

# Rules for Translating While to SimpleIR

Paul Gazzillo, COP-5621 Spring 2025

February 24, 2025

## Contents

<b>1</b>	<b>Translation Rules</b>	<b>1</b>
1.1	Notation . . . . .	1
1.2	Statements . . . . .	1
1.3	Expressions . . . . .	3
1.3.1	Boolean Expressions . . . . .	3
1.3.2	Relational Expressions . . . . .	4
1.3.3	Arithmetic Expressions . . . . .	5

## 1 Translation Rules

### 1.1 Notation

All language interpretation rules have the following format:

$$\frac{\text{evaluation of nested constructs (if necessary)}}{\langle \text{while construct to evaluate} \rangle \Rightarrow \text{result of the evaluation}}$$

Angle brackets  $\langle \rangle$  contain the symbols of a  $\langle$ while grammar construct to evaluate $\rangle$ . For example,  $\langle \text{if } b \text{ then } s_1 \text{ else } s_2 \rangle$ , refers to any if-then-else statement, where  $b$  is the placeholder (nonterminal) for Boolean expressions and  $s_1, s_2$  for statements. The arrow  $\Rightarrow$  just means “evaluates to”.

In summary, “ $\langle$ while construct to evaluate $\rangle \Rightarrow$  result of the evaluation” means “while construct evaluates to the result of the evaluation”.

### 1.2 Statements

The result of an evaluation of a statement is a list of SimpleIR instructions. We separate the list of statements in to a separate symbol for readable, e.g., GENCOMPOUND below. The goto targets are represented by  $p$  variables, and their computation is listed at the bottom of each rule that translates to goto instructions.

$$\frac{\langle s_1 \rangle \Rightarrow I_1 \quad \langle s_2 \rangle \Rightarrow I_2 \quad \cdots \quad \langle s_n \rangle \Rightarrow I_n}{\langle \text{begin } s_1; s_2; \cdots s_n \text{ end} \rangle \Rightarrow \text{GENCOMPOUND}} \quad \text{Compound}$$

$$\begin{aligned} \text{GENCOMPOUND} \equiv \\ & I_1 \\ & I_2 \\ & \cdots \\ & I_n \end{aligned}$$

$$\frac{\langle a \rangle \Rightarrow (t_a, A)}{\langle x := a \rangle \Rightarrow \text{GENASSIGNMENT}} \quad \text{Assignment}$$

$$\begin{aligned} \text{GENASSIGNMENT} \equiv \\ & A \\ & x := t_a \end{aligned}$$

$$\frac{\langle b \rangle \Rightarrow (t_b, B) \quad \langle s_1 \rangle \Rightarrow S_1 \quad \langle s_2 \rangle \Rightarrow S_2}{\langle \text{if } b \text{ then } s_1 \text{ else } s_2 \rangle \Rightarrow \text{GENIF}} \quad \text{If}$$

$$\begin{aligned} \text{GENIF} \equiv \\ & B \\ & \text{if } t_b = 0 \text{ goto } l_{\text{else}} \\ & S_1 \\ & \text{goto } l_{\text{endif}} \\ & l_{\text{else}} : S_2 \\ & l_{\text{endif}} : \end{aligned}$$

$$\frac{\langle b \rangle \Rightarrow (t_b, B) \quad \langle s \rangle \Rightarrow S}{\langle \text{while } b \text{ do } s \rangle \Rightarrow \text{GENWHILE}} \quad \text{While}$$

GENWHILE  $\equiv$   
 $l_{\text{head}} : B$   
     if  $t_b = 0$  goto  $l_{\text{end}}$   
      $S$   
     goto  $l_{\text{head}}$   
 $l_{\text{end}} :$

### 1.3 Expressions

Instead of outputting only the SimpleIR, expressions also return the name of a temporary variable that will hold the value of the expression. For instance, a Num expression results in a SimpleIR instruction that sets a new temporary variable to a constant, e.g.,  $t_8 := 5$ , so evaluation results in a tuple containing both the temporary variable and the set of instructions, i.e.,  $(t_8, t_8 := 5)$ .

#### 1.3.1 Boolean Expressions

$$\frac{}{\langle \text{true} \rangle \Rightarrow (t, t := 1)} \quad \text{True}$$

$$\frac{}{\langle \text{false} \rangle \Rightarrow (t, t := 0)} \quad \text{False}$$

$$\frac{\langle b \rangle \Rightarrow (t_b, B)}{\langle b \rangle \Rightarrow (t, \text{GENNOT})} \quad \text{Not}$$

GENNOT  $\equiv$   
 $B$   
     if  $t_b = 0$  goto  $p_1$   
      $t = 0$   
     goto  $p_{\text{end}}$   
 $p_{\text{false}} : t = 1$   
 $p_{\text{end}} :$

$$\frac{\langle b_1 \rangle \Rightarrow (t_1, B_1) \quad \langle b_2 \rangle \Rightarrow (t_2, B_2)}{\langle b_1 \text{ and } b_2 \rangle \Rightarrow (t, \text{GENAND})} \text{ And}$$

GENAND  $\equiv$

```

    B1
    B2
    if t1 = 0 goto pfalse
    if t2 = 0 goto pfalse
    t = 1
    goto pend
pfalse : t = 0
pend :
```

### 1.3.2 Relational Expressions

$$\frac{\langle b_1 \rangle \Rightarrow (t_1, B_1) \quad \langle b_2 \rangle \Rightarrow (t_2, B_2)}{\langle b_1 \text{ and } b_2 \rangle \Rightarrow (t, \text{GENOR})} \text{ Or}$$

GENOR  $\equiv$

```

    B1
    B2
    if t1 = 0 goto pright
    t = 1
    goto pend
pright : if t2 = 0 goto pfalse
    t = 1
    goto pend
pfalse : t = 0
pend :
```

$$\frac{\langle a_1 \rangle \Rightarrow (t_1, A_1) \quad \langle a_2 \rangle \Rightarrow (t_2, A_2)}{\langle a_1 \text{op}_r a_2 \rangle \Rightarrow (t, \text{GENRELOP})} \text{ Equals}$$

GENRELOP  $\equiv$

$A_1$

$A_2$

**if**  $t_1 \text{op}_r t_2$  **goto**  $p_{\text{true}}$

$t = 0$

**goto**  $p_{\text{end}}$

$p_{\text{true}} : t = 1$

$p_{\text{end}} :$

### 1.3.3 Arithmetic Expressions

$$\frac{}{\langle n \rangle \Rightarrow (t, t := n)} \text{ Num}$$

$$\frac{}{\langle x \rangle \Rightarrow (t, t := x)} \text{ Var}$$

$$\frac{\langle a_1 \rangle \Rightarrow (t_1, A_1) \quad \langle a_2 \rangle \Rightarrow (t_2, A_2)}{\langle a_1 \text{op}_a a_2 \rangle \Rightarrow (t, \text{GENARITHMETIC})} \text{ Arithmetic}$$

GENARITHMETIC  $\equiv$

$A_1$

$A_2$

$t = t_1 \text{op}_a t_2$