Final Exam

This is a closed book final exam. There are six (6) problems of approximately equal weight. You have three hours to complete this exam. Good luck!

Problem 1 [35 points]. (Equational Proof Systems). There are four parts to this problem, but the last part has double weight.

In this problem we develop another proof system for polynomial equations, based on rewriting. For your reference, the axioms and rules of the polynomial proof system we studied in Handout 3, and homework Problem 3 of that Handout, are given in the Appendix to this exam. We continue to use the notation

$$\vdash p = q$$

to indicate that $p = q$ is provable in the system of Handout 3.

The new proof system has only one inference rule: if a series of rewritings can change p into q, then $p = q$ is proved. That is,

$$\frac{p \longrightarrow^* q}{p = q}$$

The rewrite rules of the system are that a polynomial which matches either side of the equational axioms of arithmetic (cf. Table 2 in the Appendix) can be replaced by the other side of the axiom. For example, by the identity axiom for $+$,

$$\underline{e * 0} \longrightarrow 0 + (e * 0)$$

and by the inverse axiom for $+$

$$\underline{0} + (e * 0) \longrightarrow ((e * 1) + -(e * 1)) + (e * 0)$$

where, as usual, we have underlined the redex which gets replaced in the rewriting step. Continuing, we have

$$\begin{aligned}
((e * 1) + -(e * 1)) + (e * 0) &\longrightarrow (e * 0) + ((e * 1) + -(e * 1)) \\
&\longrightarrow ((e * 0) + (e * 1)) + -(e * 1) \\
&\longrightarrow (e * (0 + 1)) + -(e * 1) \\
&\longrightarrow (e * 1) + -(e * 1) \\
&\longrightarrow 0.
\end{aligned}$$

We will use the notation

$$\vdash_2 p = q$$

to indicate that $p = q$ is provable in the new proof system. So by the rewritings above, we conclude that

$$\vdash_2 e * 0 = 0.$$

1(a) [5 points]. For the last five rewriting steps above, indicate the redex and the axiom which justifies the rewriting.

1(b) [5 points]. Explain why $\vdash_2 p = q$ implies $\vdash_2 q = p$.

1(c) [10 points]. Show that $\vdash_2 p = q$ implies $\vdash p = q$. *Hint*: Use the appropriate part of Handout 3, Problem 3.

1(d) [15 points]. Conversely, show that $\vdash p = q$ implies $\vdash_2 p = q$.

Problem 2 [35 points]. (Observational Congruence). We review the definition of observational congruence of Scheme expressions:

Definition. Two Scheme expressions are *observationally distinguishable* iff one evaluates to a printable value, and the other does not evaluate to exactly that printable value. Expressions M, N are *observationally congruent*, in symbols, $M \cong N$ iff there is no Scheme context, C, such that $C[M]$ and $C[N]$ are observationally distinguishable.

2(a) [15 points]. Say that two bodies are *procedure-distinguishable* iff *exactly one* of them evaluates to a procedure. Two bodies M, N are *procedure congruent*, in symbols, $M \underset{\text{proc}}{\cong} N$ iff there is no Scheme context, C, such that $C[M]$ and $C[N]$ are procedure-distinguishable. Prove that

$$M \underset{\text{proc}}{\cong} N \text{ iff } M \cong N.$$

2(b) [10 points]. Let M be `(x (y) (y))` and N be `((lambda (z) (x z z)) (y))`. Give a context demonstrating that M and N are *not* observationally congruent in Scheme with side-effects (they are congruent in functional Scheme).

2(c) [10 points]. Briefly explain why there can be no complete proof system for observational congruence in functional Scheme.

Problem 3 [35 points]. (Polynomial Undecidability). As usual we consider multivariate polynomials with integer coefficients whose variables range over the integers. The undecidability of many problems about such polynomials follows from the insight that polynomials can simulate computer programs. We did not show how to do this in class, but for the last part of this problem we *assume* a result which specifies how polynomials act like programs and then indicate one of its consequences for polynomial decidability.

For any polynomial, p, let nn-range(p) be the nonnegative integers in the range of p.

3(a) [5 points]. Explain why nn-range(p) is half-decidable for every p.

3(b) [10 points]. Describe some half-decidable set of nonnegative integers which is not decidable.

3(c) [5 points]. Show that for all $n \geq 0$,

$$n \notin \text{nn-range}(p) \quad \text{iff} \quad \models 1 \leq (p - n)^2.$$

For the last part of this problem, you may assume:

Theorem (Polynomial Programming). For any half-decidable (that is, r.e.) set, A, of nonnegative integers, there is a polynomial, p_A, such that

$$A = \text{nn-range}(p_A).$$

3(d) [15 points]. Conclude that the valid polynomial inequalities are not half-decidable. *Hint*: Choose an undecidable A in the Polynomial Programming Theorem.

Problem 4 [35 points]. (Half-Decidability, Turing Reducibility). Define a relation $\leq_m$ on sets of S-expressions by the rule that $\mathcal{A} \leq_m \mathcal{B}$ iff there is a computable function, f, from S-expressions to S-expressions such that for all S-expressions, S,

1. $f(S)$ is defined, and
2. $S \in \mathcal{A}$ iff $f(S) \in \mathcal{B}$.

4(a) [10 points]. Show that half-decidability inherits downward under $\leq_m$.

4(b) [10 points]. Show that $\mathcal{A} \leq_m \mathcal{B}$ implies $\mathcal{A} \leq_T \mathcal{B}$ (where $\leq_T$ denotes Turing reducibility).

4(c) [15 points]. Give an example of sets $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A} \leq_T \mathcal{B}$ but not $\mathcal{A} \leq_m \mathcal{B}$. *Hint*: Use part 4(a).

Problem 5 [35 points]. (Counter Machines, Half-Decidability). Say that a 2-Counter machine is *1-2-punched* iff it halts started in configuration $(0,1,0)$ and does not halt started in configuration $(0,2,0)$. Let $\mathcal{P}$ be the set of S-expressions representing 1-2-punched 2-Counter machines[1]. Prove that neither $\mathcal{P}$ nor its complement is half-decidable.

Problem 6 [25 points]. (Semigroup Word Problem). In proving the undecidability of the Semigroup Word Problem, we used the $n+3$ symbol alphabet $\{\$, !, 0, \ldots, n\}$ in the word equations corresponding to a 2-Counter Machine of length n. Let 2-SWP be the SWP for words over a two symbol alphabet. Show that 2-SWP is undecidable.

[1] Each instruction in a Counter machine is a single symbol (`inc1`, `dec1`, etc.) or a symbol (`ifr1`, `ifr2`, etc.) followed by two nonnegative integers, and a Counter Machine is a finite sequence of these instructions. So there is a straightforward way to represent Counter Machines by S-expressions.

A Appendix: References for Polynomial Equation Proof System

Table 1: Standard Equational Inference Rules.

$$\frac{}{e = e}\ \text{(reflexivity)} \qquad \frac{e = f}{f = e}\ \text{(symmetry)} \qquad \frac{e = f, \quad f = g}{e = g}\ \text{(transitivity)}$$

$$\frac{e_1 = e_2, \quad f_1 = f_2}{e_1 + f_1 = e_2 + f_2}\ \text{(+-congruence)} \qquad \frac{e_1 = e_2, \quad f_1 = f_2}{e_1 * f_1 = e_2 * f_2}\ \text{($*$-congruence)} \qquad \frac{e = f}{-e = -f}\ \text{($-$-congruence)}$$

Table 2: Equational Axioms for Arithmetic

$(e + f) + g$	$=$	$e + (f + g)$	(associativity of $+$)
$(e * f) * g$	$=$	$e * (f * g)$	(associativity of $*$)
$e + f$	$=$	$f + e$	(commutativity of $+$)
$e * f$	$=$	$f * e$	(commutativity of $*$)
$0 + e$	$=$	e	(identity for $+$)
$1 * e$	$=$	e	(identity for $*$)
$e + (-e)$	$=$	0	(inverse for $+$)
$e * (f + g)$	$=$	$(e * f) + (e * g)$	(distributivity)

For reference; this is NOT a final exam problem:

Problem 3 from Handout 3: For arithmetic expressions e, f and variable x, the *substitution*, $e[x := f]$, of f for x in e is defined by induction on e:

$$\begin{aligned} x[x := f] &= f, \\ c[x := f] &= c \quad \text{for any constant or variable, } c \text{, distinct from } x, \\ (-e_0)[x := f] &= -(e_0[x := f]), \\ (e_0 \ op \ e_1)[x := f] &= e_0[x := f] \ op \ e_1[x := f], \quad \text{where } op \in \{+, *\}\ . \end{aligned}$$

3(a) Prove that

$\vdash f = g$ implies $\vdash e[x := f] = e[x := g]$.

3(b) Prove that

$\vdash e = g$ implies $\vdash e[x := f] = g[x := f]$.