$, we have that $A_0(x) = x + 1 < x + 2 = A_1(x)$. So suppose the this is true for some n, i.e. $A_n(x) < A_{n+1}(x)$ for all x. Then note that $A_{n+1}(x) = A_n^{(x+1)}(1) < A_{n+1}^{(x+1)}(1) = A_{n+2}(x)$. This completes the proof.

Lemma 3

$A_n(x + 1) \leq A_{n+1}(x)$

Proof: By induction on x. $A_n(1) = A_{n+1}(0)$ so its true for $x = 0$. Suppose its true for some x. Then

$$A_{n+1}(x+1) = A_n^{(x+2)}(1) = A_n(A_{n+1}(x)) \geq A_n(A_n(x+1)) \geq A_n(x+2)$$

where the first inequality follows from lemma 1 and the inductive hypothesis, and the second follows from the fact that $A_n(x + 1) \geq x + 2$ (which is implied by lemma 1). This finishes the proof.

Lemma 4

$A_n(2x) < A_{n+2}(x)$

Proof: By induction on x. For $x = 0$, we have $A_n(0) < A_{n+2}(0)$ by lemma 2. For the inductive hypothesis, suppose that $A_n(2x) < A_{n+2}(x)$ for some n. We need to show that $A_n(2x + 2) < A_{n+2}(x + 1)$. Note that

$$A_{n+2}(x+1) = A_{n+1}(A_{n+2}(x)) > A_{n+1}(A_n(2x)) \geq A_n(A_n(2x) + 1) \geq A_n(2x + 2)$$

This completes the proof.

Proof of theorem

By induction on the derivation of f. We have 5 cases to consider. (There's a lot of grunge work here, so much of the details will be omitted.)

Base cases

Case 1: $s(x) = x + 1$. We can take $M = 1$ as $s(x) = x + 1 < x + 2 = A_1(x)$.

Case 2: $z(x) = 0$. We can take $M = 0$ as $z(x) = 0 < x + 1 = A_0(x)$.

Case 3: $\pi_i^n(x_1, \ldots, x_n)$. We can take $M = 0$ as $\pi_i^n(x_1, \ldots, x_n) = x_i < A_0(\max(x_1, \ldots, x_n))$.

Induction cases

Case 4: Composition. We have a function $f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_k(\vec{x}))$. By the induction hypothesis, we can find $M, M_1, \ldots, M_k$ such that $g(\vec{y}) < A_M(\max(\vec{y}))$ and $h_i(\vec{x}) < A_{M_i}(\max(\vec{x}))$ (for $1 \leq i \leq k$). Let $N = \max(M, M_1, M_2, \ldots, M_k)$. Using the lemmas above, we can show that $f(\vec{x}) < A_{N+2}(\mathrm{vec}(\vec{x}))$.

Case 5: Primitive recursion. We have a function $f(y, \vec{x})$ where $f(0, \vec{x}) = g(\vec{x})$ and $f(y+1, \vec{x}) = h(y, \vec{x}, f(y, \vec{x}))$ (for some functions g and h). By the induction hypothesis, we can find M and M' such that $g(\vec{x}) < A_M(\max(\vec{x}))$ and $h(y, x, i) < A_{M'}(\max(\vec{x}))$. Let $k = \max(M, M') + 3$. By induction on y, we can show that $f(y, \vec{x}) < A_k(\max(y, \vec{x}))$.

Exercise 4: Self-reference

There are many solutions to this problem. Check out *http://www.nyx.net/∼gthompso/quine.htm* for a bunch of them.

Exercise 5: Partial recursive functions

Throughout this problem, let f be the function given by the Padding Lemma. That is, $\varphi_i = \varphi_{f(i)}$ and $f(i) > i$ for all i. Let $f^{(n)}(i)$ denote f composed with itself n times.

First, we claim that there is a total, injective function $\delta(i, x)$ such that $\varphi_i(x, y) = \varphi_{\delta(i,x)}(y)$, i.e. we can take the function s_1^1 (from the *snm*-property) to be injective. Since $\mathbb{N}^2$ is countable, we can put a well-ordering on it, and without loss of generality, assume $(0, 0)$ is the least element. We define δ recursively then. (We can do this explicitly using primitive recursion by using an injection between $\mathbb{N}^2$ and $\mathbb{N}$.) For the base case, we define $\delta(0, 0) = s_1^1(0, 0)$. Otherwise, $\delta(i, x) = f^{(n)}(s_1^1(i, x))$ where n is the least such that $\delta(i, x) > \delta(i', x')$ for all $(i', x') < (i, x)$. (Such an n must exist since f is a strictly increasing function.) This finishes the construction of δ.

Now, let e be the index of the function

$$\beta(x, y) = \varphi_{\sigma(\delta(x,x))}(y)$$

Therefore, we have that

$$\varphi_e(x, y) = \varphi_{f^{(n)}(e)}(x, y) = \varphi_{\delta(f^{(n)}(e),x)}(y) = \varphi_{\sigma(\delta(x,x))}(y)$$

Then it's clear that $\delta(f^{(n)}(e), f^{(n)}(e))$ is a fixed point of σ. Since we can take $n = 1, 2, 3, \ldots$ and since δ is injective, we get an enumeration of an infinite number of fixed points of σ.