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

Name (last name first): Stone, Jaclyn

Signature: Jaclyn Stone

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.

Problem 1.	30
Problem 2.	40
Problem 3.	25
Problem 4.	–
Total:	95

Problem 1. Let

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

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

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

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

✓

Since = means they either converge to the same value or diverge

✓

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

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

✓

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

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

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,

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

10

(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.

p: 0 3

if(0=0) then 1 else p(Pd(0), P(0,3)) :

1 p(Pd(0), P(0,3)) ? 0 :

1 :

: 1 $\rightarrow$ $\boxed{P(0,3) = 1}$

2

(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.

p: 2 3

if(2=0) then 1 else p(Pd(2), P(2,3))

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 $\rightarrow$ infinite loop $\rightarrow$ diverges

(Problem 2 continues on the next page)

(Problem 2 continued from the preceding page)

10 **(2c)** Give an explicit definition of the partial function $\overline{p}(x, y)$ computed by the recursive program E.

$$P(x,y) = \begin{cases} 1 & \text{if } x=0 \\ P(x,y)\uparrow & \text{o/w} \end{cases}$$

Note that if the left argument was computed first $P(x,y)$ would equal 1 $\forall x$ ~~nonzero~~, but since our two stack computation method computes the right argument first, it gets stuck for any nonzero x.

I bet you (and can prove!) there is no correct compiler for languages with strict recursion that would make $\overline{p}(1,3)$! It is not the order of evaluation, but that both arguments must be computed!

10 **(2d)** Prove that the equation $p(x, y) = \text{if } (x = 0) \text{ then } 1 \text{ else } p(Pd(x), p(x, y))$ of the program (E) has exactly one total solution and find it.

$f(x,y) = 1$

~~Guaranteed solution by the soundness theorem~~

In $\overline{p}(x, p(\ldots$

If $\exists$ another solution $f'(x,y)$, we know

$\forall y \; f'(0,y) = f(0,y) = 1$

Assume $\forall y \; f'(x,y) = f(x,y)$

Then $f'(x+1,y) = \underbrace{f'(x, f'(x,y))}_{f'(x,z)}$, $f(x+1,y) = \underbrace{f(x, f'(x,y))}_{f(x,z)}$

$f'(x,z)$ equal by assumption $f(x,z)$

So $f'(x+1,y) = f(x+1,y)$

$\boxed{\text{So } f'(x,y) = f(x,y) \;\; \forall x}$

So unique

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,

5

$$f(0,y) = g(y),$$
$$f(x+1,0) = f(x,\sigma(x)),$$
$$f(x+1,y+1) = h(f(x, f(x+1,y)), \tau(x,y))$$

By nested recursion lemma $\exists$ a solution $f(x,y)$

If $\exists$ $f'(x,y)$ another solution, $f'(0,y) = f(0,y)$ $\forall y$

and so by induction, assume

$f'(x,0) = f(x,0)$, then $f'(x+1,0) = f'(x,\sigma(x)) = f(x,\sigma$

check scratch page ☆

But why is there an f — even partial — to begin with?

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

10 True

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

10 False

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

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

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

Suppose $\mathbf{M} = (M, 0, 1, g, h)$ 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 $\mathbf{M}$ which computes $f(x)$.

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

$P_1(x) = \text{if}(x=1) \text{ then } g(x), \text{ else } g(P_1(Pd(x)))$

$P_2(x) = \text{if } (x=1) \text{ then } h(x). \text{ else } h(P_2(Pd(x)))$

$P_3(x) = \text{if}(P_2(1)=0) \text{ then } x \text{ else } P_3(S(x))$

$P_4(x) = P_1(P_3(x))$

Not sure how to do this without S, and Pd

What is Pd?

(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 $f(x)$.

(pass)	$\alpha\ \underline{x :}\ \beta \ \to \alpha\ \underline{: x}\ \beta \quad (x \in M)$
(e-call), in general	$\alpha\ \underline{\mathsf{f}_i : \vec{x}}\ \beta \ \to \alpha\ \underline{: f_i(\vec{x})}\ \beta$
(e-call), for N_0	$\alpha\ \underline{S : x}\ \beta \to \alpha\ \underline{: x+1}\ \beta$ $\alpha\ \underline{Pd : x}\ \beta \to \alpha\ \underline{: x \dot{-} 1}\ \beta$
(i-call)	$\alpha\ \underline{\mathsf{p}_i : \vec{x}}\ \beta \ \to \alpha\ \underline{E_i\{\vec{\mathsf{x}}_i :\equiv \vec{x}\} :}\ \beta$
(comp)	$\alpha\ \underline{h(A_1, \dots, A_n) :}\ \beta \ \to \alpha\ \underline{h\ A_1 \cdots A_n :}\ \beta$
(br)	$\alpha\ \underline{(\text{if } (A = 0) \text{ then } B \text{ else } C) :}\ \beta \ \to \alpha\ \underline{B\ C\ ?\ A :}\ \beta$
(br0)	$\alpha\ \underline{B\ C\ ? : 0}\ \beta \ \to \alpha\ \underline{B :}\ \beta$
(br1)	$\alpha\ \underline{B\ C\ ? : y}\ \beta \ \to \alpha\ \underline{C :}\ \beta \quad (y \neq 0)$

- The underlined words are those which change in the transition.
- $\vec{x} = x_1, \dots, x_n$ is an n-tuple of individual constants.
- In the *external call* (e-call), $f = f_i$ is a primitive partial function of M with arity$(f_i) = n_i = n$.
- In the *internal call* (i-call), p_i is an n-place recursive variable of the program E which is defined by the equation $\mathsf{p}_i(\vec{\mathsf{x}}) = E_i$.
- In the *composition transition* (comp) h is a function symbol (constant or variable) with arity$(h) = n$.

TABLE 1. Transitions of the system $\mathcal{T}(E, \mathsf{M})$.

For example, if one of the equations of E in N_0 is the explicit equation

$$p(x) = S(S(x)),$$

then, for every x, $\overline{p}(x) = x + 2$ by the following computation:

$$p : x \ \to\ S(S(x)) : \ \to\ S\ S(x) : \ \to\ S\ S\ x :$$
$$\to S\ S\ :\ x \to S\ :\ x+1 \to\ :\ x+2$$

~~$f(0,0) = f'(0,0)$~~

~~Assume $f(x,0) = f'(x,0)$~~

~~Then $f(x+1,0) = f(x,\sigma(x))$, $f'(x+1) = f'(x,\sigma(x))$~~

☆ Induction

$f(0,0) = f'(0,0)$

$f(0,y) = f'(0,y) = g(y) \quad \forall y$

So now ~~to~~ assume $f(x,y) = f'(x,y) \ \forall y$

And we NTS that $f(x+1,y) = f'(x+1,y)$

. So by another induction

$f(x+1,0) = f(x,\sigma(x))$, $f'(x+1,0) = f'(x,\sigma(x))$

Since $f(x,y) = f'(x,y)$, $f(x+1,0) = f'(x+1,0)$

So assume by subsidiary induction that

$f(x+1,y) = f'(x+1,y)$

Now $f(x+1,y+1) = h(f(x,f(x+1,y)), \tau(x,y))$

$f'(x+1,y+1) = h(f'(x,f'(x+1,y)), \tau(x,y))$

Since $f(x+1,y) = f'(x+1,y) = z$

so $h(f(x,z)\cancel{z}, \tau(x,y))$, but by our main assumption

$f(x,y) = f'(x,y) \ \forall y$ so no matter if $z > y$, since

it is $f(x,z)$ and not $f(x+1,z)$ this is ok,

so

$$h(f(x,f(x+1,y)), \tau(x,y)) = h(f'(x,f'(x+1,y)), \tau(x,y))$$

so $f(x+1,y+1) = f'(x+1,y+1)$

So $f(x,y) = f'(x,y) \ \forall x \ \forall y$ □

So $f(x,y)$ is unique

This seem correct, but the matter is not that complex: It is a double induction almost exactly like the proof of commutativity of +