Goldowsky Boris Phill 9 4

PRINT IN LARGE LETTERS (Family Name) (First Name) (Middle Name) (Course) (Year)

6.044J / 18.423J (Computability, Programming, & Logic) 12/

(Subject Number) (Subject Name) (Date)

Meyer

(Instructor's Name)

IF EXAMINATION FOR ADVANCED STANDING CHECK HERE ☐

IF A CONDITION EXAMINATION, FILL IN BELOW		IF A POSTPONED FINAL EXAMINATION, FILL IN BELOW	
(Year and term taken in Class)	(Instructor's Name)	(Year and term taken in Class)	(Instructor's Name)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Do not write below. Reserved for Examiner.

1. 20
2. 07
3. 25
4. (a) 3 (b) 3 (c) 4 (d) [redacted]07 total : [redacted] 17
5. (a) 14 (b) 1 total : 15
6. (a) 9 (b) 10 total : 19
7. 30
8. (a) 5 (b) 10 (c) [redacted] 5 total : [redacted] 20 am
9.
10.
11.
12.
13.
14.
15.

153/200

6-72 30M S14

1. Assume there was such a set:

R is recursive $\rightarrow$ There is a machine that decides membership in R.

~~$A \subseteq R$~~ $K_a \subseteq R$

$K_b \subseteq \overline{R}$

Given the machine that decides membership in R, it is easy to construct a machine that computs the following function:

$$f(x) = \begin{cases} b & \text{if } x \in R \\ a & \text{if } x \notin R \end{cases}$$ call this machine M.

Question: ~~[illegible]~~ is $M \in R$?

$\rightarrow$ if yes, ~~[illegible]~~ $f(M) = b$

so M on $d(M)$ outputs b.

so $M \in K_b \subseteq \overline{R}$

so $M \notin R$. * contradiction

20/20

$\rightarrow$ if no, $f(M) = a$

M outputs a on $d(M)$

$M \in K_a \subseteq R$

$M \in R$ * contradiction

So the assumption that a recursive set separating K_a and K_b exists must have been false.

(2)

(a) ~~Since every step of the conversion from Post Machine to PCP is~~
No step of the infinite computation of the Post Machine will cause you to write down mismatched strings in the IPCP. However, since a HALT ACCEPT/REJECT box is never reached, the only solution is an infinite one, since at every step other than halt you write down an asterisk on the end of the bottom word but not on the top one. ✓

(b) The IPCP can be shown to be non-r.e. by defining a function that reduces the complement of the HP for Post Machines to the IPCP.

Given a post machine M, it fails to halt on input w iff the post system S has an infinite solution, where S is defined as follows:

Label the machine's flowchart + translate it to pairs exactly as in Manna's reduction.

Then an infinite solution to S is the one that picks pairs following the infinite path of the program through the flowchart. This is the only solution since there is only one possible pair at each point. But if there were any point at which no pair was possible to add on to the end, then the Post Machine would also have nowhere to go and would have to have halted.

-18

(3) Given a set Σ of semigroup equations, we need a method to determine in finite time if they are degenerate. (If not degenerate, can loop forever)

since for all $\mathcal{I}$ that satisfy Σ, $\mathcal{I}(u_i) = \mathcal{I}(u_j)$ for all $u_i, u_j \in \{a,b\}^+$,
$\Sigma \models u_i = u_j$ for all $u_i, u_j \in \{a,b\}^+$.
by completeness,
$\Sigma \vdash u_i = u_j$ for all $u_i, u_j \in \{a,b\}^+$
So under the Thue system whose rules are Σ, anything in $\{a,b\}^+$ rewrites to anything else.

In particular, $\Sigma \vdash a = aa$ and $\Sigma \vdash a = b$. These two can be used to prove any other equality (use $a=aa$ to make string the length you want, the $a=b$ to make the proper ones b's).

5/5

So look for a proof of $\{a=aa, a=b\}$ from Σ. If there is one, you can find it in finite time, and report that Σ is degenerate.

because there are only a finite # of transformation to make it a time.

4 (a) $\cancel{\exists x \exists X = (\langle x \rangle \cdot Y)}$

$\neg \exists x \exists Y \left(X \doteq \langle x \rangle \cdot Y \right)$ 3/3

(b) $\exists Y_1 \exists Y_2 \left(X \doteq Y_1 \cdot \langle x \rangle \cdot Y_2 \right)$ 3/3

(c) $\exists X \forall x \left[\exists Y_1 \exists Y_2 \left(X \doteq Y_1 \cdot \langle x \rangle \cdot Y_2 \right) \right]$ 4/4

X has to be finite, so domain is finite if all x's in domain are in X.

(d) Compactness says that if all finite subsets of an infinite set of wffs can be satisfied by an ~~model~~ interpretation M, then M satisfies the whole set.

-3

The claim here is that there is an infinite set off s-wffs which is unsatisfiable, but each finite subset of which is satisfiable.

Here is such a set: (c) above (call it s-wff #0)

union the infinite set s-wff #1 : $\exists x_1 (x_1 = x_1)$

s-wff #2 : $\exists x_1 \exists x_2 (x_1 \neq x_2)$

s-wff #3 : $\exists x_1 \exists x_2 \exists x_3 (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3)$

⋮

s-wff # n $\exists x_i \left(\bigwedge_{i \neq j} x_i \neq x_j \right)$ where i, j take on all values from 1 to n.

⋮

07/10

Any finite subset is satisfiable by a model with at least n elements in its domain (where n is the number of the highest numbered s-wff in the subset), as long as the domain is finite (to satisfy s-wff #0 if it is in the subset). However the whole set would have to have a finite domain (to satisfy s-wff #0) and an infinite domain (to satisfy the rest). This is impossible.

7

1 $\text{true} \supset \exists Z(\epsilon \doteq Z\cdot Z)$ — Lemma

2 $\{\exists Z(\epsilon \doteq Z\cdot Z)\}\ Y:=\epsilon\ \{\exists Z(Y \doteq Z\cdot Z)\}$ — Assignment Axiom

3 $\{\text{true}\}\ Y:=\epsilon\ \{\exists Z(Y \doteq Z\cdot Z)\}$ — 1,2 Consequence

~~[illegible]~~

4 $\exists Z(Y \doteq Z\cdot Z) \supset (a=a \wedge \exists Z(Y \doteq Z\cdot Z))$ — Lemma

5 $\{a=a \wedge \exists Z(Y \doteq Z\cdot Z)\}\ X:=a\ \{X=a \wedge \exists Z(Y \doteq Z\cdot Z)\}$ — Assignement ax

6 $\{\exists Z(Y \doteq Z\cdot Z)\}\ X:=a\ \{X=a \wedge \exists Z(Y \doteq Z\cdot Z)\}$ — 4,5 Consequence

7 $\exists Z(Y \doteq Z\cdot Z) \equiv \neg\exists Z(\langle a\rangle\cdot Y \doteq Z\cdot Z)$ — lemma

8 $\{(X=a\wedge)\neg\exists Z(\langle a\rangle\cdot Y \doteq Z\cdot Z)\}\ Y:=\langle a\rangle\cdot Y\ \{(X=a\wedge)\neg\exists Z(Y \doteq Z\cdot Z)\}$ — Assig. Ax

9 $\{(X=a\wedge)\exists Z(Y \doteq Z\cdot Z)\}\ Y:=\langle a\rangle\cdot Y\ \{(X=a\wedge)\neg\exists Z(Y \doteq Z\cdot Z)\}$ — 7,8 Consequence

10 $\{(X=a\wedge)\exists Z(\langle a\rangle\cdot Y \doteq Z\cdot Z)\}\ Y:=\langle a\rangle\cdot Y\ \{(X=a\wedge)\exists Z(Y \doteq Z\cdot Z)\}$ — Assg. Ax.

11 $\{\neg\exists Z(Y \doteq Z\cdot Z)\}\ Y:=\langle a\rangle\cdot Y\ \{\exists Z(Y \doteq Z\cdot Z)\}$ — 7,10 Consequence

~~[illegible]~~

12 $[(X=a\wedge)\neg\exists Z(Y \doteq Z\cdot Z) \wedge Y \doteq X] \supset [(b=b)\wedge\neg\exists Z(Y \doteq Z\cdot Z)]$ — Lemma

~~[illegible]~~

13 $[\neg\exists Z(Y \doteq Z\cdot Z \wedge X=b)] \supset [\exists Z(X \doteq Z\cdot Z) \equiv (X=a)]$ — Lemma

14\. $\{\neg\exists Z(Y \doteq Z\cdot Z) \wedge b=b\}\ X:=b\ \{\neg\exists Z(Y \doteq Z\cdot Z) \wedge X=b\}$ — Assg Ax

15\. $\{(X=a\wedge)\neg\exists Z(Y \doteq Z\cdot Z) \wedge Y \doteq X\}\ X:=b\ \{\exists Z(X \doteq Z\cdot Z) \equiv (X=a)\}$ — 12,13,14 Consequence (twice)

16\. $[(X=a)\wedge\neg\exists Z(Y \doteq Z\cdot Z)\wedge Y \neq X] \supset [(X=a\wedge)\neg\exists Z(Y \doteq Z\cdot Z)]$ — Lemma

17\. $[X=a \wedge \exists Z(Y \doteq Z\cdot Z)] \supset [\exists Z(Y \doteq Z\cdot Z) \equiv (X=a)]$ — lemma

((5) at end →)

(6) (a) A term was defined as

$$T ::= T \cdot T \mid mkseq(t) \mid X.$$

if $\mathfrak{X}$ is Λ-free, then if T is X $I_v(T) = I_v(X) \neq \Lambda$.
mkseq(t) always returns a 1-element list, and so it cannot return the 0-element Λ.

9/10

(-1 for notational inaccuracies: T should be $(T)_I$, etc here)

And inductively, if $(T_1)_I \neq \Lambda$ and $(T_2)_I \neq \Lambda$, $(T_1 \cdot T_2)_I$ cannot be Λ because it must have length(T_1) + length(T_2) elements.

(b) If X is going to end up $= \Lambda$, it must either start out $= \Lambda$ or be set to Λ by an assignment statement. ✓

We can't assume it starts out as Λ because ~~it must end as Λ~~ this must work over all interpretations, and in most interps $X \neq \Lambda$. And the assignments statement ~~X := Term~~ $X :=$ Term cannot set X to Λ because no term can be Λ by part (a) above.

(-5) implicit induction on # of assignment statements executed should be explicit.

(1) It could be a different m for each finite subset

(2) The final model satisfying the infinite set is ~~not~~ (in general) yet another m'.

8 (a) To determine if W halts in some (Z_n, I_0) simply run W. So to enumerate all the n's for which ~~the~~ ~~[illegible]~~ W halts, simulate W running on Z_1 for a step, then on Z_2 and Z_1 each for a step, then Z_3, Z_2, Z_1 each for a step, and in this manner simulate all n's in parallel until the ones that will halt do halt. If W halts on Z_n, $n \in \text{Spectrum}(W)$

5/5

(b)

```
x := 0;
while x ≠ 0 do
    x := x+1;
    X := <0>·X
od;
X := <0>·X
```

10/10

(c) You could build a while program (swps) that determines what Z_n ~~[illegible]~~ it is working in (as in part (b)) and then determines if $n \in$ the RE set it is interested in, and if it is, halts. You can build one of these for any r.e. set at all, since while programs can compute any computable function. ~~tia~~ −10

[redacted]

05/15

→ more details and explanation required.

18. $\{x{=}a \wedge \neg\exists Z(Y \doteq Z\cdot Z) \wedge Y \not\doteq X\}\ Y := \langle a\rangle \cdot Y\ \{\exists Z(x \doteq Z\cdot Z) \equiv (x=a)\}$ 16,17,11 Conseq.

19. $\{x=a \wedge \neg\exists Z(Y \doteq Z\cdot Z)\}$ if $Y \doteq X$ then $x:=b$ else $Y := \langle a\rangle \cdot Y$ fi $\{\exists Z(x \doteq Z\cdot Z) \equiv (x=a)\}$
15,18 Conditional

20. $\{x=a \wedge \exists Z(Y \doteq Z\cdot Z)\}\ Y := \langle a\rangle \cdot Y$; if $Y \doteq X$ then $x:=b$ else $Y := \langle a\rangle \cdot Y$ fi $\{\exists Z(x \doteq Z\cdot Z) \equiv (x=a)\}$
9,19 Concatenation

21. $(x=a \wedge \exists Z(Y \doteq Z\cdot Z) \wedge Y \not\doteq X) \supset (x=a \wedge \exists Z(Y \doteq Z\cdot Z))$ lemma

22. $\{x=a \wedge \exists Z(Y \doteq Z\cdot Z) \wedge Y \not\doteq X\}\ Y := \langle a\rangle \cdot Y$; if .. then .. else .. fi $\{R\}$ 20,21 Conseq.

R abbreviates $\exists Z(x \doteq Z\cdot Z) \equiv (x=a)$

23. $\{x=a \wedge \exists Z(Y \doteq Z\cdot Z)\}$

while $Y \not\doteq X$ do
    $Y := \langle a\rangle \cdot Y$;
    if $Y \doteq X$ then $x:=b$ else $Y := \langle a\rangle \cdot Y$ fi
od $\{R\}$

24. $\{true\}\ Y := \epsilon;\ x := a\ \{x=a \wedge \exists Z(Y \doteq Z\cdot Z)\}$ 3,6 concatenation

25. $\{true\}$
$Y := \epsilon$; $x := a$
while $Y \not\doteq X$ do
    $Y := \langle a\rangle \cdot Y$;
    if $Y \doteq X$ then $x:=b$ else $Y := \langle a\rangle \cdot Y$ fi
od.
$\{\exists Z(x \doteq Z\cdot Z) \equiv (x=a)\}$.

(5) (a) Given any wff, we wish to determine if it is <u>valid over</u> the structure $(\{a,b\}^*, a, b, \cdot)$, and show that it is iff some S-wff is <u>valid</u>.

The wff expresses some property of elements of $\{a,b\}^*$ (its domain).
How to construct the swff:

what is the signature? are you using a, b?

1. Put the wff into Prenex form
2. Make all wff variables into sequence variables.
3. Insert $S \supset$ between the quantifiers and the body of the formula, where S is a predicate demanding that all elements of all sequence variables are one of no more than 2 distinct elements of the domain.

be more explicit

Thus all sequence variables are equivalent to strings made of only <u>a</u> and <u>b</u>.

14/20

4. Concatenation becomes sequence-variable concat.
5. Prefix the formula by $Q \supset$, where Q is a predicate demanding that <u>a</u> and <u>b</u> (sequence variables of length 1) are different.

This eliminates domains of size 1 or valuations with $a = b$.

This formula will be valid in s-logic iff the original wff was valid over the model $(\{a,b\}^*, a, b, \cdot)$.

1/5

(b) Since you cannot prove all true statements about a and b over concatenation, you

define

cannot <u>prove</u> all sequence-logic formulas (by the many-one reduction). If you could, just Xlate arbitrary wff to sequence-wff, and prove <u>that</u>.