1994 Paper 7 Question 13

Semantics of Programming Languages

Dijkstra proposed the language of guarded commands with the following syntax. Commands take the form

$$c \quad ::= \quad \mathsf{skip} \quad | \quad \mathsf{abort} \quad | \quad X := e \quad | \quad c; c \quad | \quad \mathsf{if}\ gc\ \mathsf{fi} \quad | \quad \mathsf{do}\ gc\ \mathsf{od}$$

where e is an arithmetic expression and gc stands for a guarded command of the form

$$b_1 \to c_1 \, [\!] \, b_2 \to c_2$$

for boolean expressions b_1 and b_2, called *guards*, and commands c_1 and c_2. Execution of the command skip does not result in a change of state. Following Dijkstra's intentions, if no guard evaluates to true at a state, then the guarded command is said to fail, in which case, the guarded command does not yield a final state. Otherwise, the guarded command executes as one of the commands c_i whose associated guard b_i evaluates to true. The execution of the command abort does not yield a final state from any initial state. The command $\mathsf{if}\ gc\ \mathsf{fi}$ executes as the guarded command gc, if gc does not fail, otherwise, it acts like abort. The command $\mathsf{do}\ gc\ \mathsf{od}$ executes repeatedly as the guarded command gc, while gc continues not to fail, and terminates when gc fails.

(a) Assume that boolean and arithmetic expressions have no side effects and always terminate, and that the rules for their evaluation are given. Write down a collection of rules for an inductively defined evaluation relation of the form

$$c,\ S \ \Rightarrow\ S'$$

whose sense is "starting from the initial state S, the evaluation of the command c terminates at the final state S'." [10 marks]

(b) Give the commands in Dijkstra's guarded language which simulate the standard imperative programming commands

$$\mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2 \qquad \text{and} \qquad \mathsf{while}\ b\ \mathsf{do}\ c$$

respectively. You may assume that if b is a boolean expression, then so is $\neg b$, the negation of b. [2 marks]

(c) Give an appropriate definition of *semantic equivalence* of commands with respect to the evaluation relation defined in (a). Prove that for any boolean expression b and any command c, the command

$$\mathsf{do}\ b \to c \, [\!] \, b \to c\ \mathsf{od}$$

is semantically equivalent to the command

$$\mathsf{if}\ b \to (c; \mathsf{do}\ b \to c \, [\!] \, b \to c\ \mathsf{od}) \, [\!] \, \neg b \to \mathsf{skip}\ \mathsf{fi}$$ [8 marks]