Series 8

Defined and Used variables

We recall the syntax of the **While** language:

$$\begin{array}{lcl} S & ::= & x := a \mid \texttt{skip} \mid S; S \mid \texttt{if}\ b\ \texttt{then}\ S\ \texttt{else}\ S \mid \texttt{while}\ b\ \texttt{do}\ S \\ a & ::= & n \mid x \mid a + a \\ b & := & \texttt{true} \mid \texttt{false} \mid a = a \mid a \leq a \mid \neg b \mid b \wedge b \end{array}$$

Defining assigned and used variables in statements

We want to define the functions `Def` and `Use` which associate, to each syntactic construct, the set of assigned and used variables respectively. Assigned variables are the variables appearing in the left-hand side of an assignment. Used variables are the variables appearing on the right-hand side of assignments or in expressions.

Exercise 78 (A formal definition of `Def` and `Use`) We want to define formally functions `Def` and `Use`.

1. What is the signature of these functions?
2. Give a formal definitions of those two functions.

Exercise 79 (Applying `Def` and `Use`) Apply functions `Def` and `Use` to the following code snippet

```
if a < b then c := d-y else y := e -x.
```

Adding an operator $\|$ for parallel execution

We extend the syntax of statements in the following way:

$$S ::= \cdots \mid S \parallel S.$$

Exercise 80 (Parallelism in Natural Operational Semantics) Recall the natural operational semantics of this operator.

In the following, we consider the **While** language extended with the $\|$ operator and its natural operational semantics.

Exercise 81 (Applying the parallelism operator) Applying the semantics natural operational semantics rules extended with the rules for $\|$, compute the obtained state(s) by executing the following commands on the initial state $\sigma_0 = [x \mapsto 1, y \mapsto 1]$.

1. $x := x + 1 \parallel y := y + 1$
2. $x := y + 1 \parallel y := x + 2$

Exercise 82 (A sufficient condition so that parallelism does not influence computation) Give a sufficient condition such that the two previously defined rules yield the same result. That is, when we evaluate $S_1 \parallel S_2$ in a state σ, then the obtained state σ_2 is independent of the order of evaluation of S_1 and S_2.

Hint: One can use the functions `Def` and `Use`, previously defined.

Exercise 83 Propose a unique semantic rule that evaluates $S_1 \parallel S_2$ on the state σ. This rules should propose an evaluation of S_1 and S_2 on state σ supposing that the previous condition holds.

Exercise 84 Apply the previously defined semantic rules to the statement of Exercise 81.