Mathematics 114C, Winter 2019
Yiannis N. Moschovakis
First Midterm, January 31, 2019

Name (last name first): ______________________________

Signature: ______________________________

There are 12 parts of problems, each worth 10 points for a total of 120 and 100 points count for a "perfect" score; so you can skip a part or two or afford to make a small mistake. Some of these parts are trivial and some are not—so do first those that you can do quickly.

There is a blank page at the end which you can use for scratch, and the page before it has a copy of the transition table for the recursive machine.

Solved!

Problem 1. __________

Problem 2. __________

Problem 3. __________

Problem 4. __________

Total: ____________________

Problem 1. Let

$$C(x,y,z) = \mathsf{if}\ (x=0)\ \mathsf{then}\ y\ \mathsf{else}\ z = \begin{cases} y, & \text{if } x = 0, \\ z, & \text{otherwise} \end{cases} \qquad (x,y,z \in \mathbb{N})$$

Determine whether each of the following equations is True or False under the indicated assumptions and circle which it is:

(1a) $\boxed{\text{True}}$ or $\boxed{\text{False}}$: for all partial functions $g(\vec{x}), h_1(\vec{x}), h_2(\vec{x})$ on $\mathbb{N}$,

$$C(g(\vec{x}), h_1(\vec{x}), h_2(\vec{x})) = \mathsf{if}\ (g(\vec{x}) = 0)\ \mathsf{then}\ h_1(\vec{x})\ \mathsf{else}\ h_2(\vec{x})$$

Solution. This fails if $g(\vec{x}) = 0, h_1(\vec{x})\!\downarrow$ and $h_2(\vec{x})\!\uparrow$.

(1b) $\boxed{\text{True}}$ or $\boxed{\text{False}}$: for all partial $g(\vec{x})$ and total $h_1(\vec{x}), h_2(\vec{x})$:

$$C(g(\vec{x}), h_1(\vec{x}), h_2(\vec{x})) = \mathsf{if}\ (g(\vec{x}) = 0)\ \mathsf{then}\ h_1(\vec{x})\ \mathsf{else}\ h_2(\vec{x})$$

Solution. This is true.

(1c) $\boxed{\text{True}}$ or $\boxed{\text{False}}$: for all partial $g(\vec{x})$ and $h(\vec{x})$:

$$C(g(\vec{x}), h(\vec{x}), h(\vec{x})) = \mathsf{if}\ (g(\vec{x}) = 0)\ \mathsf{then}\ h(\vec{x})\ \mathsf{else}\ h(\vec{x})$$

Solution. This is true.

Problem 2. Consider the following recursive program (with just one equation) on the partial algebra $\mathbf{N}_0 = (\mathbb{N}, 0, 1, S, Pd)$ of unary natural numbers,

$$(E) \qquad p(x,y) = \mathsf{if}\ (x=0)\ \mathsf{then}\ 1\ \mathsf{else}\ p(Pd(x), p(x,y))$$

(2a) Show the computation of $\overline{p}(0,3)$ by the recursive machine which starts with the state $p\ :\ 0\ 3$, determine if it converges or diverges and determine the value $\overline{p}(0,3)$ if it converges.

Solution.

$$\begin{array}{r}
p : 0\ 3 \\
\mathsf{if}\ (0=0)\ \mathsf{then}\ 1\ \mathsf{else}\ p(Pd(0), p(0,3)) : \\
1\ p(Pd(0), p(0,3))\ ?\ 0 : \\
1\ p(Pd(0), p(0,3))\ ?\ : 0 \\
1\ : \\
: 1
\end{array}$$

The computation converges and yields $\overline{p}(0,3) = 1$.

(2b) Show the computation of $\overline{p}(2,3)$ by the recursive machine which starts with the state $p\ :\ 2\ 3$, determine if it converges or diverges and determine the value $\overline{p}(2,3)$ if it converges.

Solution.

$$\begin{array}{r}
p : 2\ 3 \\
\mathsf{if}\ (2=0)\ \mathsf{then}\ 1\ \mathsf{else}\ p(Pd(2), p(2,3)) : \\
1\ p(Pd(2), p(2,3))\ ?\ 2 : \\
1\ p(Pd(2), p(2,3))\ ?\ : 2 \\
p(Pd(2), p(2,3))\ : \\
p\ Pd(2)\ p(2,3)\ : \\
p\ Pd(2)\ p\ 2\ 3\ : \\
p\ Pd(2)\ p\ 2\ : 3 \\
p\ Pd(2)\ p\ : 2\ 3
\end{array}$$

At this point the machine will "disregard" the first first two elements $p\ Pd(2)$ of the state (by the Stack Discipline Lemma 2C.1) and repeat the computation to reach the state

$$p\ Pd(2)\ p\ Pd(2)\ p\ :\ 2\ 3$$

then go on to reach the state

$$p\ Pd(2)\ p\ Pd(2)\ p\ Pd(2)\ p\ :\ 2\ 3$$

etc., ad infinitum, in short, it will not converge, so $\overline{p}(2,3)\uparrow$

(Problem 2 continues on the next page)

(Problem 2 continued from the preceding page)

(2c) Give an explicit definition of the partial function $\overline{p}(x, y)$ computed by the recursive program E.

Solution.

$$\overline{p}(x, y) = \begin{cases} 1, & \text{if } x = 0, \\ \uparrow & \text{otherwise, if } x \neq 0. \end{cases}$$

This can be proved by replacing 2 in the preceding part by any $x > 0$.

(2d) Prove that the equation $p(x, y) = \mathsf{if}\ (x = 0)\ \mathsf{then}\ 1\ \mathsf{else}\ p(Pd(x), p(x, y))$ of the program (E) has exactly one total solution and find it.

Solution. The total, constant function $p(x, y) = 1$ clearly satisfies the equation. If $q(x, y)$ is any total solution, we prove by induction on x that $(\forall y)[q(x, y) = 1]$.

Basis, $q(0, y) = 1$, by the definition.

Induction step:

$$p(x + 1, y) = p(x, p(x + 1, y)) = 1 \quad \text{(by the induction hypothesis).}$$

Problem 3. Suppose $g(x), h(u,v), \sigma(x), \tau(x,y)$ are total functions on $\mathbb{N}$.

(3a) Prove that there is exactly one total function $f(x,y)$ such that for all x, y,

$$\begin{aligned} f(0,y) &= g(y), \\ f(x+1,0) &= f(x,\sigma(x)), \\ f(x+1,y+1) &= h(f(x,f(x+1,y)),\tau(x,y)) \end{aligned}$$

Solution. Let $\overline{p}(x,y)$ be the partial function computed by the program

$$\begin{aligned} p(0,y) = \mathsf{if}\ (x=0)\ \mathsf{then}\ g(y) \\ \mathsf{else}\ \ \mathsf{if}\ (y=0)\ \mathsf{then}\ p(Pd(x),\sigma(Pd(x))) \\ \mathsf{else}\ h(p(Pd(x),p(x,y)),\tau(Pd(x),Pd(y))) \end{aligned}$$

in the algebra $(\mathbb{N}, 0, 1, S, Pd, g, \sigma, h, \tau)$. It satisfies the given system and it is quite easy to prove by induction on x that $(\forall y)[\overline{p}(x,y)\downarrow]$.

(3b) True or false: if g, h, σ, τ are recursive functions, then f is also recursive.

Solution. True, by the Transitivity Theorem.

(3c) True or false: if g, h, σ, τ are primitive recursive functions, then f is also primitive recursive.

Solution. False: a special case of this definition is the Ackermann function, which is not primitive recursive; take

$$g(y) = y+1,\ \sigma(x) = 1,\ \tau(x,y) = 1,\ h(u,v) = u.$$

Problem 4. The *iterates* $\phi^i : M \to M$ of a (total) function $\phi : M \to M$ are defined for $i \geq 1$ by the recursion

$$\phi^1(x) = \phi(x), \quad \phi^{i+1}(x) = \phi(\phi^i(x)), \ldots$$

so that $\phi^2(x) = \phi(\phi^1(x)) = \phi(\phi(x)),\ \phi^3(x) = \phi(\phi(\phi(x))), \ldots$

Suppose $\mathsf{M} = (M, 0, 1, g, h)$ is a total algebra with both $g, h : M \to M$ unary, and define the partial function $f : M \rightharpoonup M$ by

$$f(x) = g^n(x) \text{ where } n = (\mu i \geq 1)[h^i(x) = 0]$$

For example, if $h(x) \neq 0, h^2(x) \neq 0, h^3(x) = 0$, then $f(x) = g^3(x)$, and if $h^i(x) \neq 0$ for all i, then $f(x)\uparrow$.

(4a) Specify a recursive program of M which computes $f(x)$.

Careful: M is an arbitrary set, not necessarily $\mathbb{N}$, and so we cannot use functions on $\mathbb{N}$.

Solution. The required system has two equations,

$$\begin{aligned} p(x) &= q(x, x), \\ q(x, y) &= \mathsf{if}\ (h(y) = 0)\ \mathsf{then}\ g(x)\ \mathsf{else}\ q(g(x), h(y)) \end{aligned}$$

The claim is that

$$\overline{q}(x, y) = g^n(x) \text{ where } n \text{ is least such that } h^n(y) = 0,$$

so that $f(x) = \overline{q}(x, x) = \overline{p}(x)$

(Problem 4 continues on the next page)

(Problem 4 continued from the preceding page)

(4b) Prove the claim you made in part (4a), that the recursive program you specified actually computes the partial function $f(x)$.

SOLUTION. We will prove that for all x, y

$$\overline{q}(x,y) = g^n(x) \text{ where } n \text{ is least such that } h^n(y) = 0,$$

using the following two lemmas:

Lemma 1. For all x, y, if $h^i(y) \neq 0$ for all i, then $\overline{q}(x,y)\uparrow$.

Assume the hypothesis and look at the computation of $\overline{q}(x,y)$ by the recursive machine:

$$q \;:\; x\; y \;\rightarrow\; \textsf{if } (h(y)=0) \textsf{ then } g(x) \textsf{ else } q(g(x),h(y)) \;:$$
$$\cdots \rightarrow q(g(x),h(y)) \;\cdots\; \rightarrow q(g^2(x),h^2(y)) \;\rightarrow \ldots$$

This clearly goes on forever.

Lemma 2. For all n,

> *for all x and all y, if $n = n_y = \mu i[h^i(y)=0]$, then the recursive machine started on $q \;:\; x\; y$ terminates in the state $:\; g^n(x)$.*

This is proved by induction on $n \geq 1$.

Basis: $n = 1$, i.e., $h(y) = 0$; then easily $q \;:\; x\; y \;\rightarrow\; \cdots \;\rightarrow :\; g(x)$ in a few steps.

Induction step, $n > 1$. The key observation is that for all y,

$$h(y) \neq 0 \Longrightarrow \Big(n_y = \mu i[h^i(y)=0] = n_{h(y)} + 1\Big), \text{ where } n_{h(y)} = \mu i[h^i(h(y)) = 0] + 1$$

so $n_{h(y)} = n_y - 1 < n_y$. It follows that the induction hypothesis applies to $g(x)$ and $h(y)$, so starting on $p \;:\; g(x)\; h(y)$ it will eventually stop in the terminal state $:\; g^n(g(x)$, and, of course, $g^n(g(x) = g^{n+1}(x)$. At the same time, a simple computation shows that (as $h(y) \neq 0$), if we start the machine on $p \;:\; x\; y$ it will reach the state $p \;:\; g(x)\; h(y)$ and then go on to the correct value.

The two lemmas together imply that for all x, y,

$$\overline{q}(x,y) = g^n(x) \text{ where } n = (\mu i \geq 1)[h^i(y) = 0];$$

because if $h^i(y) \neq 0$ for all i, then both sides of this equation diverge by Lemma 1; and if there is some i such that $h^i(y) = 0$, then the two sides converge to the same value by Lemma 2.