

Write all of your answers and scratch work in your exam booklet(s), not on this exam paper; only material written in your booklet(s) will be graded. Remember to write your name on the front of your exam booklet. 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 5:15pm. When turning in your answers, place your exam paper inside your exam booklet and turn them both in together. Good luck!

- (1) (8 pts) Implement a **tail-recursive** OCaml function named `fsum` that takes as input a value  $x$  and a list  $l$  of functions and returns an integer  $i$  such that  $i = \sum_{f \in l} f(x)$ . You may assume that each function in list  $l$  has type `'a->int`, where  $x$  has type `'a`. Your code should therefore begin with:

```
let fsum (x:'a) (l:( 'a->int) list) : int = ...
```

Do not use any of the OCaml List library functions in your implementation, except that you may use `List.fold_left` if you wish. Remember that your code must be tail-recursive to receive full credit!

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

$$c ::= \dots \mid \mathbf{run}(a, c)$$

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

$$\frac{\langle a, \sigma \rangle \Downarrow n \quad n \leq 0}{\langle \mathbf{run}(a, c), \sigma \rangle \Downarrow \sigma} \quad (61)$$

- (a) (4 pts) Explain in words what the `run` command does.
- (b) (13 pts) Prove that if  $\langle \mathbf{run}(1, c), \sigma \rangle \Downarrow \sigma'$  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) (20 pts) Prove by fixed-point induction that  $\mathcal{C}[\mathbf{while} \ 0 \leq x \ \mathbf{do} \ x := x + 1] = \{(\sigma, \sigma) \mid \sigma \in \Sigma, \sigma(x) < 0\}$ .

**Hint:** Instead of proving the theorem for  $\perp$  in your base case, prove the theorem for  $\Gamma(\perp)$  as your base case.

## Solutions

(1) `let fsum (x:'a) (l:( 'a->int) list) : int =  
List.fold_left (fun s f -> s+(f x)) 0 l;;`

(2) (a) The `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  $\langle \text{run}(1, c), \sigma \rangle \Downarrow \sigma'$  must have the following form:

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

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)

$$\frac{\frac{\langle a, \sigma \rangle \rightarrow_1 \langle a', \sigma' \rangle}{\langle \text{run}(a, c), \sigma \rangle \rightarrow_1 \langle \text{run}(a', c), \sigma' \rangle}}{n \geq 1} \quad \frac{\langle \text{run}(n, c), \sigma \rangle \rightarrow_1 \langle c; \text{run}(n-1, c), \sigma \rangle}{n \leq 0}}{\langle \text{run}(n, c), \sigma \rangle \rightarrow_1 \langle \text{skip}, \sigma \rangle}$$

(4)  $\mathcal{C}[\text{while } 0 \leq x \text{ do } x := x + 1] = \text{fix}(\Gamma)$  where  $\Gamma$  is defined by

$$\begin{aligned} \Gamma(g) &= \{(\sigma, (g \circ \mathcal{C}[x := x + 1])(\sigma)) \mid \mathcal{B}[0 \leq x](\sigma) = T\} \cup \\ &\quad \{(\sigma, \sigma) \mid \mathcal{B}[0 \leq x](\sigma) = F\} \\ &= \{(\sigma, g(\sigma[x \mapsto \sigma(x) + 1])) \mid 0 \leq \sigma(x)\} \cup \\ &\quad \{(\sigma, \sigma) \mid 0 > \sigma(x)\} \end{aligned}$$

**Base Case:** When  $g = \perp$ , the first half of the definition of  $\Gamma(g)$  is the empty set (because  $g$  is undefined for all inputs). Thus  $\Gamma(\perp) = \{(\sigma, \sigma) \mid \sigma(x) < 0\}$ .

**Inductive Case:** Assume as the inductive hypothesis that  $g = \{(\sigma, \sigma) \mid \sigma(x) < 0\}$ .

We must prove that  $\Gamma(g) = \{(\sigma, \sigma) \mid \sigma(x) < 0\}$ .

Define  $\sigma_0 = \sigma[x \mapsto \sigma(x) + 1]$ . By inductive hypothesis,  $g(\sigma_0)$  is only defined when  $\sigma_0(x) < 0 \iff \sigma(x) + 1 < 0 \iff \sigma(x) < -1$ . The first half of the definition of  $\Gamma$  is therefore only defined for stores  $\sigma$  such that  $\sigma(x) \geq 0$  and  $\sigma(x) < -1$ . Since these two requirements are contradictory,  $\Gamma(g)$  consists only of elements in the second half of the definition, and therefore  $\Gamma(g) = \{(\sigma, \sigma) \mid \sigma(x) < 0\}$ .

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

## Large-step Semantics of IMP

### Commands

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

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

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

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

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

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

## Boolean Expressions

$$\langle \text{true}, \sigma \rangle \Downarrow T \quad (7)$$

$$\langle \text{false}, \sigma \rangle \Downarrow F \quad (8)$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow n_1 \quad \langle a_2, \sigma \rangle \Downarrow n_2}{\langle a_1 \leq a_2, \sigma \rangle \Downarrow n_1 \leq n_2} \quad (9)$$

$$\frac{\langle b_1, \sigma \rangle \Downarrow p \quad \langle b_2, \sigma \rangle \Downarrow q}{\langle b_1 \ \&\& \ b_2, \sigma \rangle \Downarrow p \wedge q} \quad (10)$$

$$\frac{\langle b_1, \sigma \rangle \Downarrow p \quad \langle b_2, \sigma \rangle \Downarrow q}{\langle b_1 \ || \ b_2, \sigma \rangle \Downarrow p \vee q} \quad (11)$$

$$\frac{\langle b, \sigma \rangle \Downarrow p}{\langle !b, \sigma \rangle \Downarrow \neg p} \quad (12)$$

## Arithmetic Expressions

$$\langle n, \sigma \rangle \Downarrow n \quad (13)$$

$$\langle v, \sigma \rangle \Downarrow \sigma(v) \quad (14)$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow n_1 \quad \langle a_2, \sigma \rangle \Downarrow n_2}{\langle a_1 + a_2, \sigma \rangle \Downarrow n_1 + n_2} \quad (15)$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow n_1 \quad \langle a_2, \sigma \rangle \Downarrow n_2}{\langle a_1 - a_2, \sigma \rangle \Downarrow n_1 - n_2} \quad (16)$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow n_1 \quad \langle a_2, \sigma \rangle \Downarrow n_2}{\langle a_1 * a_2, \sigma \rangle \Downarrow n_1 n_2} \quad (17)$$

## Small-step Semantics of IMP

### Commands

$$\frac{\langle c_1, \sigma \rangle \rightarrow_1 \langle c'_1, \sigma' \rangle}{\langle c_1; c_2, \sigma \rangle \rightarrow_1 \langle c'_1; c_2, \sigma' \rangle} \quad (18)$$

$$\langle \text{skip}; c_2, \sigma \rangle \rightarrow_1 \langle c_2, \sigma \rangle \quad (19)$$

$$\frac{\langle a, \sigma \rangle \rightarrow_1 \langle a', \sigma' \rangle}{\langle v := a, \sigma \rangle \rightarrow_1 \langle v := a', \sigma' \rangle} \quad (20)$$

$$\langle v := n, \sigma \rangle \rightarrow_1 \langle \text{skip}, \sigma[v \mapsto n] \rangle \quad (21)$$

$$\frac{\langle b, \sigma \rangle \rightarrow_1 \langle b', \sigma' \rangle}{\langle \text{if } b \text{ then } c_1 \text{ else } c_2, \sigma \rangle \rightarrow_1 \langle \text{if } b' \text{ then } c_1 \text{ else } c_2, \sigma' \rangle} \quad (22)$$

$$\langle \text{if true then } c_1 \text{ else } c_2, \sigma \rangle \rightarrow_1 \langle c_1, \sigma \rangle \quad (23)$$

$$\langle \text{if false then } c_1 \text{ else } c_2, \sigma \rangle \rightarrow_1 \langle c_2, \sigma \rangle \quad (24)$$

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

## Boolean Expressions

$$\frac{\langle a_1, \sigma \rangle \rightarrow_1 \langle a'_1, \sigma' \rangle}{\langle a_1 \leq a_2, \sigma \rangle \rightarrow_1 \langle a'_1 \leq a_2, \sigma' \rangle} \quad (26)$$

$$\frac{\langle a_2, \sigma \rangle \rightarrow_1 \langle a'_2, \sigma' \rangle}{\langle n_1 \leq a_2, \sigma \rangle \rightarrow_1 \langle n_1 \leq a'_2, \sigma' \rangle} \quad (27)$$

$$\frac{n_1 \leq n_2}{\langle n_1 \leq n_2, \sigma \rangle \rightarrow_1 \langle \text{true}, \sigma \rangle} \quad (28)$$

$$\frac{n_1 > n_2}{\langle n_1 \leq n_2, \sigma \rangle \rightarrow_1 \langle \text{false}, \sigma \rangle} \quad (29)$$

$$\frac{\langle b_1, \sigma \rangle \rightarrow_1 \langle b'_1, \sigma' \rangle \quad op \in \{ \&\&, || \}}{\langle b_1 op b_2, \sigma \rangle \rightarrow_1 \langle b'_1 op b_2, \sigma' \rangle} \quad (30)$$

$$\langle \text{true} \&\& b_2, \sigma \rangle \rightarrow_1 \langle b_2, \sigma \rangle \quad (31)$$

$$\langle \text{false} \&\& b_2, \sigma \rangle \rightarrow_1 \langle \text{false}, \sigma \rangle \quad (32)$$

$$\langle \text{true} || b_2, \sigma \rangle \rightarrow_1 \langle \text{true}, \sigma \rangle \quad (33)$$

$$\langle \text{false} || b_2, \sigma \rangle \rightarrow_1 \langle b_2, \sigma \rangle \quad (34)$$

$$\frac{\langle b, \sigma \rangle \rightarrow_1 \langle b', \sigma' \rangle}{\langle !b, \sigma \rangle \rightarrow_1 \langle !b', \sigma' \rangle} \quad (35)$$

$$\langle !\text{true}, \sigma \rangle \rightarrow_1 \langle \text{false}, \sigma \rangle \quad (36)$$

$$\langle !\text{false}, \sigma \rangle \rightarrow_1 \langle \text{true}, \sigma \rangle \quad (37)$$

## Arithmetic Expressions

$$\langle v, \sigma \rangle \rightarrow_1 \langle \sigma(v), \sigma \rangle \quad (38)$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_1 \langle a'_1, \sigma' \rangle \quad op \in \{ +, -, * \}}{\langle a_1 op a_2, \sigma \rangle \rightarrow_1 \langle a'_1 op a_2, \sigma' \rangle} \quad (39)$$

$$\frac{\langle a_2, \sigma \rangle \rightarrow_1 \langle a'_2, \sigma' \rangle \quad op \in \{ +, -, * \}}{\langle n_1 op a_2, \sigma \rangle \rightarrow_1 \langle n_1 op a'_2, \sigma' \rangle} \quad (40)$$

$$\langle n_1 + n_2, \sigma \rangle \rightarrow_1 \langle n_1 + n_2, \sigma \rangle \quad (41)$$

$$\langle n_1 - n_2, \sigma \rangle \rightarrow_1 \langle n_1 - n_2, \sigma \rangle \quad (42)$$

$$\langle n_1 * n_2, \sigma \rangle \rightarrow_1 \langle n_1 n_2, \sigma \rangle \quad (43)$$

## Denotational Semantics

$$\Sigma = v \rightarrow \mathbb{Z}$$

$$\mathcal{A} : a \rightarrow (\Sigma \rightarrow \mathbb{Z})$$

$$\mathcal{B} : b \rightarrow (\Sigma \rightarrow \{T, F\})$$

$$\mathcal{C} : c \rightarrow (\Sigma \rightarrow \Sigma)$$

## Arithmetic Expressions

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

$$\mathcal{A}[x] = \{(\sigma, \sigma(n)) \mid \sigma \in \Sigma\} \quad (45)$$

$$\mathcal{A}[a_1 + a_2] = \{(\sigma, n_1 + n_2) \mid \sigma \in \Sigma, n_1 = \mathcal{A}[a_1](\sigma), n_2 = \mathcal{A}[a_2](\sigma)\} \quad (46)$$

$$\mathcal{A}[a_1 - a_2] = \{(\sigma, n_1 - n_2) \mid \sigma \in \Sigma, n_1 = \mathcal{A}[a_1](\sigma), n_2 = \mathcal{A}[a_2](\sigma)\} \quad (47)$$

$$\mathcal{A}[a_1 * a_2] = \{(\sigma, n_1 n_2) \mid \sigma \in \Sigma, n_1 = \mathcal{A}[a_1](\sigma), n_2 = \mathcal{A}[a_2](\sigma)\} \quad (48)$$

## Boolean Expressions

$$\mathcal{B}[\text{true}] = \{(\sigma, T) \mid \sigma \in \Sigma\} \quad (49)$$

$$\mathcal{B}[\text{false}] = \{(\sigma, F) \mid \sigma \in \Sigma\} \quad (50)$$

$$\mathcal{B}[a_1 \leq a_2] = \{(\sigma, T) \mid \sigma \in \Sigma, \mathcal{A}[a_1](\sigma) \leq \mathcal{A}[a_2](\sigma)\} \cup \{(\sigma, F) \mid \sigma \in \Sigma, \mathcal{A}[a_1](\sigma) > \mathcal{A}[a_2](\sigma)\} \quad (51)$$

$$\mathcal{B}[b_1 \ \&\& \ b_2] = \{(\sigma, T) \mid \sigma \in \Sigma, \mathcal{B}[b_1](\sigma) = T, \mathcal{B}[b_2](\sigma) = T\} \cup \{(\sigma, F) \mid \sigma \in \Sigma, \mathcal{B}[b_1](\sigma) = F\} \cup \{(\sigma, F) \mid \sigma \in \Sigma, \mathcal{B}[b_2](\sigma) = F\} \quad (52)$$

$$\mathcal{B}[b_1 \ || \ b_2] = \{(\sigma, T) \mid \sigma \in \Sigma, \mathcal{B}[b_1](\sigma) = T\} \cup \{(\sigma, T) \mid \sigma \in \Sigma, \mathcal{B}[b_2](\sigma) = T\} \cup \{(\sigma, F) \mid \sigma \in \Sigma, \mathcal{B}[b_1](\sigma) = F, \mathcal{B}[b_2](\sigma) = F\} \quad (53)$$

$$\mathcal{B}[\!|b] = \{(\sigma, T) \mid \sigma \in \Sigma, \mathcal{B}[b](\sigma) = F\} \cup \{(\sigma, F) \mid \sigma \in \Sigma, \mathcal{B}[b](\sigma) = T\} \quad (54)$$

## Commands

$$\mathcal{C}[\text{skip}] = \{(\sigma, \sigma) \mid \sigma \in \Sigma\} \quad (55)$$

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

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

$$\mathcal{C}[\text{if } b \text{ then } c_1 \text{ else } c_2] = \{(\sigma, \mathcal{C}[c_1](\sigma)) \mid \sigma \in \Sigma, \mathcal{B}[b](\sigma) = T\} \cup \{(\sigma, \mathcal{C}[c_2](\sigma)) \mid \sigma \in \Sigma, \mathcal{B}[b](\sigma) = F\} \quad (58)$$

$$\mathcal{C}[\text{while } b \text{ do } c] = \bigcup_{i \geq 0} \Gamma^i(\perp_{\Sigma \rightarrow \Sigma}) = \text{fix}(\Gamma) \quad (59)$$

$$\text{where } \Gamma(g) = \{(\sigma, (g \circ \mathcal{C}[c])(\sigma)) \mid \sigma \in \Sigma, \mathcal{B}[b](\sigma) = T\} \cup \{(\sigma, \sigma) \mid \sigma \in \Sigma, \mathcal{B}[b](\sigma) = F\}$$