Exercise Sheet 2

Due date: November 8^{th}. You can hand in your solutions at the start of the exercise class.

Hint: Notation is as in the lecture. That is, c is a program, b a Boolean expression, σ a program state, etc.

Task 1: Operational Semantics & Derivation Trees (2 points)

Consider the following program:

```
c:  x := 23; y := 42;
    while(x ≤ y) do
        y := y - x; x := x - 4
    end
```

Depict the derivation tree for $\langle c, \sigma \rangle \to \sigma'$, where σ is some arbitrary, but fixed, initial state.

Task 2: Operational Semantics of other Statements (1 point)

Extend the rule system defining the (big-step) execution relation $\to$ from the lecture (Def. 3.2) to incorporate for a statement `repeat` c `until` b.

Task 3: Termination (2 points)

Prove that $\langle \texttt{while}\ b\ \texttt{do}\ c\ ,\ \sigma \rangle \to \sigma'$ implies that $\langle b, \sigma' \rangle \to \texttt{false}$.

Task 4: Variables that do not matter (5 points)

In this exercise, we use a variant of the WHILE language from the lecture that neither contains `if-then-else` constructs nor `while` loops. It does, however, contain `repeat-until` loops.

(a) Define a recursive function

$$\texttt{mod}:\quad \texttt{Cmd} \to 2^{Var}$$

that computes the set of all variables that are modified by a program. That is, those variables that occur on the left-hand side of assignments.

(b) Define a recursive function

$$\texttt{dep}:\quad \texttt{Cmd} \to 2^{Var}$$

that computes the set of variables that are read by a program. That is, those variables that occur on the right-hand side of assignments or in loop guards.

Hint: You may use the function $FV : \texttt{AExp} \cup \texttt{BExp} \to \mathcal{P}(Var)$ that computes the set of free variables of an arithmetic or Boolean expression(see Def. 2.4 for a definition restricted to arithmetic expressions).

(c) We consider two program states σ_1, σ_2 *equivalent* with respect to a set of variables $R \subseteq \mathit{Var}$, written $\sigma_1 =_R \sigma_2$ if they coincide for all variables in R. Formally,

$$\sigma_1 =_R \sigma_2 \quad \text{iff} \quad \forall x \in R \ : \ \sigma_1(x) = \sigma_2(x)$$

Show for every program c and states σ_1, σ_2 with

- $\sigma_1 =_{\mathtt{dep}(c)} \sigma_2$,
- $\langle c, \sigma_1 \rangle \to \sigma_1'$, and
- $\langle c, \sigma_2 \rangle \to \sigma_2'$

that $\sigma_1' =_{\mathtt{mod}(c)} \sigma_2'$.

Hint: You may use the following auxiliary results without proof:

(a) $\langle c, \sigma \rangle \to \sigma'$ and $x \notin \mathtt{mod}(c)$ implies $\sigma'(x) = \sigma(x)$.

(b) $\sigma_1 =_{\mathtt{dep}(c)} \sigma_2$ implies $(\exists \sigma_1' : \langle c, \sigma_1 \rangle \to \sigma_1'$ iff $\exists \sigma_2' : \langle c, \sigma_2 \rangle \to \sigma_2')$.