

## 1995 Paper 5 Question 12

### Semantics

The abstract syntax of IMP commands is given by the following grammar:

$$\begin{aligned} Com ::= & \text{skip} \mid Pvar := Iexp \mid Com ; Com \mid \\ & \text{if } Bexp \text{ then } Com \text{ else } Com \mid \text{while } Bexp \text{ do } Com \end{aligned}$$

where  $Iexp$  and  $Bexp$  are syntactic categories of integer and boolean expressions and  $Pvar$  is a set of program variables. Let  $States$  be  $[Pvar \rightarrow \mathbb{Z}]$  and  $Cont$ , the cpo of *continuations*, be  $[States \rightarrow A_{\perp}]$ , where  $A$  is an unspecified set of program *answers*. A continuation represents what is to be done with the state resulting from the execution of a command in order to return the result of the whole program.

The *continuation semantics* of IMP is defined by giving the meaning  $\llbracket C \rrbracket$  of each  $C \in Com$  as a function which takes a continuation, representing what is to be done when the command has finished, together with a state in which the command is to be executed, and returns an answer:

$$\llbracket - \rrbracket : Com \rightarrow (Cont \rightarrow (States \rightarrow A_{\perp})).$$

One clause of the definition of  $\llbracket C \rrbracket$  is

$$\llbracket \text{skip} \rrbracket k S = k(S).$$

Complete the definition of the continuation semantics of IMP commands (expressing their usual behaviour). You may assume that the functions

$$\begin{aligned} \llbracket - \rrbracket & : Iexp \rightarrow (States \rightarrow \mathbb{Z}) \\ \llbracket - \rrbracket & : Bexp \rightarrow (States \rightarrow \mathbb{B}) \quad \text{where } \mathbb{B} = \{true, false\} \end{aligned}$$

have already been defined. [9 marks]

Now add a new command **abort** to  $Com$  and a new error value  $Err$  to  $A$ . The intended behaviour of **abort** is immediately to terminate the entire program, returning  $Err$ . Extend the continuation semantics of IMP by giving the definition of  $\llbracket \text{abort} \rrbracket$ . [2 marks]

Now add two further new command forms:

$$Com ::= \dots \mid \text{abort} \mid \text{exit} \mid Com \text{ orelse } Com$$

The intended behaviour of  $(C_1 \text{ orelse } C_2)$  is that it executes exactly like  $C_1$  unless  $C_1$  hits an **exit** command, in which case further execution of  $C_1$  is abandoned and  $C_2$  is executed starting in the state at which  $C_1$  encountered the **exit**. If  $C_1$  does not encounter an **exit** then  $C_2$  is ignored. An **exit** command without an enclosing **orelse** behaves like **abort**.

## 1995 Paper 5 Question 12 (continued)

Give a revised continuation semantics to every command of IMP with `abort`, `exit` and `orelse` which reflects this behaviour and in which the denotation of  $C \in Com$  is a function which takes *two* continuations and a state and returns an element of  $A_{\perp}$ :

$$\llbracket - \rrbracket : Com \rightarrow (Cont \rightarrow (Cont \rightarrow (States \rightarrow A_{\perp}))).$$

Hint: The first continuation is the ordinary default continuation and the second is the continuation to be applied if the command `exits`. [9 marks]