Program executions

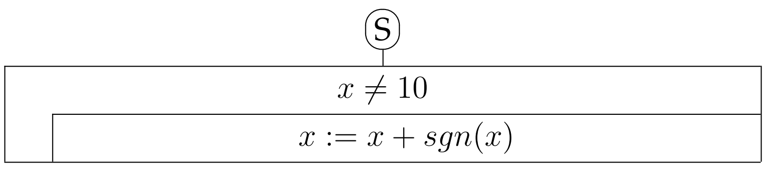
We defined the program as a relation that maps from the statespace A to the set of finite and infinite sequences containing elements of $\bar{A} \cup \{fail\}$ such that

- The program assigns at least one sequence to every state in the statespace A.
- Every sequence that is assigned to a state a by the program, begins with the state a (the same state to which it is assigned).
- In case there is a $fail$ state in a sequence (where the sequence represents a possible execution of the program, i.e. it is an element of $\mathcal{R}_S$), it has to be the last element of the sequence (indicating a failure).
- In case an execution of a program is finite, the last element of the sequence that represents the execution has to be a normal state (element of the statespace A, not an element of $\bar{A}$) or the special $fail$ state.

In this section we consider programs in order to analyse the sequences assigned to the states of the statespace by the programs.

1 Example1

Let $H = \{a \in \mathbb{Z} \mid a \geqslant -5\}$
$A = (x\colon H)$



In the program, sgn denotes the signum function. For simplicity, let us denote a state of the statespace by a, instead of using the official notation $\{x{:}a\}$.
As we know, any sequence assigned to the state a by the program S begins with the same state it is assigned to, a. We also know, that the loop will stop if its loopcondition becomes $false$, that is when the value of variable x is 10 (in other words, when we are in the state $\{x{:}10\}$).
Our program is a loop. Whenever the loopcondition holds, we execute the body of the loop. The loopbody is the assignment $x := x + sgn(x)$. Since $sgn(x) = -1$ for any x negative number, we decrease the value of x by 1 in the loopbody in case the value of x is less than zero. Notice, that -5 is the smallest possible value of x. The program $x := x + sgn(x)$ will abort executing from the state where $x = -5$.

- Write down the sequences assigned to the states 4, 13, -2, 0 and 10 by the program S.

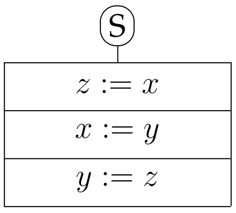
The finite sequence $< 4, 5, 6, 7, 8, 9, 10 >$ assigned to the state 4. The program assigns a finite sequence to the state -2 as well: $< -2, -3, -4, -5, fail >$, this sequence aims to express that the program terminates starting from the state -2, but decreasing the value of x in the state -5 causes a failure.

$S(13) = \{< 13, 14, 15, 16, \ldots >\}$
$S(0) = \{< 0, 0, 0, 0, \ldots >\}$
$S(10) = \{< 10 >\}$

2 Example2

Notice that it does matter what the base-statespace of a program is. Also remember, that due to the definition of program, every sequence assigned to a state begins with the same state it is assigned to, and (in case the sequence is finite) the last element of a sequence is a state of the base-statespace A, or the special $fail$ state. So, the definition of program allows for sequences to contain intermediate states from any statespace B where A is a subspace of B. In practice it means, that during the execution of the program, using of auxiliary variables is allowed; these variables are created and destroyed during the execution of the program, and their corresponding values are recorded while they are existing.

Consider the following program and let us suppose that the base-statespece of S is $A = (x{:}\mathbb{Z}, y{:}\mathbb{Z})$. It makes sense, as usually we use this program to solve the problem where we want to swap to integers x and y, this is why variable z (its type is also integer) can be considered as an auxiliary variable.



Let us write down the sequence that is assigned to the state $\{x{:}3, y{:}8\}$ by this program. Remember, that the sequence has to begin with the same state it is assigned to, and the last element of the sequence (in case it is a finite sequence) is from statespace A or (in case the program terminates due to a failure) is the $fail$ state. During the execution of the program we record the actual values that belong to the variable z, which means the intermediate states of the sequence are elements of the statespace $(x{:}\mathbb{Z}, y{:}\ \mathbb{Z}, z{:}\mathbb{Z})$. As we know, the program $y := z$ sets the value of y to the value that belongs to z and does not change the value of x and z. Our S program is a sequence of three assignments and the following sequence is assigned to the state $\{x{:}3, y{:}8\}$:

$$< \{x{:}3, y{:}8\}, \{x{:}3, y{:}8, z{:}3\}, \{x{:}8, y{:}8, z{:}3\}, \{x{:}8, y{:}3\} >$$