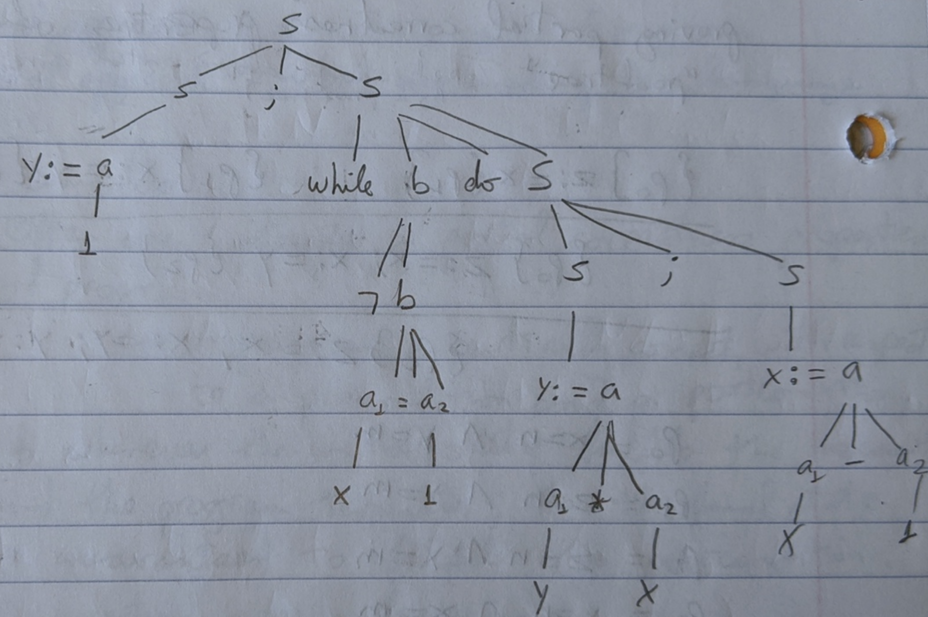
Exercise 1.2

$y := 1;\ \text{while}\ \neg(x=1)\ \text{do}\ (y := y * x;\ x := x - 1)$



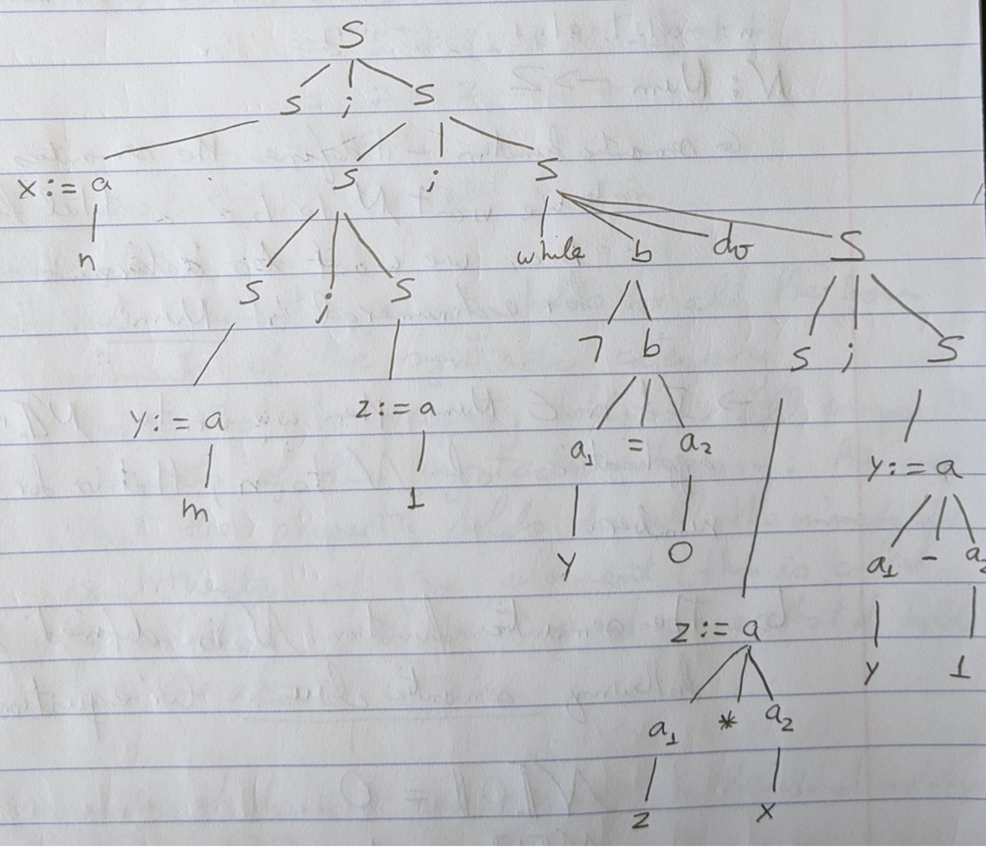
Exercise 1.2

Assume $x := n;$

$y := m;$

$\leadsto z := \underbrace{n * \cdots * n}_{m \text{ times}}$

$x := n;\ y := m;\ z := 1;\ \text{while } \neg(y = 0) \text{ do } (z := z * x;\ y := y - 1)$



Exercise 1.8 Assume that $s\ x = 3$ and determine
$\mathcal{B}[\![\neg(x=1)]\!]s$.

$$\mathcal{B}[\![\neg(x=1)]\!]s = \mathcal{B}[\![x=1]\!]s = ?$$
$$(\mathcal{A}[\![x]\!]s \ ? \ \mathcal{A}[\![1]\!]s) = ?$$
$$(s\ x \ ? \ \mathcal{N}[\![1]\!]) = ?$$
$$(3 \neq 1) = ?$$
$$(3 \neq 1) = ff$$
$$= tt$$

x	y	z
17	5	0
12	5	1
7	5	2
2	5	3

Exercise 2.3 Consider the statement

$$z := 0;\ \text{while } y \le x \text{ do } (z := z+1;\ x := x - y)$$

Construct a derivation tree for this statement when executed in a state where x has the value 17 and y has the value 5.

$$\frac{\langle z := 0,\ s\rangle \to s_{1750} \qquad T_1}{\langle z := 0;\ \text{while } y \le x \text{ do } (z := z+1;\ x := x-y),\ s\rangle \to s_{253}}$$

where $s_{253} = s[x \mapsto 2][y \mapsto 5][z \mapsto 3]$

$s_{1750} = s[x \mapsto 17][y \mapsto 5][z \mapsto 0]$

T_1 root $\to$ $\langle \text{while } y \le x \text{ do } (z := z+1;\ x := x-y),\ s_{1750}\rangle \to s_{253}$

$\mathcal{B}[\![y \le x]\!] s_{1750} = tt$

T_1:

$$\frac{T_2 \qquad T_3}{\langle \text{while } y \le x \text{ do } (z := z+1;\ x := x-y),\ s_{1750}\rangle \to s_{253}}$$

T_2 root $\to$ $\langle z := z+1;\ x := x-y,\ s_{1750}\rangle \to s_{1251}$

T_3 root $\to$ $\langle \text{while } y \le x \text{ do } (z := z+1;\ x := x-y),\ s_{1251}\rangle \to s_{253}$

T_2:

$$\frac{\langle z := z+1,\ s_{1750}\rangle \to s_{1751} \qquad \langle x := x-y,\ s_{1751}\rangle \to s_{1251}}{\langle z := z+1;\ x := x-y,\ s_{1750}\rangle \to s_{1251}}$$

where $s_{1751} = s[x \mapsto 17][y \mapsto 5][z \mapsto 1]$

$s_{1251} = s[x \mapsto 12][y \mapsto 5][z \mapsto 1]$

T_3:

$$\frac{T_4 \qquad\qquad T_5}{\langle \text{while } y \le x \text{ do } (z := z+1;\ x := x-y),\ s_{1251} \rangle \to s_{253}}$$

$T_{4\,root} \to \langle z := z+1;\ x := x-y,\ s_{1251} \rangle \to s_{752}$

$T_{5\,root} \to \langle \text{while } y \le x \text{ do } (z := z+1;\ x := x-y),\ s_{752} \rangle \to s_{253}$

T_4:

$$\frac{\langle z : z+1,\ s_{1251} \rangle \to s_{1252} \qquad \langle x := x-y,\ s_{1252} \rangle \to s_{752}}{\langle z : z+1;\ x := x-y,\ s_{1251} \rangle \to s_{752}}$$

$\boxed{\text{where: } \underline{s_{1252}} = s[x \mapsto 12][y \mapsto 5][z \mapsto 2];\ \underline{s_{752}} = s[x \mapsto 7][y \mapsto 5][z \mapsto 2]}$

T_5:

$$\frac{\dfrac{\langle z : z+1,\ s_{752} \rangle \to s_{753} \qquad \langle x := x-y,\ s_{753} \rangle \to s_{253}}{\langle z := z+1;\ x := x-y,\ s_{752} \rangle \to s_{253}} \qquad T_6}{\langle \text{while } y \le x \text{ do } (z := z+1;\ x := x-y),\ s_{752} \rangle \to s_{253}}$$

$T_{6\,root}$: $\langle \text{while } y \le x \text{ do } (z : z+1;\ x := x-y),\ s_{253} \rangle \to s_{253}$

$\hookrightarrow \mathcal{B}[\![y \le x]\!]\, s_{253} = ff$

where $s_{753} = s[x \mapsto 7][y \mapsto 5][z \mapsto 3]$

Exercise 2.4 Consider the following statements

- while $\neg(x=1)$ do $(y:=y*x;\ x:=x-1)$
- while $1 \le x$ do $(y:=y*x;\ x=x-1)$
- while true do skip

For each statement determine whether or not it always terminates and whether or not it always loops.

- while $\neg(x=1)$ do $(y:=y*x;\ x:=x-1)$

for instance, $s\ x=3;\ y:=1$ s_{61}

$$\frac{\langle y:=y*x;\ x:=x-1,\ s_{13}\rangle \to s_{32},\quad \langle \text{while } \neg(x=1) \text{ do}(\ldots)\ s_{32}\rangle}{\langle \text{while } \neg(x=1) \text{ do } (y:=y*x;\ x:=x-1),\ s_{13}\rangle \to s_{61}}$$

x will get to -1 and will terminate

↳ $\boxed{\text{Always Terminates}}$

- while $1 \le x$ do $(y:=y*x;\ x=x-1)$

for instance, $s\ x=3;\ y=1$

↳ same as before x will be less than 1 and will

$\boxed{\text{Always terminates}}$

$$\frac{\langle y:=y*x;\ x:=x-1,\ s_{13}\rangle \to s_{32},\quad \langle \text{while } 1\le x \text{ do}(\ldots)\ s_{32} \to s_{61}}{\langle \text{while } 1 \le x \text{ do } (y:=y*x;\ x=x-1),\ s_{13}\rangle \to s_{61}}$$

- while true do skip

↳ $\boxed{\text{always loops}}$

$\mathcal{B}[\![\text{true}]\!]\, s = tt$

$$\frac{\langle \text{skip}, s\rangle \to s,\quad \langle \text{while true do skip}, s\rangle \to s''}{\langle \text{while true do skip}, s\rangle \to s''}$$

↳ we keep looping

Exercise 2.6 Construct a statement showing that $S_1; S_2$ is not, in general, semantically equivalent to $S_2; S_1$

Suppose $s\ y = 0$
$s\ x = 2$

$S_1 \to y := 1$
$S_2 \to \text{while } \neg(x=1) \text{ do } (y := y * x;\ x := x-1)$

$$
\begin{array}{c}
\langle y := y*x,\ s_{21}\rangle \to s_{22}, \quad \langle x := x-1,\ s_{22}\rangle \to s_{12} \\ \hline
\langle y := y*x;\ x := x-1,\ s_{21}\rangle \to s_{12}, \quad \langle \text{while } \neg(x=1) \text{ do } (\cdots),\ s_{12}\rangle \to s_{12} \\ \hline
\langle y := 1,\ s\rangle \to s_{21}, \quad \langle \text{while } \neg(x=1) \text{ do } (y := y*x;\ x := x-1),\ s_{21}\rangle \to s_{12} \\ \hline
\langle y := 1;\ \text{while } \neg(x=1) \text{ do } (y := y*x;\ x := x-1),\ s\rangle \to s_{12}
\end{array}
$$

where $s_{12} = s[x \mapsto 1][y \mapsto 2]$

$S_1 \to \text{while } \neg(x=1) \text{ do } (y := y * x;\ x := x-1)$
$S_2 \to y := 1$

$$
\begin{array}{c}
\langle y := y*x,\ s\rangle \to s_{20}, \quad \langle x := x-1,\ s_{20}\rangle \to s_{10} \\ \hline
\langle y := y*x;\ x := x-1,\ s\rangle \to s_{10}, \quad \langle \text{while } \neg(x=1) \text{ do } (\cdots),\ s_{10}\rangle \to s_{10} \\ \hline
\langle \text{while } \neg(x=1) \text{ do } (y := y*x;\ x := x-1),\ s\rangle \to s_{10}, \quad \langle y := 1,\ s_{10}\rangle \to s_{11} \\ \hline
\langle \text{while } \neg(x=1) \text{ do } (y := y*x;\ x := x-1);\ y := 1,\ s\rangle \to s_{11}
\end{array}
$$

where $s_{11} = s[x \mapsto 1][y \mapsto 1]$

$\Rightarrow$ $S_1; S_2$ is not semantically equivalent to $S_2; S_1$ in this example

Exercise 2.11 The semantics of arithmetic expressions is given by the function $\mathcal{A}$. We can also use an operational approach and define a natural semantics for the arithmetic expressions. It will have to kinds of configurations

$\langle a, s \rangle$ denoting that $\underline{a}$ has to be evaluated in state $\underline{s}$, and

z denoting the final value (an element of $\mathbb{Z}$)

The transition relation $\rightarrow_{Aexp}$ has the form

$\langle a, s \rangle \rightarrow_{Aexp} z$

where the idea is that $\underline{a}$ evaluates to $\underline{z}$ in state $\underline{s}$

Transition system:

$$\langle n, s \rangle \rightarrow_{Aexp} \mathcal{N}[\![n]\!]$$

$$\langle x, s \rangle \rightarrow_{Aexp} s\,x$$

$$\frac{\langle a_1, s \rangle \rightarrow_{Aexp} z_1, \quad \langle a_2, s \rangle \rightarrow_{Aexp} z_2}{\langle a_1 + a_2, s \rangle \rightarrow_{Aexp} z} \quad \text{where } z = z_1 + z_2$$

$$\frac{\langle a_1, s \rangle \rightarrow_{Aexp} z_1, \quad \langle a_2, s \rangle \rightarrow_{Aexp} z_2}{\langle a_1 * a_2, s \rangle \rightarrow_{Aexp} z} \quad \text{where } z = z_1 * z_2$$

$$\frac{\langle a_1, s \rangle \rightarrow_{Aexp} z_1, \quad \langle a_2, s \rangle \rightarrow_{Aexp} z_2}{\langle a_1 - a_2, s \rangle \rightarrow_{Aexp} z} \quad \text{where } z = z_1 - z_2$$

Exercise 2.17 Extend <u>While</u> with the construct repeat S until b and specify the structural operational semantics for it. (The semantics for the repeat construct is not allowed to rely on the existence of a while-construct).

$[\text{repeat}_{sos}]$ $\quad \langle \text{repeat } S \text{ until } b, s\rangle \Rightarrow$

$\langle S; \text{ if } b \text{ then skip else (repeat } S \text{ until } b), s\rangle$

Exercise 2.33 Consider the statement

$x := -1$; while $x \leq 0$ do $(x := x - 1$ or $x := (-1) * x)$

Given a state $\underline{s}$ describe the set of final states that may result according to the natural semantics. Further describe the set of derivation sequences that are specified by the structural operational semantics. Based on this discuss whether or not you would regard the n.s as being equiv to the s.o.s for this particular statement.

n.s:

$$\dfrac{\langle x:=-1, s\rangle \to s_{-1} \quad \dfrac{\dfrac{\langle x:=(-1)*x,\ s_{-1}\rangle \to s_1}{\langle x:=x-1 \text{ or } x:=(-1)*x,\ s_{-1}\rangle \to s_1 \quad \text{while } x\le 0 \text{ do}(\dots),\ s_1 \to s_1}}{\text{while } x\le 0 \text{ do}(\dots \text{ or } \dots),\ s_{-1} \to s_1}}{\langle x:=-1;\ \text{while } x\le 0 \text{ do } (x:=x-1 \text{ or } x:=(-1)*x),\ s\rangle \to s_1}$$

where $s_{-1} = s[x \mapsto -1]$

and $s_1 = s[x \mapsto 1]$

↳ We decide to execute S_2 in the S_1 or S_2, everytime we go into the while statement, so that we don't loop forever.

S.O.S.:

$\langle x:=-1;$ while $x\le 0$ do $(x:=x-1$ or $x:=(-1)*x),\ s\rangle$

$\Rightarrow \langle$while $x\le 0$ do$(x:=x-1$ or $x:=(-1)*x),\ s[x\mapsto -1]\rangle$

$\Rightarrow \langle$if $x\le 0$ then $((x:=x-1$ or $x:=(-1)*x);$

while $x\le 0$ do $(x:=x-1$ or $x:=(-1)*x))$

else skip, $s[x\mapsto -1]\rangle$

$\Rightarrow \langle x:=x-1$ or $x:=(-1)*x$; while $x\le 0$ do $(\dots),\ s[x\mapsto 1]\rangle$

$\Rightarrow \langle$ while $x\le 0$ do$(x:=x-1$ or $x:=(-1)*x),\ s[x\mapsto 1]\rangle$

$\Rightarrow \langle$ if $x\le 0$ then $((x:=x-1$ or $x:=(-1)*x);$

while $x\le 0$ do $(x:=x-1$ or $x:=(-1)*x))$

else skip, $s[x\mapsto 1]\rangle$

$\Rightarrow \langle$skip, $s[x\mapsto 1]\rangle$

$\Rightarrow s[x\mapsto 1]$

↳ In the same way we decide to choose S_2

$\Rightarrow$ We can see the difference in the number of steps while using the 2 different approaches