Computer Science 131, Fall 2000
Sample Solution for Assignment 2

September 26, 2000

1 Warm-up (10%)

1. (a)

```
let x be 3̅ in
  let f be (fun g(y:Int):Int is x+y) in
    let x be 4̅ in
      f(x)
```

(b)

```
let f be (fun g(y:Int):Int is 3̅+y) in
  let x be 4̅ in
    f(x)
```

(c)

```
let x be 4̅ in
  (fun g(y:Int):Int is 3̅+y)(x)
```

(d)

```
(fun g(y:Int):Int is 3̅+y)(4̅)
```

(e)

```
3̅+4̅
```

(f)

```
7̅
```

2. (a)

```
let fact be (fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))
in
    fact 2̅
```

(b)

```
(fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))(2̅)
```

(c)

```
if 2̅<1̅ then 1̅
   else 2̅*(fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))(2̅ - 1̅)
```

(d)

```
2̅*(fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))(2̅ - 1̅)
```

(e)

```
2̅*(fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))(1̅)
```

```
(f)     2̅*(if 1̅<1̅ then 1̅
             else (fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))(1̅ - 1̅))

(g)     2̅*(1̅*(fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))(1̅ - 1̅))

(h)     2̅*(1̅*(fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))(0̅))

(i)     2̅*(1̅*(if 0̅<1̅ then 1̅
             else (fun g(y:Int):Int is if y<1̅ then 1̅ else y*g(y-1̅))(0̅ - 1̅)))

(j)     2̅*(1̅*1̅)

(k)     2̅*1̅

(l)     2̅
```

2 Adding pairs to NQSML. (65%)

Consider the following extension of NQSML of adding pairs. The abstract syntax is extended as follows:

$$
\begin{array}{rcl}
v & ::= & \cdots \\
 & \mid & \langle v_1, v_2 \rangle \\
 & & \\
e & ::= & \cdots \\
 & \mid & \langle e_1, e_2 \rangle \\
 & \mid & e.1 \\
 & \mid & e.2 \\
 & & \\
t, u & ::= & \cdots \\
 & \mid & t_1 \times t_2
\end{array}
$$

A pair of values is considered a value. The new syntax allows creation of a pair, and operations to project out the first or second component of a pair. Unlike SML, there is no pattern-matching for pairs, so in this abstract syntax a function to raise an integer to an positive integer power would look like:

```
fun p(arg:Int×Int):Int is
  let x be arg.1 in
     let n be arg.2 in
       if (n < 1̅) then 1̅ else p(⟨x,n − 1̅⟩)
```

The new typing rules are:

$$\frac{\Gamma \vdash e_1 : t_1 \qquad \Gamma \vdash e_2 : t_2}{\Gamma \vdash \langle e_1, e_2 \rangle : t_1 \times t_2} \tag{26}$$

$$\frac{\Gamma \vdash e : t_1 \times t_2}{\Gamma \vdash e.1 : t_1} \tag{27}$$

$$\frac{\Gamma \vdash e : t_1 \times t_2}{\Gamma \vdash e.2 : t_2} \tag{28}$$

and the new evaluation rules are:

$$\frac{e_1 \quad \rightarrow \quad e_1'}{\langle e_1, e_2 \rangle \quad \rightarrow \quad \langle e_1', e_2 \rangle} \tag{29}$$

$$\frac{e_2 \quad \rightarrow \quad e_2'}{\langle v_1, e_2 \rangle \quad \rightarrow \quad \langle v_1, e_2' \rangle} \tag{30}$$

$$\frac{e \quad \rightarrow \quad e'}{e.1 \quad \rightarrow \quad e'.1} \tag{31}$$

$$\frac{}{\langle v_1, v_2 \rangle.1 \quad \rightarrow \quad v_1} \tag{32}$$

$$\frac{e \quad \rightarrow \quad e'}{e.2 \quad \rightarrow \quad e'.2} \tag{33}$$

$$\frac{}{\langle v_1, v_2 \rangle.2 \quad \rightarrow \quad v_2} \tag{34}$$

Lemma 1 (Inversion Extension)

1. *If $\Gamma \vdash \langle e_1, e_2 \rangle : t$ then $t = t_1 \times t_2$ for some types t_1 and t_2, where $\Gamma \vdash e_1 : t_1$ and $\Gamma \vdash e_2 : t_2$.*

2. *If $\Gamma \vdash e.1 : t_1$ then $\Gamma \vdash e : t_1 \times t_2$ for some type t_2.*

3. *If $\Gamma \vdash e.2 : t_2$ then $\Gamma \vdash e : t_1 \times t_2$ for some type t_1.*

Lemma 2 (Canonical Forms Extension)

1. *If $\vdash v : t_1 \times t_2$ then v is an pair of the form $\langle v_1, v_2 \rangle$.*

Proposition 3 (Type Preservation)
If $\vdash e : t$ and $e \quad \rightarrow \quad e'$ then $\vdash e' : t$.

Proof: By induction on the proof that $e \quad \rightarrow \quad e'$.

- Case: Rule 29. Then $e = \langle e_1, e_2 \rangle$ and $t = t_1 \times t_2$. By Inversion, $\vdash e_1 : t_1$ and $\vdash e_2 : t_2$. By the inductive hypothesis applied to $e_1 \quad \rightarrow \quad e_1'$, we have $\vdash e_1' : t_1$. Therefore $\vdash \langle e_1', e_2 \rangle : t_1 \times t_2$ as required.

- Case: Rule 30. Then $e = \langle v_1, e_2 \rangle$ and $t = t_1 \times t_2$. By Inversion, $\vdash v_1 : t_1$ and $\vdash e_2 : t_2$. By the inductive hypothesis applied to $e_2 \quad \rightarrow \quad e_2'$, we have $\vdash e_2' : t_2$. Therefore $\vdash \langle v_1, e_2' \rangle : t_1 \times t_2$ as required.

- Case: Rule 31. Then $e = e_1.1$ and $e' = e_1'.1$ where $e_1 \;\longrightarrow\; e_1'$. By inversion, $\vdash e_1 : t \times t_2$ for some type t_2. By the inductive hypothesis, $\vdash e_1' : t \times t_2$. Thus $\vdash e_1'.1 : t$ as required.
- Case: Rule 32. Then $e = \langle v_1, v_2 \rangle.1$ and $e' = v_1$. By inversion, $\vdash \langle v_1, v_2 \rangle : t \times t_2$ for some type t_2. By inversion again, $\vdash v_1 : t$, as required.
- Case: Rule 33 and Rule 34. Exactly analogous to the two previous cases.

∎

Proposition 4 (Progress)
If $\vdash e : t$ *then either* e *is a value or there exists* e' *such that* $e \;\longrightarrow\; e'$.

Proof: By induction on the proof of $\vdash e : t$, and cases on the last rule used.

- Case: Rule 26. Then $e = \langle e_1, e_2 \rangle$ and $t = t_1 \times t_2$ and there are sub-proofs $\vdash e_1 : t_1$ and $\vdash e_2 : t_2$. There are three subcases to consider.
  - Subcase: e_1 is not a value. By the inductive hypothesis, there exists e_1' such that $e_1 \;\longrightarrow\; e_1'$. Thus by Rule 29 we have $\langle e_1, e_2 \rangle \;\longrightarrow\; \langle e_1', e_2 \rangle$.
  - Subcase: e_1 is a value, but e_2 is not. By the inductive hypothesis there exists e_2' such that $e_2 \;\longrightarrow\; e_2'$. By Rule 30 we have $\langle e_1, e_2 \rangle \;\longrightarrow\; \langle e_1, e_2' \rangle$.
  - Subcase: e_1 and e_2 are both values. Then the pair $\langle e_1, e_2 \rangle$ is also a value.
- Case: Rule 27. Then $e = e_1.1$ and and there is a sub-proof $\vdash e_1 : t \times t_2$ for some type t_2. There are two subcases to consider:
  - Subcase: e_1 is not a value. By the inductive hypothesis there exists e_1' such that $e_1 \;\longrightarrow\; e_1'$. Thus $e \;\longrightarrow\; e_1'.1$.
  - Subcase: e_1 is a value. By the Canonical Forms lemma, e_1 must be of the form $\langle v_1, v_2 \rangle$ for some values v_1 and v_2. Thus by Rule 32 we have $e \;\longrightarrow\; v_1$.
- Case: Rule 28. Exactly analogous to the previous case.

∎

3 Natural Semantics (25%)

$$\frac{}{v \Downarrow v} \tag{35}$$

$$\frac{e_1 \Downarrow \overline{\mathtt{m}_1} \qquad e_2 \Downarrow \overline{\mathtt{m}_2}}{e_1 + e_2 \Downarrow \overline{\mathtt{m}_1 + \mathtt{m}_2}} \tag{36}$$

$$\frac{e_1 \Downarrow \overline{\mathtt{m}_1} \qquad e_2 \Downarrow \overline{\mathtt{m}_2}}{e_1 < e_2 \Downarrow \overline{\mathtt{m}_1 < \mathtt{m}_2}} \tag{37}$$

$$\frac{e_1 \Downarrow \overline{\texttt{tt}} \qquad e_2 \Downarrow v_2}{\texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 \Downarrow v_2} \tag{38}$$

$$\frac{e_1 \Downarrow \overline{\texttt{ff}} \qquad e_3 \Downarrow v_3}{\texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 \Downarrow v_3} \tag{39}$$

$$\frac{e_1 \Downarrow v_1 \qquad e_2[x{\mapsto}v_1] \Downarrow v_2}{\texttt{let } x \texttt{ be } e_1 \texttt{ in } e_2 \Downarrow v_2} \tag{40}$$

$$\frac{\begin{array}{c} e_1 \Downarrow (\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1') \qquad e_2 \Downarrow v_2 \\ (e_1'[x{\mapsto}v_2])[f{\mapsto}(\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1')] \Downarrow v \end{array}}{e_1\, e_2 \Downarrow v} \tag{41}$$

1.

Proposition 5
If $e \Downarrow v$ *then* $e \to^* v$.

Proof: By induction on the proof of $e \Downarrow v$, and cases on the last rule used.

- Case: Rule 35. Then $v \to^* v$ because $\to^*$ is reflexive.
- Case: Rule 36. Then $e = e_1 + e_2$ and $v = \overline{\mathtt{m}_1 + \mathtt{m}_2}$, and there are subproofs of $e_1 \Downarrow \overline{\mathtt{m}_1}$ and $e_2 \Downarrow \overline{\mathtt{m}_2}$. Then by the inductive hypothesis we have $e_1 \to^* \overline{\mathtt{m}_1}$ and $e_2 \to^* \overline{\mathtt{m}_2}$. Therefore $e_1 + e_2 \to^* \mathtt{m}_1 + e_2$ and $\overline{\mathtt{m}_1} + e_2 \to^* \overline{\mathtt{m}_1} + \overline{\mathtt{m}_2}$ and $\overline{\mathtt{m}_1} + \overline{\mathtt{m}_2} \to \overline{\mathtt{m}_1 + \mathtt{m}_2}$. By transitivity, $e_1 + e_2 \to^* \overline{\mathtt{m}_1 + \mathtt{m}_2}$.
- Case: Rule 37. Analogous to previous case.
- Case: Rule 38. Then $e = \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3$ and there are subproofs that $e_1 \Downarrow \overline{\texttt{tt}}$ and $e_2 \Downarrow v$. By the inductive hypothesis, $e_1 \to^* \overline{\texttt{tt}}$ and $e_2 \to^* v$. Thus $\texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 \to^* \texttt{if } \overline{\texttt{tt}} \texttt{ then } e_2 \texttt{ else } e_3$ and $\texttt{if } \overline{\texttt{tt}} \texttt{ then } e_2 \texttt{ else } e_3 \to e_2$. By transitivity, $\texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 \to^* v$.
- Case: Rule 39. Analogous to previous case.
- Case: Rule 40. Then $e = \texttt{let } x \texttt{ be } e_1 \texttt{ in } e_2$ and there are subproofs that $e_1 \Downarrow v_1$ and that $e_2[x{\mapsto}v_1] \Downarrow v$. By the inductive hypothesis, $e_1 \to^* v_1$. Thus $(\texttt{let } x \texttt{ be } e_1 \texttt{ in } e_2) \to^* (\texttt{let } x \texttt{ be } v_1 \texttt{ in } e_2)$ and $e_2[x{\mapsto}v_1] \to^* v$. Since $(\texttt{let } x \texttt{ be } v_1 \texttt{ in } e_2) \to e_2[x{\mapsto}v_1]$, we have $(\texttt{let } x \texttt{ be } e_1 \texttt{ in } e_2) \to v$ as required.
- Case: Rule 41. Then $e = e_1\, e_2$ and there are subproofs of $e_1 \Downarrow (\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1')$ and $e_2 \Downarrow v_2$ and $(e_1'[x{\mapsto}v_2])[f{\mapsto}(\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1')] \Downarrow v$. By the inductive hypothesis, $e_1 \to^* (\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1')$ and $e_2 \to^* v_2$ and $(e_1'[x{\mapsto}v_2])[f{\mapsto}(\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1')] \to^* v$. Therefore $e_1\, e_2 \to^* (\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1')\, e_2 \to^* (\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1')\, v_2 \to (e_1'[x{\mapsto}v_2])[f{\mapsto}(\texttt{fun } f(x{:}t_1){:}t_2 \texttt{ is } e_1')] \to^* v$ as required.

■

2.

Proposition 6
If $e \to e'$ *and* $e' \Downarrow v$ *then* $e \Downarrow v$.

Proof: By induction on the proof that $e \to e'$.

- Case: Rule 11. Then $e = e_1 + e_2$ and $e' = e_1' + e_2$ and there is a subproof that $e \to e'$. Also, if e' is an addition expression then $e_1' \Downarrow \overline{\mathtt{m}_1}$ and $e_2 \Downarrow \overline{\mathtt{m}_2}$ and $v = \overline{\mathtt{m}_1 + \mathtt{m}_2}$. (This is an inversion property for the $\Downarrow$ relation.) By the inductive hypothesis, $e_1 \Downarrow \overline{\mathtt{m}_1}$. Therefore $e \Downarrow \overline{\mathtt{m}_1 + \mathtt{m}_2}$ as required.
- Case: Rule 12: Analogous to previous case.
- Case: Rule 13: Then $e = \overline{\mathtt{m}_1} + \overline{\mathtt{m}_2}$ and $e' = v = \overline{\mathtt{m}_1 + \mathtt{m}_2}$. Obviously $e \Downarrow v$.
- Case: Rules 14–16: Analogous to cases for addition.
- Case: Rule 17: Analogous to the case for rule 11.
- Case: Rule 18 and 19. Obvious by definition of $\Downarrow$ that if $e_2 \Downarrow v$ then $\mathtt{if}\ \overline{\mathtt{tt}}\ \mathtt{then}\ e_2\ \mathtt{else}\ e_3 \Downarrow v$, and similarly if $e_3 \Downarrow v$ then $\mathtt{if}\ \overline{\mathtt{ff}}\ \mathtt{then}\ e_2\ \mathtt{else}\ e_3 \Downarrow v$.
- Case: Rule 20. Then $e = (\mathtt{let}\ x\ \mathtt{be}\ e_1\ \mathtt{in}\ e_2)$ and $e' = (\mathtt{let}\ x\ \mathtt{be}\ e_1'\ \mathtt{in}\ e_2)$ and $e_1 \to e_1'$ and $e_1' \Downarrow v_1$ and $e_2[x{\mapsto}v_1] \Downarrow v$. By the inductive hypothesis, $e_1 \Downarrow v_1$. Thus $\mathtt{let}\ x\ \mathtt{be}\ e_1\ \mathtt{in}\ e_2 \Downarrow v$.
- Case: Rule 21. Obvious by definition of $\Downarrow$.
- Case: Rules 22 and 23. Analogous to cases for addition.
- Case: Rule 24. Obvious by definition of $\Downarrow$.

■

3.

Proposition 7
If $e \to^* v$ *then* $e \Downarrow v$.

Proof: By induction on n, the number of $\to$ steps needed to reach v from e.

- Case: $n = 0$. Then $e = v$. Since e is a value, we have $e \Downarrow v$ by Rule 35.
- Case: $n > 0$. Then $e \to e' \to^* v$. By the inductive hypothesis, $e' \Downarrow v$. By the previous part, then, $e \Downarrow v$.

■