Dynamic Types, Concurrency, Type and effect system
Section and Practice Problems

Apr 24–27, 2018

---

1 Dynamic types and contracts

(a) To make sure you understand the operational semantics of dynamic types and exceptions, show the execution of the following program under the semantics of Section 1 of the Lecture 22 notes.

$$\text{let } f = \lambda x.\, 42 + x \text{ in}$$
$$\text{let } g = \lambda y.\, (y \text{ true}) + 42 \text{ in}$$
$$g\ f$$

(b) Modify the program from question (a) by adding appropriate error handlers (i.e., expressions of the form try e_1 catch $x.\ e_2$ to catch the type error and return the integer 42 as the final result of the program. There are multiple places in the program where you can insert an error handler to achieve the desired result. Show three variations and their executions. (Note that the semantics for the execution of your programs is from Part 2 (Exception handling) of the Lecture 22 notes.)

(c) Modify the program from question (a) by adding appropriate dynamic type checks to raise the error as early as possible. When does your program detect the error?

(d) Modify the program from question (a) by adding contracts that specify the types of the input and output of f and g. Show the execution of the modified program.

2 Concurrency

(a) Consider the following program.

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

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

$$\text{let foo} = \text{ref } 2 \text{ in}$$
$$\text{let } y = (\text{foo} := !\text{foo} + !\text{foo}\ ||\ \text{foo} := 1) \text{ in}$$
$$!\text{foo}$$

What are the possible final values of the program?

3   Type and effect system

Recall the type and effect system to ensure determinacy, covered in Lecture 25.

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

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

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?