Final Exam

This exam is OPEN BOOK. You have three hours. Good luck!

Problem 1. Consider the body

```
(define u (lambda () (lambda (x) (* 2 x v))))
(define v 4)
(define w (u))
(define x (+ 1 v)
(define y (lambda () x))
(define z (lambda () y))
(cons v w)
```

1(a) Circle the variables whose definitions in the body above can be garbage-collected at the next step in the rewriting model of Scheme:

u v w x y z

1(b) Circle the variables whose definitions will have been garbage-collected when the evaluation of the body is complete:

u v w x y z

Problem 2. Consider an equation in Scheme of the following form:

$$\texttt{(if } M \ N \ N\texttt{)} = N.$$

2(a) Let N be the variable `z`. Circle the expressions below which can be substituted for M so that the above equation holds as an observational congruence in functional Scheme:

```
#t
#f
(spin)
(null? (list N))
(zero? (- N N))
(null? N)
(null? y)
N
(N)
```

2(b) Now let N be `(z)`. Circle the expressions below which can be substituted for M so that the above equation holds as an observational congruence in *functional* Scheme:

```
#t
#f
(spin)
(null? (list N))
(zero? (- N N))
(null? N)
(null? y)
N
(N)
```

2(c) Mark with a check those expressions in part 2(b) which keep the equation holding as an observational congruence even in Scheme *with side-effects*.

Problem 3. Show that every recursively enumerable (r.e.) set of expressions is many-one reducible to the Halting Problem for Scheme.

Problem 4. We consider strings over the symbols $\{r, l\}$ with the interpretation that r means the 90^o clockwise rotation of a square, l means the 90^o counterclockwise rotation, and string concatenation denotes composing rotations.

4(a) How many different values of strings are there under this interpretation?

4(b) Describe "canonical forms" for the different values.

4(c) Write out a set of equations which are a *complete* set of axioms for this interpretation and briefly explain why they are.

Problem 5. We showed in class that the set of string equations which follow from a *finite* set of string equations is r.e. Explain why the set of string equations which follow from an *r.e.*, possibly infinite, set of string equations is still r.e.

Problem 6. We consider (first-order) assertions over the *arithmetic signature* consisting of the binary function symbols, $+, \times$ and constants 0,1.

6(a) Write an arithmetic assertion which is true under the the nonnegative integer interpretation, but false under the integer interpretation.

6(b) Write an arithmetic assertion which is true under the the real number interpretation, but false under the rational number interpretation.

6(c) Write an arithmetic assertion which is true under the the integers modulo three interpretation, but false under the integers mod two interpretation.

Problem 7. We now consider assertions over the signature consisting of a single binary function symbol, f.

7(a) Write an assertion whose meaning is precisely that f is an associative operator.

We can capture the meaning of string equations using such assertions as follows.

Any string of symbols, s, can be represented by a term, f-term(s), over the f-signature by explicitly writing f for string concatenation, replacing each symbol, a, by a corresponding variable, x_a, and associating to the right. For example,

$$f\text{-term}(abddc) = f(x_a, f(x_b, f(x_d, f(x_d, x_c)))).$$

Now, for any particular strings s_i, t_i, s, t over the alphabet $\{a, b, c\}$, for example, the claim that the set of string equations $\{s_1 = t_1, \ldots, s_n = t_n\}$ implies $s = t$ is equivalent to claiming the *validity* of the assertion saying:

> ["f is associative" and
> "for all x_a, x_b, x_c, $(f\text{-term}(s_1) = f\text{-term}(t_1) \wedge \ldots f\text{-term}(s_n) = f\text{-term}(t_n))$]
> $\Longrightarrow$ f-term$(s) = f$-term(t)."

7(b) Write out, without abbreviations, the assertion over the signature f which means that

$$\{abb = ca, cc = aba\} \models aac = cb$$

7(c) Conclude that the Uniform String Equation Problem is many-one reducible to the Validity Problem for assertions.

7(d) Conclude that the Validity Problem for assertions is not decidable.

Problem 8. We now consider assertions over the string interpretation.

8(a) It might seem even easier to reduce the string equation problem to the *truth* of assertions over the string interpretation. For example, try reducing the question whether

$$abb = ca, cc = aba \models aac = cb$$

to the question whether the assertion

$$[abb = ca \wedge cc = aba] \implies aac = cb$$

is true over the interpretation with domain $\{a, b, c\}^*$. Briefly explain why this does not work at all.

8(b) Even though the reduction above does not work, the Uniform String Equation Problem really is many-one reducible to the Truth Problem for assertions over the string interpretation. Explain why. (*Hint*: Appeal to the theorem from class that every r.e. set of strings is *expressible* by an assertion over the string interpretation.)

8(c) Prove that the Truth Problem for assertions over the string interpretation is *not* many-one reducible to the Uniform String Equation Problem.