IMP: A Simple Imperative Language
Big-step Semantics
CPSC 509: Programming Language Principles

Ronald Garcia[*]

24 October 2016
(**Time Stamp**: 15:11, Tuesday 19th March, 2024)

Syntax

$$n \in \mathbb{Z}, \quad bv \in \textsc{Bool}, \quad X \in \textsc{Loc}, \quad a \in \textsc{AExp}, \quad b \in \textsc{BExp}, \quad c \in \textsc{Com},$$

$$\begin{array}{rcl}
a & ::= & X \mid n \mid a \mathbin{\textsf{+}} a \mid a \mathbin{\textsf{-}} a \mid a * a \\
b & ::= & \textsf{true} \mid \textsf{false} \mid a = a \mid a \leq a \mid \neg b \mid b \wedge b \mid b \vee b \\
c & ::= & \textsf{skip} \mid X \mathbin{\textsf{:=}} a \mid c ; c \mid \textsf{if}\ b\ \textsf{then}\ c\ \textsf{else}\ c \mid \textsf{while}\ b\ \textsf{do}\ c \\
bv & ::= & \textsf{true} \mid \textsf{false}
\end{array}$$

Big-step Semantics

$$\sigma \in \textsc{Store} = \textsc{Loc} \to \mathbb{Z}$$
$$\textsc{ACfg} = \textsc{AExp} \times \textsc{Store}, \quad \textsc{BCfg} = \textsc{BExp} \times \textsc{Store}, \quad \textsc{CCfg} = \textsc{Com} \times \textsc{Store}$$

$$\sigma_z \in \textsc{Store}$$
$$\sigma_z(X) = 0$$

$$\begin{array}{l}
\cdot[\cdot \mapsto \cdot] : \textsc{Store} \times \textsc{Loc} \times \mathbb{Z} \to \textsc{Store} \\
\sigma[X_0 \mapsto n](X_0) = n \\
\sigma[X_0 \mapsto n](X_1) = \sigma(X_1) \quad \text{if } X_0 \neq X_1
\end{array}$$

$\boxed{\Downarrow_{\text{AExp}} \subseteq \text{ACfg} \times \mathbb{Z}}$

$$\frac{}{\langle n, \sigma\rangle \Downarrow_{\text{AExp}} n}\ (\text{enum}) \qquad \frac{}{\langle X, \sigma\rangle \Downarrow_{\text{AExp}} \sigma(X)}\ (\text{eloc}) \qquad \frac{\langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \quad \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2}{\langle a_1 \mathbin{\texttt{+}} a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_1 + n_2}\ (\text{eplus})$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \quad \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2}{\langle a_1 \mathbin{\texttt{-}} a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_1 - n_2}\ (\text{eminus}) \qquad \frac{\langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \quad \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2}{\langle a_1 * a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_1 * n_2}\ (\text{etimes})$$

$\boxed{\Downarrow_{\text{BExp}} \subseteq \text{BCfg} \times \text{Bool}}$

$$\frac{}{\langle \texttt{true}, \sigma\rangle \Downarrow_{\text{BExp}} \texttt{true}}\ (\text{etrue}) \qquad \frac{}{\langle \texttt{false}, \sigma\rangle \Downarrow_{\text{BExp}} \texttt{false}}\ (\text{efalse})$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \quad \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2}{\langle a_1 \mathbin{\texttt{=}} a_2, \sigma\rangle \Downarrow_{\text{BExp}} bv}\ (\text{eeq}) \begin{cases} bv = \texttt{true} \text{ if } n_1 = n_2 \\ bv = \texttt{false} \text{ if } n_1 \neq n_2 \end{cases}$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \quad \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2}{\langle a_1 \leq a_2, \sigma\rangle \Downarrow_{\text{BExp}} bv}\ (\text{eleq}) \begin{cases} bv = \texttt{true} \text{ if } n_1 \leq n_2 \\ bv = \texttt{false} \text{ if } n_1 > n_2 \end{cases}$$

$$\frac{\langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv_1}{\langle \neg b, \sigma\rangle \Downarrow_{\text{BExp}} bv_2}\ (\text{enot}) \begin{cases} bv_2 = \texttt{true} \text{ if } bv_1 = \texttt{false} \\ bv_2 = \texttt{false} \text{ if } bv_1 = \texttt{true} \end{cases}$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow_{\text{BExp}} bv_1 \quad \langle b_2, \sigma\rangle \Downarrow_{\text{BExp}} bv_2}{\langle b_1 \wedge b_2, \sigma\rangle \Downarrow_{\text{BExp}} bv_3}\ (\text{eand}) \begin{cases} bv_3 = \texttt{true} \text{ if } bv_1 = bv_2 = \texttt{true} \\ bv_3 = \texttt{false} \text{ if otherwise} \end{cases}$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow_{\text{BExp}} bv_1 \quad \langle b_2, \sigma\rangle \Downarrow_{\text{BExp}} bv_2}{\langle b_1 \vee b_2, \sigma\rangle \Downarrow_{\text{BExp}} bv_3}\ (\text{eor}) \begin{cases} bv_3 = \texttt{true} \text{ if } bv_1 = \texttt{true} \text{ or } bv_2 = \texttt{true} \\ bv_3 = \texttt{false} \text{ if otherwise} \end{cases}$$

$\boxed{\Downarrow_{\text{Com}} \subseteq \text{CCfg} \times \text{Store}}$

$$\frac{}{\langle \texttt{skip}, \sigma\rangle \Downarrow_{\text{Com}} \sigma}\ (\text{eskip}) \qquad \frac{\langle a, \sigma\rangle \Downarrow_{\text{AExp}} n}{\langle X \mathbin{\texttt{:=}} a, \sigma\rangle \Downarrow_{\text{Com}} \sigma[X \mapsto n]}\ (\text{eassign})$$

$$\frac{\langle c_1, \sigma\rangle \Downarrow_{\text{Com}} \sigma' \quad \langle c_2, \sigma'\rangle \Downarrow_{\text{Com}} \sigma''}{\langle c_1 \texttt{;} c_2, \sigma\rangle \Downarrow_{\text{Com}} \sigma''}\ (\text{eseq}) \qquad \frac{\langle b, \sigma\rangle \Downarrow_{\text{BExp}} \texttt{true} \quad \langle c_1, \sigma\rangle \Downarrow_{\text{Com}} \sigma'}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow_{\text{Com}} \sigma'}\ (\text{eif-t})$$

$$\frac{\langle b, \sigma\rangle \Downarrow_{\text{BExp}} \texttt{false} \quad \langle c_2, \sigma\rangle \Downarrow_{\text{Com}} \sigma'}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow_{\text{Com}} \sigma'}\ (\text{eif-f}) \qquad \frac{\langle b, \sigma\rangle \Downarrow_{\text{BExp}} \texttt{false}}{\langle \texttt{while } b \texttt{ do } c, \sigma\rangle \Downarrow_{\text{Com}} \sigma}\ (\text{ewhile-f})$$

$$\frac{\langle b, \sigma\rangle \Downarrow_{\text{BExp}} \texttt{true} \quad \langle c, \sigma\rangle \Downarrow_{\text{Com}} \sigma' \quad \langle \texttt{while } b \texttt{ do } c, \sigma'\rangle \Downarrow_{\text{Com}} \sigma''}{\langle \texttt{while } b \texttt{ do } c, \sigma\rangle \Downarrow_{\text{Com}} \sigma''}\ (\text{ewhile-t})$$

$$\text{Pgm} = \text{Com}, \quad \text{Obs} = \text{Store} \cup \{\infty\}$$

$$\begin{aligned} &eval_{IMP} : \text{Pgm} \stackrel{\text{dens}}{\to} \text{Obs} \\ &eval_{IMP}(c) = \sigma \text{ if } \langle c, \sigma_z\rangle \Downarrow_{\text{Com}} \sigma \\ &eval_{IMP}(c) = \infty \text{ otherwise} \end{aligned}$$

Reasoning Principles

Proposition 1 (Backward Reasoning for $\langle a, \sigma \rangle \Downarrow_{\text{AExp}} n$, Distinguishing a)**.**

1. $\forall X \in \text{Loc}. \forall \sigma \in \text{Store}. \forall n \in \mathbb{Z}.\ \langle X, \sigma \rangle \Downarrow_{\text{AExp}} n \Rightarrow n = \sigma(X)$;
2. $\forall n_1, n_2 \in \mathbb{Z}. \forall \sigma \in \text{Store}.\ \langle n_1, \sigma \rangle \Downarrow_{\text{AExp}} n_2 \Rightarrow n_2 = n_1$;
3. $\forall a_1, a_2 \in \text{AExp}. \forall \sigma \in \text{Store}. \forall n \in \mathbb{Z}.\ \langle a_1 + a2, \sigma \rangle \Downarrow_{\text{AExp}} n \Rightarrow \exists n_1, n_2 \in \mathbb{Z}.\ \langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1 \wedge \langle a2, \sigma \rangle \Downarrow_{\text{AExp}} n_2 \wedge n = n_1 + n_2$;
4. $\forall a_1, a_2 \in \text{AExp}. \forall \sigma \in \text{Store}. \forall n \in \mathbb{Z}.\ \langle a_1 - a_2, \sigma \rangle \Downarrow_{\text{AExp}} n \Rightarrow \exists n_1, n_2 \in \mathbb{Z}.\ \langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1 \wedge \langle a2, \sigma \rangle \Downarrow_{\text{AExp}} n_2 \wedge n = n_1 - n_2$;
5. $\forall a_1, a_2 \in \text{AExp}. \forall \sigma \in \text{Store}. \forall n \in \mathbb{Z}.\ \langle a_1 * a_2, \sigma \rangle \Downarrow_{\text{AExp}} n \Rightarrow \exists n_1, n_2 \in \mathbb{Z}.\ \langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1 \wedge \langle a2, \sigma \rangle \Downarrow_{\text{AExp}} n_2 \wedge n = n_1 * n_2$;

Proposition 2 (Principle of Derivation Induction for $\langle a, \sigma \rangle \Downarrow_{\text{AExp}} n$)**.**
Let Φ be a predicate on derivations $\mathcal{D} \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{AExp}}}]$. Then $\Phi(\mathcal{D})$ holds for all derivations $\mathcal{D} \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{AExp}}}]$ if:

1. $\forall n \in \mathbb{Z}. \forall \sigma \in \text{Store}.\ \Phi\left(\dfrac{}{\langle n, \sigma \rangle \Downarrow_{\text{AExp}} n}\ \textit{(enum)}\right)$;
2. $\forall X \in \text{Loc}. \forall \sigma \in \text{Store}.\ \Phi\left(\dfrac{}{\langle X, \sigma \rangle \Downarrow_{\text{AExp}} \sigma(X)}\ \textit{(eloc)}\right)$;
3. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall \sigma \in \text{Store}. \forall \mathcal{D}_1, \mathcal{D}_2 \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{AExp}}}]$.
$\mathcal{D}_1 :: \langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1 \wedge \mathcal{D}_2 :: \langle a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_2 \wedge \Phi(\mathcal{D}_1) \wedge \Phi(\mathcal{D}_2) \Rightarrow$
$\Phi\left(\dfrac{\overset{\mathcal{D}_1}{\langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1} \quad \overset{\mathcal{D}_2}{\langle a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_2}}{\langle a_1 + a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_1 + n_2}\ \textit{(eplus)}\right)$;
4. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall \sigma \in \text{Store}. \forall \mathcal{D}_1, \mathcal{D}_2 \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{AExp}}}]$.
$\mathcal{D}_1 :: \langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1 \wedge \mathcal{D}_2 :: \langle a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_2 \wedge \Phi(\mathcal{D}_1) \wedge \Phi(\mathcal{D}_2) \Rightarrow$
$\Phi\left(\dfrac{\overset{\mathcal{D}_1}{\langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1} \quad \overset{\mathcal{D}_2}{\langle a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_2}}{\langle a_1 \text{-} a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_1 - n_2}\ \textit{(eminus)}\right)$;
5. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall \sigma \in \text{Store}. \forall \mathcal{D}_1, \mathcal{D}_2 \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{AExp}}}]$.
$\mathcal{D}_1 :: \langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1 \wedge \mathcal{D}_2 :: \langle a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_2 \wedge \Phi(\mathcal{D}_1) \wedge \Phi(\mathcal{D}_2) \Rightarrow$
$\Phi\left(\dfrac{\overset{\mathcal{D}_1}{\langle a_1, \sigma \rangle \Downarrow_{\text{AExp}} n_1} \quad \overset{\mathcal{D}_2}{\langle a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_2}}{\langle a_1 * a_2, \sigma \rangle \Downarrow_{\text{AExp}} n_1 * n_2}\ \textit{(etimes)}\right)$.

Proposition 3 (Principle of Rule Induction for $\langle a, \sigma \rangle \Downarrow_{\text{AExp}} n$)**.**
Let Φ be a predicate on $\langle \langle a, \sigma \rangle, n \rangle \in \text{ACfg} \times \text{Store}$. Then $\Phi(\langle a, \sigma \rangle, n)$ holds for all $\langle a, \sigma \rangle \Downarrow_{\text{AExp}} n$ if:

1. $\forall n \in \mathbb{Z}. \forall \sigma \in \text{Store}.\ \Phi(\langle n, \sigma \rangle, n)$;
2. $\forall X \in \text{Loc}. \forall \sigma \in \text{Store}.\ \Phi(\langle X, \sigma \rangle, \sigma(X))$;
3. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall \sigma \in \text{Store}.\ \Phi(\langle a_1, \sigma \rangle, n_1) \wedge \Phi(\langle a_2, \sigma \rangle, n_2) \Rightarrow \Phi(\langle a_1 + a_2, \sigma \rangle, n_1 + n_2)$;
4. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall \sigma \in \text{Store}.\ \Phi(\langle a_1, \sigma \rangle, n_1) \wedge \Phi(\langle a_2, \sigma \rangle, n_2) \Rightarrow \Phi(\langle a_1 \text{-} a_2, \sigma \rangle, n_1 - n_2)$;
5. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall \sigma \in \text{Store}.\ \Phi(\langle a_1, \sigma \rangle, n_1) \wedge \Phi(\langle a_2, \sigma \rangle, n_2) \Rightarrow \Phi(\langle a_1 * a_2, \sigma \rangle, n_1 * n_2)$.

Proposition 4 (Principle of Derivation Induction for $\langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv$).
Let Φ be a predicate on derivations $\mathcal{D} \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{BExp}}}]$. Then $\Phi(\mathcal{D})$ holds for all derivations $\mathcal{D} \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{BExp}}}]$ if:

1. $\forall \sigma \in \text{Store}.\ \Phi\left(\dfrac{}{\langle \mathsf{true}, \sigma\rangle \Downarrow_{\text{BExp}} \mathsf{true}}\ (\textit{etrue})\right)$;

2. $\forall \sigma \in \text{Store}.\ \Phi\left(\dfrac{}{\langle \mathsf{false}, \sigma\rangle \Downarrow_{\text{BExp}} \mathsf{false}}\ (\textit{efalse})\right)$;

3. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall bv \in \text{Bool}. \forall \sigma \in \text{Store}. (n_1 = n_2 \Rightarrow bv = \mathsf{true}) \wedge$
$(n_1 \neq n_2 \Rightarrow bv = \mathsf{false}) \wedge \langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \wedge \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2 \Rightarrow \Phi\left(\dfrac{}{\langle a_1 = a_2, \sigma\rangle \Downarrow_{\text{BExp}} bv}\ (\textit{eeq})\right)$;

4. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall bv \in \text{Bool}. \forall \sigma \in \text{Store}. (n_1 \leq n_2 \Rightarrow bv = \mathsf{true}) \wedge$
$(n_1 > n_2 \Rightarrow bv = \mathsf{false}) \wedge \langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \wedge \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2 \Rightarrow \Phi\left(\dfrac{}{\langle a_1 \leq a_2, \sigma\rangle \Downarrow_{\text{BExp}} bv}\ (\textit{eleq})\right)$;

5. $\forall b \in \text{BExp}. \forall bv_1, bv_2 \in \text{Bool}. \forall \sigma \in \text{Store}. \forall \mathcal{D} \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{BExp}}}]. (bv_1 = \mathsf{false} \Rightarrow bv_2 = \mathsf{true}) \wedge$
$(bv_1 = \mathsf{true} \Rightarrow bv_2 = \mathsf{false}) \wedge \mathcal{D} :: \langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv_1 \wedge \Phi(\mathcal{D}) \Rightarrow \Phi\left(\dfrac{\begin{matrix}\mathcal{D}\\ \langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv_1\end{matrix}}{\langle \neg b, \sigma\rangle \Downarrow_{\text{BExp}} bv_2}\ (\textit{enot})\right)$;

6. $\forall b_1, b_2 \in \text{BExp}. \forall bv_1, bv_2, bv_3 \in \text{Bool}. \forall \sigma \in \text{Store}. \forall \mathcal{D}_1, \mathcal{D}_2, \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{BExp}}}].$
$(bv_1 = \mathsf{true} \wedge bv_2 = \mathsf{true} \Rightarrow bv_3 = \mathsf{true}) \wedge (\neg(bv_1 = \mathsf{true} \wedge bv_2 = \mathsf{true}) \Rightarrow bv_3 = \mathsf{true}) \wedge$
$\mathcal{D}_1 :: \langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv_1 \wedge \mathcal{D}_2 :: \langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv_1 \wedge \Phi(\mathcal{D}_1) \wedge \Phi(\mathcal{D}_2) \Rightarrow$
$\Phi\left(\dfrac{\begin{matrix}\mathcal{D}_1\\ \langle b_1, \sigma\rangle \Downarrow_{\text{BExp}} bv_1\end{matrix}\begin{matrix}\mathcal{D}_2\\ \langle b_2, \sigma\rangle \Downarrow_{\text{BExp}} bv_2\end{matrix}}{\langle b_1 \wedge b_2, \sigma\rangle \Downarrow_{\text{BExp}} bv_3}\ (\textit{eand})\right)$;

7. $\forall b_1, b_2 \in \text{BExp}. \forall bv_1, bv_2, bv_3 \in \text{Bool}. \forall \sigma \in \text{Store}. \forall \mathcal{D}_1, \mathcal{D}_2, \in \text{Deriv}[\mathcal{R}_{\Downarrow_{\text{BExp}}}].$
$(bv_1 = \mathsf{true} \vee bv_2 = \mathsf{true} \Rightarrow bv_3 = \mathsf{true}) \wedge (\neg(bv_1 = \mathsf{true} \vee bv_2 = \mathsf{true}) \Rightarrow bv_3 = \mathsf{true}) \wedge$
$\mathcal{D}_1 :: \langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv_1 \wedge \mathcal{D}_2 :: \langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv_1 \wedge \Phi(\mathcal{D}_1) \wedge \Phi(\mathcal{D}_2) \Rightarrow$
$\Phi\left(\dfrac{\begin{matrix}\mathcal{D}_1\\ \langle b_1, \sigma\rangle \Downarrow_{\text{BExp}} bv_1\end{matrix}\begin{matrix}\mathcal{D}_2\\ \langle b_2, \sigma\rangle \Downarrow_{\text{BExp}} bv_2\end{matrix}}{\langle b_1 \vee b_2, \sigma\rangle \Downarrow_{\text{BExp}} bv_3}\ (\textit{eand})\right)$.

Proposition 5 (Principle of Rule Induction for $\langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv$).
Let Φ be a predicate on $\langle\langle b, \sigma\rangle, bv\rangle \in \text{BCfg} \times \text{Store}$. Then $\Phi(\langle b, \sigma\rangle, bv)$ holds for all $\langle b, \sigma\rangle \Downarrow_{\text{BExp}} bv$ if:

1. $\forall \sigma \in \text{Store}. \Phi(\langle \mathsf{true}, \sigma\rangle, \mathsf{true})$;

2. $\forall \sigma \in \text{Store}. \Phi(\langle \mathsf{false}, \sigma\rangle, \mathsf{false})$;

3. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall bv \in \text{Bool}. \forall \sigma \in \text{Store}. (n_1 = n_2 \Rightarrow bv = \mathsf{true}) \wedge$
$(n_1 \neq n_2 \Rightarrow bv = \mathsf{false}) \wedge \langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \wedge \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2 \Rightarrow \Phi(\langle a_1 = a_2, \sigma\rangle, bv)$;

4. $\forall a_1, a_2 \in \text{AExp}. \forall n_1, n_2 \in \mathbb{Z}. \forall bv \in \text{Bool}. \forall \sigma \in \text{Store}. (n_1 \leq n_2 \Rightarrow bv = \mathsf{true}) \wedge$
$(n_1 > n_2 \Rightarrow bv = \mathsf{false}) \wedge \langle a_1, \sigma\rangle \Downarrow_{\text{AExp}} n_1 \wedge \langle a_2, \sigma\rangle \Downarrow_{\text{AExp}} n_2 \Rightarrow \Phi(\langle a_1 \leq a_2, \sigma\rangle, bv)$;

5. $\forall b \in \text{BExp}. \forall bv_1, bv_2 \in \text{Bool}. \forall \sigma \in \text{Store}. (bv_1 = \mathsf{false} \Rightarrow bv_2 = \mathsf{true}) \wedge$
$(bv_1 = \mathsf{true} \Rightarrow bv_2 = \mathsf{false}) \wedge \Phi(\langle b, \sigma\rangle, bv_1) \Rightarrow \Phi(\langle \neg b, \sigma\rangle, bv_2)$;

6. $\forall b_1, b_2 \in \text{BExp}. \forall bv_1, bv_2, bv_3 \in \text{Bool}. \forall \sigma \in \text{Store}. (bv_1 = \mathsf{true} \wedge bv_2 = \mathsf{true} \Rightarrow bv_3 = \mathsf{true}) \wedge$
$(\neg(bv_1 = \mathsf{true} \wedge bv_2 = \mathsf{true}) \Rightarrow bv_3 = \mathsf{true}) \wedge$
$\Phi(\langle b, \sigma\rangle, bv_1) \wedge \Phi(\langle b, \sigma\rangle, bv_1) \Rightarrow \Phi(\langle b_1 \wedge b_2, \sigma\rangle, bv_3)$;

7. $\forall b_1, b_2 \in \text{BExp}. \forall bv_1, bv_2, bv_3 \in \text{Bool}. \forall \sigma \in \text{Store}. (bv_1 = \mathsf{true} \vee bv_2 = \mathsf{true} \Rightarrow bv_3 = \mathsf{true}) \wedge$
$(\neg(bv_1 = \mathsf{true} \vee bv_2 = \mathsf{true}) \Rightarrow bv_3 = \mathsf{true}) \wedge$
$\Phi(\langle b, \sigma\rangle, bv_1) \wedge \Phi(\langle b, \sigma\rangle, bv_1) \Rightarrow \Phi(\langle b_1 \vee b_2, \sigma\rangle, bv_3)$.

Proposition 6 (Principle of Derivation Induction for $\langle c, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'$).
Let Φ be a predicate on derivations $\mathcal{D} \in \mathrm{DERIV}[\mathcal{R}_{\Downarrow_{\mathrm{COM}}}]$. Then $\Phi(\mathcal{D})$ holds for all derivations $\mathcal{D} \in \mathrm{DERIV}[\mathcal{R}_{\Downarrow_{\mathrm{COM}}}]$ if:

1. $\forall \sigma \in \mathrm{STORE}.\ \Phi\left(\dfrac{}{\langle \textsf{skip}, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma}\ \textit{(eskip)}\right)$;
2. $\forall a \in \mathrm{AEXP}. \forall n \in \mathbb{Z}. \forall X \in \mathrm{LOC}. \forall \sigma \in \mathrm{STORE}.\ \langle a, \sigma\rangle \Downarrow_{\mathrm{AEXP}} n \Rightarrow$
   $\Phi\left(\dfrac{}{\langle X := a, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma[X \mapsto n]}\ \textit{(eassign)}\right)$;
3. $\forall c_1, c_2 \in \mathrm{COM}. \forall \sigma, \sigma', \sigma'' \in \mathrm{STORE}. \forall \mathcal{D}_1, \mathcal{D}_2 \in \mathrm{DERIV}.\ D_1 :: \langle c_1, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma' \wedge D_2 :: \langle c_2, \sigma'\rangle \Downarrow_{\mathrm{COM}} \sigma'' \wedge$
   $\Phi(D_1) \wedge \Phi(D_2) \Rightarrow \Phi\left(\dfrac{\begin{matrix}\mathcal{D}_1 \\ \langle c_1, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'\end{matrix} \quad \begin{matrix}\mathcal{D}_2 \\ \langle c_2, \sigma'\rangle \Downarrow_{\mathrm{COM}} \sigma''\end{matrix}}{\langle c_1; c_2, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma''}\ \textit{(eseq)}\right)$;
4. $\forall b \in \mathrm{BEXP}. \forall c_1, c_2 \in \mathrm{COM}. \forall \sigma, \sigma' \in \mathrm{STORE}. \forall \mathcal{D}_1 \in \mathrm{DERIV}.\ \langle b, \sigma\rangle \Downarrow_{\mathrm{BEXP}} \textsf{true} \wedge \mathcal{D}_1 :: \langle c_1, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma' \wedge$
   $\Phi(\mathcal{D}_1) \Rightarrow \Phi\left(\dfrac{\begin{matrix}\mathcal{D}_1 \\ \langle c_1, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'\end{matrix}}{\langle \textsf{if}\ b\ \textsf{then}\ c_1\ \textsf{else}\ c_2, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'}\ \textit{(eif-t)}\right)$;
5. $\forall b \in \mathrm{BEXP}. \forall c_1, c_2 \in \mathrm{COM}. \forall \sigma, \sigma' \in \mathrm{STORE}. \forall \mathcal{D}_2 \in \mathrm{DERIV}.\ \langle b, \sigma\rangle \Downarrow_{\mathrm{BEXP}} \textsf{false} \wedge \mathcal{D}_2 :: \langle c_2, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma' \wedge$
   $\Phi(\mathcal{D}_2) \Rightarrow \Phi\left(\dfrac{\begin{matrix}\mathcal{D}_2 \\ \langle c_2, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'\end{matrix}}{\langle \textsf{if}\ b\ \textsf{then}\ c_1\ \textsf{else}\ c_2, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'}\ \textit{(eif-f)}\right)$;
6. $\forall b \in \mathrm{BEXP}. \forall c \in \mathrm{COM}. \forall \sigma \in \mathrm{STORE}.\ \langle b, \sigma\rangle \Downarrow_{\mathrm{BEXP}} \textsf{false} \Rightarrow \Phi\left(\dfrac{}{\langle \textsf{while}\ b\ \textsf{do}\ c, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma}\ \textit{(ewhile-f)}\right)$;
7. $\forall b \in \mathrm{BEXP}. \forall c \in \mathrm{COM}. \forall \sigma, \sigma', \sigma'' \in \mathrm{STORE}. \forall \mathcal{D}_1, \mathcal{D}_2 \in \mathrm{DERIV}.$
   $\langle b, \sigma\rangle \Downarrow_{\mathrm{BEXP}} \textsf{true} \wedge \mathcal{D}_1 :: \langle c_1, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma' \wedge \mathcal{D}_2 :: \langle \textsf{while}\ b\ \textsf{do}\ c, \sigma'\rangle \Downarrow_{\mathrm{COM}} \sigma'' \wedge$
   $\Phi(\mathcal{D}_1) \wedge \Phi(\mathcal{D}_2) \Rightarrow \Phi\left(\dfrac{\begin{matrix}\mathcal{D}_1 \\ \langle c, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'\end{matrix} \quad \begin{matrix}\mathcal{D}_2 \\ \langle \textsf{while}\ b\ \textsf{do}\ c, \sigma'\rangle \Downarrow_{\mathrm{COM}} \sigma''\end{matrix}}{\langle \textsf{while}\ b\ \textsf{do}\ c, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma''}\ \textit{(ewhile-t)}\right)$.

Proposition 7 (Principle of Rule Induction for $\langle c, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'$).
Let Φ be a predicate on $\langle \langle c, \sigma\rangle, \sigma'\rangle \in \mathrm{CCFG} \times \mathrm{STORE}$. Then $\Phi(\langle c, \sigma\rangle, \sigma')$ holds for all $\langle c, \sigma\rangle \Downarrow_{\mathrm{COM}} \sigma'$ if:

1. $\forall \sigma \in \mathrm{STORE}.\ \Phi(\langle \textsf{skip}, \sigma\rangle, \sigma)$;
2. $\forall a \in \mathrm{AEXP}. \forall n \in \mathbb{Z}. \forall X \in \mathrm{LOC}. \forall \sigma \in \mathrm{STORE}.\ \langle a, \sigma\rangle \Downarrow_{\mathrm{AEXP}} n \Rightarrow \Phi(\langle X := a, \sigma\rangle, \sigma[X \mapsto n])$;
3. $\forall c_1, c_2 \in \mathrm{COM}. \forall \sigma, \sigma', \sigma'' \in \mathrm{STORE}.\ \Phi(\langle c_1, \sigma\rangle, \sigma') \wedge \Phi(\langle c_2, \sigma'\rangle, \sigma'') \Rightarrow \Phi(\langle c_1; c_2, \sigma\rangle, \sigma'')$;
4. $\forall b \in \mathrm{BEXP}. \forall c_1, c_2 \in \mathrm{COM}. \forall \sigma, \sigma' \in \mathrm{STORE}.\ \langle b, \sigma\rangle \Downarrow_{\mathrm{BEXP}} \textsf{true} \wedge \Phi(\langle c_1, \sigma\rangle, \sigma') \Rightarrow$
   $\Phi(\langle \textsf{if}\ b\ \textsf{then}\ c_1\ \textsf{else}\ c_2, \sigma\rangle, \sigma')$;
5. $\forall b \in \mathrm{BEXP}. \forall c_1, c_2 \in \mathrm{COM}. \forall \sigma, \sigma' \in \mathrm{STORE}.\ \langle b, \sigma\rangle \Downarrow_{\mathrm{BEXP}} \textsf{false} \wedge \Phi(\langle c_2, \sigma\rangle, \sigma') \Rightarrow$
   $\Phi(\langle \textsf{if}\ b\ \textsf{then}\ c_1\ \textsf{else}\ c_2, \sigma\rangle, \sigma')$;
6. $\forall b \in \mathrm{BEXP}. \forall c \in \mathrm{COM}. \forall \sigma \in \mathrm{STORE}.\ \langle b, \sigma\rangle \Downarrow_{\mathrm{BEXP}} \textsf{false} \Rightarrow \Phi(\langle \textsf{while}\ b\ \textsf{do}\ c, \sigma\rangle, \sigma)$;
7. $\forall b \in \mathrm{BEXP}. \forall c \in \mathrm{COM}. \forall \sigma, \sigma', \sigma'' \in \mathrm{STORE}.\ \langle b, \sigma\rangle \Downarrow_{\mathrm{BEXP}} \textsf{true} \wedge$
   $\Phi(\langle c_1, \sigma\rangle, \sigma') \wedge \Phi(\langle \textsf{while}\ b\ \textsf{do}\ c, \sigma'\rangle, \sigma'') \Rightarrow \Phi(\langle \textsf{while}\ b\ \textsf{do}\ c, \sigma\rangle, \sigma'')$.