CS 6371	**Advanced Programming Languages**	SAMPLE
SPRING 2014		MIDTERM EXAM

Write your name at the top of this exam paper and turn it in as the front page of your submission. You may take a single, two-sided sheet of notes with you into the exam. All other books or notes must remain closed throughout the exam. You will have the duration of the class period to complete the exam; all papers must be turned in by 2:15pm. Good luck!

(1) A number can be encoded as a list of digits. For example, the list `[1;0;2;4]` encodes the number 1024 in base-10, and the list `[15;15]` encodes 0xFF=255 in base-16. In this problem you will implement OCaml functions that convert between integers and digit lists. Do not use any `List` library functions in your solutions except that you may use `List.fold_left` if you wish. In case you need them, the OCaml integer-division and integer-modulo operators are `(x/y)` and `(x mod y)`, respectively.

- **(a)** **(7 pts)** Implement a **tail-recursive** OCaml function `(digitize b n)` that converts an integer `n` into a base-`b` digit list. You may assume that $\texttt{b} \geq 2$ and $\texttt{n} \geq 0$.
- **(b)** **(6 pts)** Implement a **tail-recursive** OCaml function `(undigitize b dl)` that converts a base-`b` digit list `dl` to an integer. You may assume that $\texttt{b} \geq 2$ and that no elements of `dl` are less than 0 or greater than $\texttt{b} - 1$.

(2) Consider the following extension to IMP, which defines the syntax and large-step semantics of a new command called `run`:

$$c ::= \ldots \mid \texttt{run}(a, c)$$

$$\frac{\langle a, \sigma \rangle \Downarrow n \qquad n \geq 1 \qquad \langle c; \texttt{run}(n - 1, c), \sigma \rangle \Downarrow \sigma'}{\langle \texttt{run}(a, c), \sigma \rangle \Downarrow \sigma'} \tag{60}$$

$$\frac{\langle a, \sigma \rangle \Downarrow n \qquad n \leq 0}{\langle \texttt{run}(a, c), \sigma \rangle \Downarrow \sigma} \tag{61}$$

- **(a)** **(4 pts)** Explain in words what the `run` command does.
- **(b)** **(13 pts)** Prove that if $\langle \texttt{run}(1, c), \sigma \rangle \Downarrow \sigma'$ then $\langle c, \sigma \rangle \Downarrow \sigma'$.

(3) **(5 pts)** Write small-step operational semantic rules for `run` that are equivalent to the large-step rules given above.

(4) Consider the following recursive definition of function f:

$$f(x) = \big(x{=}100 \rightarrow 0 \mid x{>}100 \rightarrow f(x-1)+1 \mid x{<}100 \rightarrow f(x+1)+1\big)$$

- **(a)** **(1 pt)** Define a non-recursive functional F whose least fixed point is f.
- **(b)** **(4 pts)** Give a closed-form function definition h such that $h = f$.
- **(c)** **(15 pts)** Prove by fixed point induction that $h = \mathit{fix}(F)$.

Solutions

(1) **(a)**

```
let digitize b n =
    let rec f dl n = if n<=0 then dl else f ((n mod b)::dl) (n/b)
    in f [] n;;
```

(b)

```
let undigitize b = List.fold_left (fun a d -> a*b+d) 0;;
```

(2) **(a)** The $\texttt{run}(a, c)$ command first evaluates a. If the resulting integer n is zero or less, nothing happens. Otherwise command c is executed n times consecutively.

(b) Any derivation of $\langle \texttt{run}(1, c), \sigma\rangle \Downarrow \sigma'$ must have the following form:

$$\dfrac{\langle 1,\sigma\rangle \Downarrow 1 \quad 1 \geq 1 \qquad \dfrac{\dfrac{\mathcal{D}}{\langle c,\sigma\rangle \Downarrow \sigma_2} \qquad \dfrac{\dfrac{\langle 1,\sigma_2\rangle \Downarrow 1 \quad \langle 1,\sigma_2\rangle \Downarrow 1}{\langle 1 - 1,\sigma_2\rangle \Downarrow 0}(16) \qquad 0 \leq 0}{\langle \texttt{run}(1 - 1, c),\sigma_2\rangle \Downarrow \sigma'}(61)}{\langle c; \texttt{run}(1 - 1, c),\sigma\rangle \Downarrow \sigma'}(2)}{\langle \texttt{run}(1, c),\sigma\rangle \Downarrow \sigma'}(60)$$

Observe that rule (61) in the above derivation requires that $\sigma_2 = \sigma'$. Therefore the above derivation includes a sub-derivation $\mathcal{D}$ of judgment $\langle c, \sigma\rangle \Downarrow \sigma'$.

(3) Here is one possible answer:

$$\dfrac{\langle a,\sigma\rangle \rightarrow_1 \langle a',\sigma'\rangle}{\langle \texttt{run}(a, c),\sigma\rangle \rightarrow_1 \langle \texttt{run}(a', c),\sigma'\rangle} \qquad \dfrac{n \geq 1}{\langle \texttt{run}(n, c),\sigma\rangle \rightarrow_1 \langle c; \texttt{run}(n - 1, c),\sigma\rangle}$$

$$\dfrac{n \leq 0}{\langle \texttt{run}(n, c),\sigma\rangle \rightarrow_1 \langle \texttt{skip},\sigma\rangle}$$

(4) **(a)** $F(g) = \lambda x.\big(x{=}100 \rightarrow 0 \mid x{>}100 \rightarrow g(x-1)+1 \mid x{<}100 \rightarrow g(x+1)+1\big)$

(b) $h(x) = |x - 100|$

(c) *Proof.* Define proposition $P(g) = \forall x \in g^{\leftarrow} \,.\, g(x) = |x - 100|$. We wish to prove that $P(\mathit{fix}(F))$ holds.

Base Case: $P(\perp)$ holds vacuously.

Inductive Case: As the inductive hypothesis, assume that $P(g)$ holds. We must prove that $P(F(g))$ holds. Let $x \in F(g)^{\leftarrow}$ be given.

Case 1: Suppose $x = 100$. Then by definition of F, $F(g)(x) = 0 = |x - 100|$.

Case 2: Suppose $x > 100$. Then by definition of F, $F(g)(x) = g(x-1)+1$. By inductive hypothesis, $g(x-1) = |x-1-100|$. Since $x > 100$, $|x-1-100|+1 = x-1-100+1 = x-100 = |x-100|$.

Case 3: Suppose $x < 100$. Then by definition of F, $F(g)(x) = g(x+1)+1$. Since $x < 100$, $|x+1-100|+1 = -(x+1-100)+1 = -(x-100) = |x-100|$. □

Reference

For your reference, here is the syntax, large-step operational semantics, small-step operational semantics, and denotational semantics of IMP that were defined in class. These definitions will be provided to you with your midterm exam.

Syntax of IMP

commands	$c ::= \texttt{skip} \mid c_1; c_2 \mid v \texttt{:=} a \mid \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2 \mid \texttt{while } b \texttt{ do } c$
boolean expressions	$b ::= \texttt{true} \mid \texttt{false} \mid a_1 \texttt{<=} a_2 \mid b_1 \,\texttt{\&\&}\, b_2 \mid b_1 \,\texttt{||}\, b_2 \mid \texttt{!}b$
arithmetic expressions	$a ::= n \mid v \mid a_1 \texttt{+} a_2 \mid a_1 \texttt{-} a_2 \mid a_1 \texttt{*} a_2$
variable names	v
integer constants	n

Large-step Semantics of IMP

Commands

$$\langle \texttt{skip}, \sigma \rangle \Downarrow \sigma \tag{1}$$

$$\frac{\langle c_1, \sigma \rangle \Downarrow \sigma_2 \qquad \langle c_2, \sigma_2 \rangle \Downarrow \sigma'}{\langle c_1; c_2, \sigma \rangle \Downarrow \sigma'} \tag{2}$$

$$\frac{\langle a, \sigma \rangle \Downarrow n}{\langle v \texttt{:=} a, \sigma \rangle \Downarrow \sigma[v \mapsto n]} \tag{3}$$

$$\frac{\langle b, \sigma \rangle \Downarrow T \qquad \langle c_1, \sigma \rangle \Downarrow \sigma'}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma \rangle \Downarrow \sigma'} \tag{4}$$

$$\frac{\langle b, \sigma \rangle \Downarrow F \qquad \langle c_2, \sigma \rangle \Downarrow \sigma'}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma \rangle \Downarrow \sigma'} \tag{5}$$

$$\frac{\langle \texttt{if } b \texttt{ then } (c; \texttt{while } b \texttt{ do } c) \texttt{ else skip}, \sigma \rangle \Downarrow \sigma'}{\langle \texttt{while } b \texttt{ do } c, \sigma \rangle \Downarrow \sigma'} \tag{6}$$

Boolean Expressions

$$\langle \texttt{true}, \sigma\rangle \Downarrow T \tag{7}$$

$$\langle \texttt{false}, \sigma\rangle \Downarrow F \tag{8}$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow n_1 \qquad \langle a_2, \sigma\rangle \Downarrow n_2}{\langle a_1 \texttt{<=} a_2, \sigma\rangle \Downarrow n_1 \leq n_2} \tag{9}$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow p \qquad \langle b_2, \sigma\rangle \Downarrow q}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma\rangle \Downarrow p \wedge q} \tag{10}$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow p \qquad \langle b_2, \sigma\rangle \Downarrow q}{\langle b_1 \,\texttt{||}\, b_2, \sigma\rangle \Downarrow p \vee q} \tag{11}$$

$$\frac{\langle b, \sigma\rangle \Downarrow p}{\langle \texttt{!}b, \sigma\rangle \Downarrow \neg p} \tag{12}$$

Arithmetic Expressions

$$\langle n, \sigma\rangle \Downarrow n \tag{13}$$

$$\langle v, \sigma\rangle \Downarrow \sigma(v) \tag{14}$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow n_1 \qquad \langle a_2, \sigma\rangle \Downarrow n_2}{\langle a_1 \,\texttt{+}\, a_2, \sigma\rangle \Downarrow n_1 + n_2} \tag{15}$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow n_1 \qquad \langle a_2, \sigma\rangle \Downarrow n_2}{\langle a_1 \,\texttt{-}\, a_2, \sigma\rangle \Downarrow n_1 - n_2} \tag{16}$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow n_1 \qquad \langle a_2, \sigma\rangle \Downarrow n_2}{\langle a_1 \,\texttt{*}\, a_2, \sigma\rangle \Downarrow n_1 n_2} \tag{17}$$

Small-step Semantics of IMP

Commands

$$\frac{\langle c_1, \sigma\rangle \rightarrow_1 \langle c_1', \sigma'\rangle}{\langle c_1; c_2, \sigma\rangle \rightarrow_1 \langle c_1'; c_2, \sigma'\rangle} \tag{18}$$

$$\langle \texttt{skip}; c_2, \sigma\rangle \rightarrow_1 \langle c_2, \sigma\rangle \tag{19}$$

$$\frac{\langle a, \sigma\rangle \rightarrow_1 \langle a', \sigma'\rangle}{\langle v \,\texttt{:=}\, a, \sigma\rangle \rightarrow_1 \langle v \,\texttt{:=}\, a', \sigma'\rangle} \tag{20}$$

$$\langle v \,\texttt{:=}\, n, \sigma\rangle \rightarrow_1 \langle \texttt{skip}, \sigma[v \mapsto n]\rangle \tag{21}$$

$$\frac{\langle b, \sigma\rangle \rightarrow_1 \langle b', \sigma'\rangle}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \rightarrow_1 \langle \texttt{if } b' \texttt{ then } c_1 \texttt{ else } c_2, \sigma'\rangle} \tag{22}$$

$$\langle \texttt{if true then } c_1 \texttt{ else } c_2, \sigma\rangle \rightarrow_1 \langle c_1, \sigma\rangle \tag{23}$$

$$\langle \texttt{if false then } c_1 \texttt{ else } c_2, \sigma\rangle \rightarrow_1 \langle c_2, \sigma\rangle \tag{24}$$

$$\langle \texttt{while } b \texttt{ do } c, \sigma\rangle \rightarrow_1 \langle \texttt{if } b \texttt{ then } (c; \texttt{while } b \texttt{ do } c) \texttt{ else skip}, \sigma\rangle \tag{25}$$

Boolean Expressions

$$\frac{\langle a_1, \sigma\rangle \rightarrow_1 \langle a_1', \sigma'\rangle}{\langle a_1 \texttt{<=} a_2, \sigma\rangle \rightarrow_1 \langle a_1' \texttt{<=} a_2, \sigma'\rangle} \tag{26}$$

$$\frac{\langle a_2, \sigma\rangle \rightarrow_1 \langle a_2', \sigma'\rangle}{\langle n_1 \texttt{<=} a_2, \sigma\rangle \rightarrow_1 \langle n_1 \texttt{<=} a_2', \sigma'\rangle} \tag{27}$$

$$\frac{n_1 \leq n_2}{\langle n_1 \texttt{<=} n_2, \sigma\rangle \rightarrow_1 \langle \texttt{true}, \sigma\rangle} \tag{28}$$

$$\frac{n_1 > n_2}{\langle n_1 \texttt{<=} n_2, \sigma\rangle \rightarrow_1 \langle \texttt{false}, \sigma\rangle} \tag{29}$$

$$\frac{\langle b_1, \sigma\rangle \rightarrow_1 \langle b_1', \sigma'\rangle \quad op \in \{\, \texttt{\&\&}\,,\ \texttt{||}\,\}}{\langle b_1\ op\ b_2, \sigma\rangle \rightarrow_1 \langle b_1'\ op\ b_2, \sigma'\rangle} \tag{30}$$

$$\langle \texttt{true}\, \texttt{\&\&}\, b_2, \sigma\rangle \rightarrow_1 \langle b_2, \sigma\rangle \tag{31}$$

$$\langle \texttt{false}\, \texttt{\&\&}\, b_2, \sigma\rangle \rightarrow_1 \langle \texttt{false}, \sigma\rangle \tag{32}$$

$$\langle \texttt{true}\, \texttt{||}\, b_2, \sigma\rangle \rightarrow_1 \langle \texttt{true}, \sigma\rangle \tag{33}$$

$$\langle \texttt{false}\, \texttt{||}\, b_2, \sigma\rangle \rightarrow_1 \langle b_2, \sigma\rangle \tag{34}$$

$$\frac{\langle b, \sigma\rangle \rightarrow_1 \langle b', \sigma'\rangle}{\langle !b, \sigma\rangle \rightarrow_1 \langle !b', \sigma'\rangle} \tag{35}$$

$$\langle !\texttt{true}, \sigma\rangle \rightarrow_1 \langle \texttt{false}, \sigma\rangle \tag{36}$$

$$\langle !\texttt{false}, \sigma\rangle \rightarrow_1 \langle \texttt{true}, \sigma\rangle \tag{37}$$

Arithmetic Expressions

$$\langle v, \sigma\rangle \rightarrow_1 \langle \sigma(v), \sigma\rangle \tag{38}$$

$$\frac{\langle a_1, \sigma\rangle \rightarrow_1 \langle a_1', \sigma'\rangle \qquad op \in \{\, \texttt{+}\,,\ \texttt{-}\,,\ \texttt{*}\,\}}{\langle a_1\ op\ a_2, \sigma\rangle \rightarrow_1 \langle a_1'\ op\ a_2, \sigma'\rangle} \tag{39}$$

$$\frac{\langle a_2, \sigma\rangle \rightarrow_1 \langle a_2', \sigma'\rangle \qquad op \in \{\, \texttt{+}\,,\ \texttt{-}\,,\ \texttt{*}\,\}}{\langle n_1\ op\ a_2, \sigma\rangle \rightarrow_1 \langle n_1\ op\ a_2', \sigma'\rangle} \tag{40}$$

$$\langle n_1 \texttt{+} n_2, \sigma\rangle \rightarrow_1 \langle n_1 + n_2, \sigma\rangle \tag{41}$$

$$\langle n_1 \texttt{-} n_2, \sigma\rangle \rightarrow_1 \langle n_1 - n_2, \sigma\rangle \tag{42}$$

$$\langle n_1 \texttt{*} n_2, \sigma\rangle \rightarrow_1 \langle n_1 n_2, \sigma\rangle \tag{43}$$

Denotational Semantics

$$\begin{aligned}
&\Sigma = v \rightharpoonup \mathbb{Z}\\
&\mathcal{A} : a \rightarrow (\Sigma \rightharpoonup \mathbb{Z})\\
&\mathcal{B} : b \rightarrow (\Sigma \rightharpoonup \{T, F\})\\
&\mathcal{C} : c \rightarrow (\Sigma \rightharpoonup \Sigma)
\end{aligned}$$

Arithmetic Expressions

$$\mathcal{A}[\![n]\!] = \{(\sigma, n) \mid \sigma \in \Sigma\} \tag{44}$$

$$\mathcal{A}[\![x]\!] = \{(\sigma, \sigma(x)) \mid \sigma \in \Sigma,\ x \in \sigma^{\leftarrow}\} \tag{45}$$

$$\mathcal{A}[\![a_1 \texttt{+} a_2]\!] = \{(\sigma, n_1 + n_2) \mid n_1 = \mathcal{A}[\![a_1]\!]\sigma,\ n_2 = \mathcal{A}[\![a_2]\!]\sigma\} \tag{46}$$

$$\mathcal{A}[\![a_1 \texttt{-} a_2]\!] = \{(\sigma, n_1 - n_2) \mid n_1 = \mathcal{A}[\![a_1]\!]\sigma,\ n_2 = \mathcal{A}[\![a_2]\!]\sigma\} \tag{47}$$

$$\mathcal{A}[\![a_1 \texttt{*} a_2]\!] = \{(\sigma, n_1 n_2) \mid n_1 = \mathcal{A}[\![a_1]\!]\sigma,\ n_2 = \mathcal{A}[\![a_2]\!]\sigma\} \tag{48}$$

Boolean Expressions

$$\mathcal{B}[\![\texttt{true}]\!] = \{(\sigma, T) \mid \sigma \in \Sigma\} \tag{49}$$

$$\mathcal{B}[\![\texttt{false}]\!] = \{(\sigma, F) \mid \sigma \in \Sigma\} \tag{50}$$

$$\begin{aligned}\mathcal{B}[\![a_1 \texttt{<=} a_2]\!] = {} & \{(\sigma, T) \mid \mathcal{A}[\![a_1]\!]\sigma \leq \mathcal{A}[\![a_2]\!]\sigma\} \cup \\ & \{(\sigma, F) \mid \mathcal{A}[\![a_1]\!]\sigma > \mathcal{A}[\![a_2]\!]\sigma\}\end{aligned} \tag{51}$$

$$\begin{aligned}\mathcal{B}[\![b_1 \,\texttt{\&\&}\, b_2]\!] = {} & \{(\sigma, T) \mid \mathcal{B}[\![b_1]\!]\sigma = T,\ \mathcal{B}[\![b_2]\!]\sigma = T\} \cup \\ & \{(\sigma, F) \mid \mathcal{B}[\![b_1]\!]\sigma = F\} \cup \\ & \{(\sigma, F) \mid \mathcal{B}[\![b_2]\!]\sigma = F\}\end{aligned} \tag{52}$$

$$\begin{aligned}\mathcal{B}[\![b_1 \,\texttt{||}\, b_2]\!] = {} & \{(\sigma, T) \mid \mathcal{B}[\![b_1]\!]\sigma = T\} \cup \\ & \{(\sigma, T) \mid \mathcal{B}[\![b_2]\!]\sigma = T\} \cup \\ & \{(\sigma, F) \mid \mathcal{B}[\![b_1]\!]\sigma = F,\ \mathcal{B}[\![b_2]\!]\sigma = F\}\end{aligned} \tag{53}$$

$$\begin{aligned}\mathcal{B}[\![\texttt{!}b]\!] = {} & \{(\sigma, T) \mid \mathcal{B}[\![b]\!]\sigma = F\} \cup \\ & \{(\sigma, F) \mid \mathcal{B}[\![b]\!]\sigma = T\}\end{aligned} \tag{54}$$

Commands

$$\mathcal{C}[\![\texttt{skip}]\!] = \{(\sigma, \sigma) \mid \sigma \in \Sigma\} \tag{55}$$

$$\mathcal{C}[\![v := a]\!] = \{(\sigma, \sigma[v \mapsto n]) \mid n = \mathcal{A}[\![a]\!]\sigma\} \tag{56}$$

$$\mathcal{C}[\![c_1; c_2]\!] = \{(\sigma, \mathcal{C}[\![c_2]\!](\mathcal{C}[\![c_1]\!]\sigma)) \mid \sigma \in \Sigma\} = \mathcal{C}[\![c_2]\!] \circ \mathcal{C}[\![c_1]\!] \tag{57}$$

$$\begin{aligned}\mathcal{C}[\![\texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2]\!] = {} & \{(\sigma, \mathcal{C}[\![c_1]\!]\sigma) \mid \mathcal{B}[\![b]\!]\sigma = T\} \cup \\ & \{(\sigma, \mathcal{C}[\![c_2]\!]\sigma) \mid \mathcal{B}[\![b]\!]\sigma = F\}\end{aligned} \tag{58}$$

$$\mathcal{C}[\![\texttt{while}\ b\ \texttt{do}\ c]\!] = \bigcup_{i \geq 0} \Gamma^i(\bot_{\Sigma \rightharpoonup \Sigma}) = \mathit{fix}(\Gamma) \tag{59}$$

$$\begin{aligned}\text{where} \quad \Gamma(g) = {} & \{(\sigma, (g \circ \mathcal{C}[\![c]\!])(\sigma)) \mid \mathcal{B}[\![b]\!]\sigma = T\} \cup \\ & \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!]\sigma = F\}\end{aligned}$$