Exersice Sheet 1

——— Sample Solution ———

Task 1: Recursion and Structural Induction

(a)

$$
\begin{aligned}
z\,[x := a'] &= z \\
y\,[x := a'] &= \begin{cases} a', & \text{if } x = y \\ y, & \text{if } x \neq y \end{cases} && y \in \text{Var} \\
(a_1 \oplus a_2)\,[x := a'] &= a_1\,[x := a'] \oplus a_2\,[x := a'] && \text{for } \oplus \in \{+,\ -,\ *\}
\end{aligned}
$$

(b)

occ : AExp x Var $\to \mathbb{N}$

$$
\begin{aligned}
\text{occ}\,(z,\ x) &= 0 \\
\text{occ}\,(y,\ x) &= \begin{cases} 1, & \text{if } x = y \\ 0, & \text{if } x \neq y \end{cases} && y \in \text{Var} \\
\text{occ}\,(a_1 \oplus a_2,\ x) &= \text{occ}\,(a_1,\ x) + \text{occ}\,(a_2,\ x) && \text{for } \oplus \in \{+,\ -,\ *\}
\end{aligned}
$$

(c)

The statement $\text{FV}\,(a\,[x := a']) \subseteq (\text{FV}\,(a) \setminus \{x\}) \cup \text{FV}\,(a')$ will by shown by structural induction on the structure of arithmetic expressions a.

<u>Induction Base:</u>

<u>Case 1:</u> $a = z,\ z \in \mathbb{Z}$

$$\text{FV}\,(z\,[x := a']) \overset{(a)}{=} \underbrace{\text{FV}\,(z)}_{=\emptyset} = \text{FV}\,(z) \setminus \{x\} \subseteq (\text{FV}\,(z) \setminus \{x\}) \cup \text{FV}\,(a')$$

<u>Case 2:</u> $a = y,\ y \neq x \quad y \in \text{Var}$

$$\text{FV}\,(y\,[x := a']) \overset{(a)}{=} \underbrace{\text{FV}\,(y)}_{=\{y\}} \subseteq (\text{FV}\,(y) \setminus \{x\}) \cup \text{FV}\,(a')$$

<u>Case 3:</u> $a = x$

$$\text{FV}\,(x\,[x := a']) \overset{(a)}{=} \text{FV}\,(a') \subseteq (\text{FV}\,(y) \setminus \{x\}) \cup \text{FV}\,(a')$$

Induction Hypothesis:

Assume for all subexpressions $\bar{a}$ of a that
$\mathrm{FV}(\bar{a}[x := a']) \subseteq (\mathrm{FV}(\bar{a}) \setminus \{x\}) \cup \mathrm{FV}(a')$ holds.

Induction Step:

Let $\oplus \in \{+, -, *\}$ and $a_1,\ a_2 \in \mathrm{AExp}$.

$$
\begin{aligned}
&\mathrm{FV}((a_1 \oplus a_2)[x := a']) \overset{(a)}{=} \mathrm{FV}(a_1 [x := a_1] \oplus a_2 [x := a'])\\
&\overset{(\text{Def } 2.4)}{=} \mathrm{FV}(a_1 [x := a']) \cup \mathrm{FV}(a_2 [x := a'])\\
&\overset{\textbf{I.H.}}{\subseteq} (\mathrm{FV}(a_1) \setminus \{x\}) \cup \mathrm{FV}(a') \cup (\mathrm{FV}(a_2) \setminus \{x\}) \cup \mathrm{FV}(a')\\
&= (\mathrm{FV}(a_1) \cup \mathrm{FV}(a_2)) \setminus \{x\} \cup \mathrm{FV}(a')\\
&\overset{(\text{Def } 2.4)}{=} (\mathrm{FV}(a_1 \oplus a_2)) \setminus \{x\} \cup \mathrm{FV}(a')
\end{aligned}
$$

□

(d)

(i)

$$\mathrm{length}(a[x := a']) = \mathrm{length}(a) + \mathrm{occ}(a,\ x) \cdot (\mathrm{length}(a') - 1)$$

(ii)

The correctness of the proposed formula will be proven by structural induction on the structure of arithmetic expressions a.

Induction Base:

Case 1: $a = z,\ z \in \mathbb{Z}$

$$
\begin{aligned}
\mathrm{length}(z[x := a']) &\overset{(a)}{=} \mathrm{length}(z)\\
&= \mathrm{length}(z) + 0 \cdot (\mathrm{length}(a') - 1)\\
&\overset{(b)}{=} \mathrm{length}(z) + \mathrm{occ}(z,\ x) \cdot (\mathrm{length}(a') - 1)
\end{aligned}
$$

Case 2: $a = y,\ y \neq x \quad y \in \mathrm{Var}$

$$
\begin{aligned}
\mathrm{length}(y[x := a']) &\overset{(a)}{=} \mathrm{length}(y)\\
&= \mathrm{length}(y) + 0 \cdot (\mathrm{length}(a') - 1)\\
&\overset{(b)}{=} \mathrm{length}(y) + \mathrm{occ}(y,\ x) \cdot (\mathrm{length}(a') - 1)
\end{aligned}
$$

Case 3: $a = x$

$$
\begin{aligned}
\mathrm{length}(x[x := a']) &\overset{(a)}{=} \mathrm{length}(a')\\
&= 1 + \mathrm{length}(a') - 1\\
&= 1 + 1 \cdot (\mathrm{length}(a') - 1)\\
&\overset{(b)}{=} \mathrm{length}(x) + \mathrm{occ}(x,\ x) \cdot (\mathrm{length}(a') - 1)
\end{aligned}
$$

Induction Hypothesis:

Assume for all subexpressions $\bar{a}$ of a that
$\text{length}(\bar{a}[x := a']) = \text{length}(\bar{a}) + \text{occ}(\bar{a}, x) \cdot (\text{length}(a') - 1)$ holds.

Induction Step:

Let $\oplus \in \{+, -, *\}$ and $a_1, a_2 \in \text{AExp}$.

$$\begin{aligned}
&\text{length}((a_1 \oplus a_2)[x := a']) \overset{(a)}{=} \text{length}(a_1[x := a'] \oplus a_2[x := a']) \\
&\overset{\text{def length}}{=} 1 + \text{length}(a_1[x := a']) + \text{length}(a_2[x := a']) \\
&\overset{\textbf{I.H.}}{=} 1 + (\text{length}(a_1) + \text{occ}(a_1, x) \cdot (\text{length}(a') - 1)) + \text{length}(a_2[x := a']) \\
&\overset{\textbf{I.H.}}{=} 1 + (\text{length}(a_1) + \text{occ}(a_1, x) \cdot (\text{length}(a') - 1)) \\
&\qquad + (\text{length}(a_2) + \text{occ}(a_2, x) \cdot (\text{length}(a') - 1)) \\
&= (1 + \text{length}(a_1) + \text{length}(a_2)) \\
&\qquad + (\text{occ}(a_1, x) + \text{occ}(a_2, x)) \cdot (\text{length}(a') - 1) \\
&\overset{\text{Def}}{=} \text{length}(a_1 \oplus a_2) + \text{occ}(a_1 \oplus a_2, x) \cdot (\text{length}(a') - 1)
\end{aligned}$$

$\square$

Task 2: The Programming Language WHILE

(a)

```
x := 1;          //f_0
y := 1;          //f_1
i := 1;
while (i < n) do
    t := x + y;      //f_{n+2}
    x := y;          //f_{n+1}
    y := t;          //f_{n+2}
    i := i + 1
end
```

(b)