

Harvard School of Engineering and Applied Sciences — CS 152: Programming Languages
Concurrency, Type and effect system, Message passing
Section and Practice Problems

Apr 14–15, 2016

1 Concurrency

(a) Consider the following program.

$$(3 + 7) \parallel ((\lambda x. x + 1) 2 + 5)$$

Show an execution sequence for this program (i.e., give a sequence of expressions such that $e_0 \rightarrow e_1 \rightarrow \dots \rightarrow e_n$ where $e_0 = (3 + 7) \parallel ((\lambda x. x + 1) 2 + 5)$ and e_n is a value.

Now give a different execution sequence for this program.

How many different execution sequences of this program are there?

(b) Consider the following program.

```
let foo = ref 2 in
let y = (foo := !foo + !foo || foo := 1) in
!foo
```

What are the possible final values of the program?

2 Type and effect system

Recall the type and effect system to ensure determinacy, covered in Lecture 17.

(a) Consider the program (from class) of a bank balance, where the bank balance is in the region A .

$$\text{let bal} = \text{ref}_A 0 \text{ in } (\text{let } y = (\text{bal} := !\text{bal} + 25 \parallel \text{bal} := !\text{bal} + 50) \text{ in } !\text{bal})$$

Try to produce a typing derivation for this program (using the type-and-effect typing rules from lecture). Where do the typing rules fail? Why?

(b) Write a program that allocates two locations (in different regions) and reads and writes from both of them. Moreover, make sure that your program is well-typed according to the type-and-effect system. Is your program deterministic?

3 Synchronous sends and receives

- (a) To make sure you understand the operational semantics of sends and receives, show the execution of the following program. The initial configuration consists of a single thread, and your answer should show the sequence of configurations that program execution will step through.

```
let  $c = \text{newchan}_{\text{int}}$  in  
spawn (1 + recv from  $c$ );  
send 6 to  $c$ 
```

- (b) Consider the following program. What are the possible final values of the main (i.e., initial) thread?

```
let  $c = \text{newchan}_{\text{int}}$  in  
let  $d = \text{newchan}_{\text{int}}$  in  
let  $g = \text{newchan}_{\text{int chan}}$  in  
spawn send  $c$  to  $g$ ;  
spawn send  $d$  to  $g$ ;  
spawn send 3 to  $c$ ;  
spawn send 4 to  $c$ ;  
spawn send 5 to  $d$ ;  
spawn send 6 to  $d$ ;  
(recv from (recv from  $g$ )) + recv from  $c$ 
```

- (c) Write a program that creates a channel c , and then sends the Fibonacci sequence over the channel (i.e., it sends 0, 1, 1, 2, 3, 5, 8, ...). Your program will spawn a thread to compute and send the Fibonacci sequence, and the other thread will consume the values produced. (You may assume you have a recursive operator $\mu x. e.$)

4 First-class synchronization

- (a) Write a program that creates a channel c , and then sends the Fibonacci sequence over the channel (i.e., it sends 0, 1, 1, 2, 3, 5, 8, ...), but does not block waiting for the sequence to be consumed. Spawn one thread to compute and send the Fibonacci sequence, but make the other thread consume (i.e., receive) just a single value from the channel. What are the possible values received by the other thread? (You may assume you have a recursive operator $\mu x. e.$)

- (b) Consider the following program, which uses events. Using the translation given in class, translate it to the language with blocking sends and receives without events.

```
let  $c = \text{newchan}_{\text{int}}$  in  
spawn (sendEvt 8 to  $c$ ; send 9 to  $c$ );  
let  $r = \text{recvEvt}$  from  $c$  in  
(recv from  $c$ ) + sync  $r$ 
```