Quiz 1 Solutions

Problem 1 [20 points]. For any arithmetic expression e, the sequence of equalities and inequalities below is a proof in our formal system for proving inequalities. (The Appendix contains all the rules of the system.) For each line, write the line numbers of its antecedents in the space to its left; if a line is an axiom, write its own line number. For example, the required antecedent numbers for lines (1) and (4) have been filled in for you.

(1)	`0 + (-0) = -0`	(1)
(2)	`0 + (-0) = 0`	(2)
(1)	`-0 = 0 + (-0)`	(3)
(3) (2)	`-0 = 0`	(4)
(5)	`0 <= 1`	(5)
(5)	`-1 <= -0`	(6)
(6) (4.5)	`-1 <= 0`	(7)
(7)	`-1 + e <= 0 + e`	(8)
(9)	`0 + e = e`	(9)
(8) (9.5)	`-1 + e <= e`	(10)

Correction made in class: insert lines

(4)	`-0 <= 0`	(4.5)
(9)	`0 + e <= e`	(9.5)

Problem 2 [20 points]. Explain why `( x <= (x * x) )` is not provable in our formal system.

> All the axioms are valid over the real numbers, $\mathbb{R}$, and the inference rules all preserve validity over $\mathbb{R}$, so all provable inequalities must be valid over $\mathbb{R}$. But the inequality above is not valid over $\mathbb{R}$, e.g., it fails for $\mathtt{x} = 1/2$, and hence it is not provable in our system.

Problem 3 [30 points].

3(a) [15 points].

$$\frac{(e \texttt{ <= } f), \quad (g \texttt{ <= } h)}{(\ (e \texttt{ + } g) \texttt{ <= } (f \texttt{ + } h)\ )} \quad (\texttt{++-<=}\text{-congruence})$$

Explain why adding the rule (++-<=-congruence) to our proof system will not change the set of provable inequalities.

We claim that if $\vdash_{\leq} (e \texttt{ <= } f)$ and $\vdash_{\leq} (g \texttt{ <= } h)$, then $\vdash_{\leq} (\ (e \texttt{ + } g) \texttt{ <= } (f \texttt{ +} h)\ )$. So any use of the (++-<=-congruence) rule in a proof could be replaced by an ordinary $\vdash_{\leq}$-proof of its consequent, and there is no need ever to use this rule.

To prove the claim, note that from $\vdash_{\leq} (e \texttt{ <= } f)$, we can conclude that

$$\vdash_{\leq} (\ (e \texttt{ + } g) \texttt{ <= } (f \texttt{ + } g)\ )$$

by (+-<=-congruence). From $\vdash_{\leq} (g \texttt{ <= } h)$, we can conclude that $\vdash_{\leq} (\ (g \texttt{ + } f) \texttt{ <= } (h \texttt{ + } f)\ )$ by (+-<=-congruence) again, and then

$$\vdash_{\leq} (\ (f \texttt{ + } g) \texttt{ <= } (f \texttt{ + } h)\ )$$

by (+-commutativity), ($\leq$-reflexivity), and ($\leq$-transitivity). A final application of the ($\leq$-transitivity) rule to the displayed equations shows that

$$\vdash_{\leq} (\ (e \texttt{ + } g) \texttt{ <= } (f \texttt{ + } h)\ )$$

as required.

3(b) [15 points]. Explain why deleting the rule ($\leq$-antisymmetry) will not change the set of provable inequalities.

The consequent of the rule is an equation, so any use of the ($\leq$-antisymmetry) rule in a proof could be replaced by inserting a purely equational proof of the consequent. Such an equational proof always exists by *completeness* of our equational system.

Problem 4 [15 points].

```
(define l 1)[1]
(define m 2)
(define n 3)
(define add_n (lambda (m) (+ m n)))
(define mul_m_add5+n (lambda (n) (+ (2 3) (add_n (* m n)))))
(mul_m_add5+n ( (lambda (m) (+ m n)) 5 ))
```

4(a) [5 points]. Underline *all* the redexes in this body.

There are seven in all: 1 G, 1 L, and 5 I.

4(b) [10 points]. Write out the body that results from one rewriting at the lambda application redex followed by all possible applications of the garbage collection rule.

```
(define m 2)
(define n 3)
(define add_n (lambda (m) (+ m n)))
(define mul_m_add5+n (lambda (n) (+ (2 3) (add_n (* m n)))))
(define m_1 5)
(mul_m_add5+n (+ m_1 n))
```

Problem 5 [15 points]. Give an example of a functional kernel Scheme body that is

5(a) [5 points]. an ⟨immediate value⟩ that is not terminated.

`(lambda (x)` M`)`

where M is *any* body which is not terminated, e.g.,

```
(lambda (x) (+ 0 1))
```

5(b) [5 points]. a terminated expression that is not an ⟨immediate value⟩.

```
(2 3)
```

or any other such "error-causing" primitive application.

[1] Technically, the *whole body* is the redex of the garbage collection rule, but this particular definition is the part of the redex which gets erased.

5(c) [5 points]. a body that *cannot* reach a terminated body by repeated rewriting.

```
(define f (lambda (x) (f x))) (f 1)
```

```
( (lambda (x) (x x)) (lambda (x) (x x)) )
```

are representative examples.