5 Reference

For your reference, here is the syntax, large-step operational semantics, and typing rules of SIMPL that should be the basis of your type-checker implementation.

5.1 Syntax of SIMPL

commands	$c ::= \texttt{skip} \mid c_1; c_2 \mid v\texttt{:=}e \mid \texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2$
	$\mid \texttt{while}\ e\ \texttt{do}\ c \mid \texttt{int}\ v \mid \texttt{bool}\ v$
expressions	$e ::= n \mid \texttt{true} \mid \texttt{false} \mid v \mid e_1\ aop\ e_2 \mid e_1\ bop\ e_2 \mid e_1\texttt{<=}e_2 \mid \texttt{!}e$
arithmetic operators	$aop ::= \texttt{+} \mid \texttt{-} \mid \texttt{*}$
boolean operators	$bop ::= \texttt{\&\&} \mid \texttt{||}$
variable names	v
integer constants	n
types	$\tau ::= int \mid bool$
stores	$\sigma : v \rightharpoonup \mathbb{Z} \cup \{T, F\}$
typing contexts	$\Gamma : v \rightharpoonup \tau \times \{T, F\}$

5.2 Large-step Semantics of SIMPL

5.2.1 Commands

$$\langle \texttt{skip}, \sigma\rangle \Downarrow \sigma \quad (1)$$

$$\langle \texttt{int}\ v, \sigma\rangle \Downarrow \sigma \quad (2)$$

$$\langle \texttt{bool}\ v, \sigma\rangle \Downarrow \sigma \quad (3)$$

$$\frac{\langle c_1, \sigma\rangle \Downarrow \sigma_2 \qquad \langle c_2, \sigma_2\rangle \Downarrow \sigma'}{\langle c_1; c_2, \sigma\rangle \Downarrow \sigma'} \quad (4)$$

$$\frac{\langle e, \sigma\rangle \Downarrow n}{\langle v\texttt{:=}e, \sigma\rangle \Downarrow \sigma[v \mapsto n]} \quad (5)$$

$$\frac{\langle e, \sigma\rangle \Downarrow p}{\langle v\texttt{:=}e, \sigma\rangle \Downarrow \sigma[v \mapsto p]} \quad (6)$$

$$\frac{\langle e, \sigma\rangle \Downarrow T \qquad \langle c_1, \sigma\rangle \Downarrow \sigma'}{\langle \texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2, \sigma\rangle \Downarrow \sigma'} \quad (7)$$

$$\frac{\langle e, \sigma\rangle \Downarrow F \qquad \langle c_2, \sigma\rangle \Downarrow \sigma'}{\langle \texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2, \sigma\rangle \Downarrow \sigma'} \quad (8)$$

$$\frac{\langle \texttt{if}\ e\ \texttt{then}\ (c; \texttt{while}\ e\ \texttt{do}\ c)\ \texttt{else skip}, \sigma\rangle \Downarrow \sigma'}{\langle \texttt{while}\ e\ \texttt{do}\ c, \sigma\rangle \Downarrow \sigma'} \quad (9)$$

5.2.2 Expressions

$$\langle n, \sigma\rangle \Downarrow n \tag{10}$$

$$\langle \texttt{true}, \sigma\rangle \Downarrow T \tag{11}$$

$$\langle \texttt{false}, \sigma\rangle \Downarrow F \tag{12}$$

$$\langle v, \sigma\rangle \Downarrow \sigma(v) \tag{13}$$

$$\frac{\langle e_1, \sigma\rangle \Downarrow n_1 \qquad \langle e_2, \sigma\rangle \Downarrow n_2}{\langle e_1 \texttt{<=} e_2, \sigma\rangle \Downarrow n_1 \leq n_2} \tag{14}$$

$$\frac{\langle e_1, \sigma\rangle \Downarrow p \qquad \langle e_2, \sigma\rangle \Downarrow q}{\langle e_1 \texttt{ \&\& } e_2, \sigma\rangle \Downarrow p \wedge q} \tag{15}$$

$$\frac{\langle e_1, \sigma\rangle \Downarrow p \qquad \langle e_2, \sigma\rangle \Downarrow q}{\langle e_1 \texttt{ || } e_2, \sigma\rangle \Downarrow p \vee q} \tag{16}$$

$$\frac{\langle e, \sigma\rangle \Downarrow p}{\langle \texttt{!}e, \sigma\rangle \Downarrow \neg p} \tag{17}$$

$$\frac{\langle e_1, \sigma\rangle \Downarrow n_1 \qquad \langle e_2, \sigma\rangle \Downarrow n_2}{\langle e_1 \texttt{ + } e_2, \sigma\rangle \Downarrow n_1 + n_2} \tag{18}$$

$$\frac{\langle e_1, \sigma\rangle \Downarrow n_1 \qquad \langle e_2, \sigma\rangle \Downarrow n_2}{\langle e_1 \texttt{ - } e_2, \sigma\rangle \Downarrow n_1 - n_2} \tag{19}$$

$$\frac{\langle e_1, \sigma\rangle \Downarrow n_1 \qquad \langle e_2, \sigma\rangle \Downarrow n_2}{\langle e_1 \texttt{ * } e_2, \sigma\rangle \Downarrow n_1 n_2} \tag{20}$$

5.3 Typing Rules for SIMPL

5.3.1 Commands

$$\Gamma \vdash \texttt{skip} : \Gamma \tag{21}$$

$$\frac{v \notin \Gamma^{\leftarrow}}{\Gamma \vdash \texttt{int } v : \Gamma[v \mapsto (int, F)]} \tag{22}$$

$$\frac{v \notin \Gamma^{\leftarrow}}{\Gamma \vdash \texttt{bool } v : \Gamma[v \mapsto (bool, F)]} \tag{23}$$

$$\frac{\Gamma \vdash c_1 : \Gamma_2 \qquad \Gamma_2 \vdash c_2 : \Gamma'}{\Gamma \vdash c_1; c_2 : \Gamma'} \tag{24}$$

$$\frac{\Gamma \vdash e : \tau \qquad \Gamma(v) = (\tau, p)}{\Gamma \vdash v \texttt{:=} e : \Gamma[v \mapsto (\tau, T)]} \tag{25}$$

$$\frac{\Gamma \vdash e : bool \qquad \Gamma \vdash c_1 : \Gamma_1 \qquad \Gamma \vdash c_2 : \Gamma_2}{\Gamma \vdash \texttt{if } e \texttt{ then } c_1 \texttt{ else } c_2 : \Gamma} \tag{26}$$

$$\frac{\Gamma \vdash e : bool \qquad \Gamma \vdash c : \Gamma_1}{\Gamma \vdash \texttt{while } e \texttt{ do } c : \Gamma} \tag{27}$$

5.3.2 Expressions

$$\Gamma \vdash n : int \tag{28}$$

$$\Gamma \vdash \texttt{true} : bool \tag{29}$$

$$\Gamma \vdash \texttt{false} : bool \tag{30}$$

$$\frac{\Gamma(v) = (\tau, T)}{\Gamma \vdash v : \tau} \tag{31}$$

$$\frac{\Gamma \vdash e_1 : int \qquad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 \ aop \ e_2 : int} \tag{32}$$

$$\frac{\Gamma \vdash e_1 : bool \qquad \Gamma \vdash e_2 : bool}{\Gamma \vdash e_1 \ bop \ e_2 : bool} \tag{33}$$

$$\frac{\Gamma \vdash e_1 : int \qquad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 \texttt{<=} e_2 : bool} \tag{34}$$

$$\frac{\Gamma \vdash e : bool}{\Gamma \vdash \texttt{!}e : bool} \tag{35}$$