Rules for Translating While to While3Addr

Paul Gazzillo, COP-5621 Spring 2024

January 25, 2024

Contents

1 Translation Rules

1.1 Notation

All language interpretation rules have the following format:

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

All expressions and statements are evaluated under a current program counter p. Angle brackets $\langle\ \rangle$ contain the symbols of a $\langle$while grammar construct to evaluate$\rangle$. For example, $\langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2 \rangle$, refers to any if-then-else statement, where b is the placeholder (nonterminal) for Boolean expressions and s_1, s_2 for statements. The turnstile $\vdash$ indicates a context, as in the evaluation of the if statement happens under the context p. In our semantics, the only context is the current program counter p, which will be used to compute the addresses for branches. The arrow $\Downarrow$ just means "evaluates to".

In summary, "$p \vdash \langle$while construct to evaluate$\rangle \Downarrow$ result of the evaluation" means "given the program counter p the while construct evaluates to the result of the evaluation".

1.2 Statements

The result of an evaluation of a statement is a list of While3Addr instructions. We separate the list of statements in to a separate symbol for readable, e.g.,

$\textsc{GenCompound}$ 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.

$$\frac{p_1 \vdash \langle s_1 \rangle \Downarrow I_1 \qquad p_2 \vdash \langle s_2 \rangle \Downarrow I_2 \qquad \cdots \qquad p_n \vdash \langle s_n \rangle \Downarrow I_n}{p \vdash \langle \texttt{begin}\ s_1;\ s_2;\ \cdots\ s_n\ \texttt{end} \rangle \Downarrow \textsc{GenCompound}}\ \text{Compound}$$

$$\begin{aligned} \textsc{GenCompound} \equiv & \\ & I_1 \\ & I_2 \\ & \cdots \\ & I_n \end{aligned}$$

$$p_1 = p, p_2 = p_1 + |I_1|, \cdots, p_n = p_{n-1} + |I_{n-1}|$$

$$\frac{p \vdash \langle a \rangle \Downarrow (t_a, A)}{p \vdash \langle x := a \rangle \Downarrow \textsc{GenAssignment}}\ \text{Assignment}$$

$$\begin{aligned} \textsc{GenAssignment} \equiv & \\ & A \\ & x := t_a \end{aligned}$$

$$\frac{p \vdash \langle b \rangle \Downarrow (t_b, B) \qquad p_{\text{if}} \vdash \langle s_1 \rangle \Downarrow S_1 \qquad p_{\text{else}} \vdash \langle s_2 \rangle \Downarrow S_2}{p \vdash \langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2 \rangle \Downarrow \textsc{GenIf}}\ \text{If}$$

$$\begin{array}{rl} \textsc{GenIf} \equiv & \\ & B \\ & \texttt{if}\ t_b = 0\ \texttt{goto}\ p_{\text{else}} \\ & S_1 \\ & \texttt{goto}\ p_{\text{endif}} \\ p_{\text{else}} : & S_2 \\ p_{\text{endif}} : & \end{array}$$

$$p_{\text{if}} = |B| + 1, p_{\text{else}} = p_{\text{if}} + |S_1| + 1, p_{\text{endif}} = p_{\text{else}} + |S_2| + 1$$

$$\frac{p \vdash \langle b \rangle \Downarrow (t_b, B) \qquad p_{\text{if}} \vdash \langle s \rangle \Downarrow S}{p \vdash \langle \texttt{while}\ b\ \texttt{do}\ s \rangle \Downarrow \textsc{GenWhile}}\ \text{While}$$

$$\begin{array}{rl} \textsc{GenWhile} \equiv & \\ p_{\text{head}} : & B \\ & \texttt{if}\ t_b = 0\ \texttt{goto}\ p_{\text{end}} \\ & S \\ & \texttt{goto}\ p_{\text{head}} \\ p_{\text{end}} : & \end{array}$$

$$p_{\text{head}} = p, p_{\text{body}} = p_{\text{head}} + |B| + 1, p_{\text{end}} = p_{\text{body}} + |S| + 1$$

1.3 Expressions

Instead of outputting only the While3Addr, 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 While3Addr instruction that sets a new temporary variable to a constant, e.g., $t := 5$, so evaluation results in a tuple containing both the temporary variable and the set of instructions, i.e., $(t, t := 5)$.

1.3.1 Boolean Expressions

$$\frac{}{p \vdash \langle\ \texttt{true}\ \rangle \Downarrow (t, t := 1)}\ \text{True}$$

$$\frac{}{p \vdash \langle\ \texttt{false}\ \rangle \Downarrow (t, t := 0)}\ \text{False}$$

$$\frac{p \vdash \langle b \rangle \Downarrow (t_b, B)}{p \vdash \langle b \rangle \Downarrow (t, \textsc{GenNot})}\ \text{Not}$$

```
GenNot ≡
                  $B$
                  if $t_1 = 0$ goto $p_1$
                  $t = 0$
                  goto $p_{\text{end}}$
      $p_{\text{false}}$ : $t = 1$
        $p_{\text{end}}$ :
```

$$p_{\text{false}} = p + |B| + 3, p_{\text{end}} = p_{\text{false}} + 1$$

$$\frac{p \vdash \langle b_1 \rangle \Downarrow (t_1, B_1) \qquad p_2 \vdash \langle b_2 \rangle \Downarrow (t_2, B_2)}{p \vdash \langle b_1 \ \texttt{and}\ b_2 \rangle \Downarrow (t, \textsc{GenAnd})}\ \text{And}$$

```
GenAnd ≡
                  $B_1$
                  $B_2$
                  if $t_1 = 0$ goto $p_{\text{false}}$
                  if $t_2 = 0$ goto $p_{\text{false}}$
                  $t = 1$
                  goto $p_{\text{end}}$
      $p_{\text{false}}$ : $t = 0$
        $p_{\text{end}}$ :
```

$$p_2 = p + |B_1|, p_{\text{false}} = p_2 + |B_2| + 3, p_{\text{end}} = p_{\text{false}} + 1$$

1.3.2 Relational Expressions

$$\frac{p \vdash \langle b_1 \rangle \Downarrow (t_1, B_1) \qquad p_2 \vdash \langle b_2 \rangle \Downarrow (t_2, B_2)}{p \vdash \langle b_1 \ \texttt{and} \ b_2 \rangle \Downarrow (t, \textsc{GenOr})} \ \text{Or}$$

$$\begin{array}{rl}
\textsc{GenOr} \equiv & \\
 & B_1 \\
 & B_2 \\
 & \texttt{if } t_1 = 0 \texttt{ goto } p_{\text{right}} \\
 & t = 1 \\
 & \texttt{goto } p_{\text{end}} \\
p_{\text{right}} : & \texttt{if } t_2 = 0 \texttt{ goto } p_{\text{false}} \\
 & t = 1 \\
 & \texttt{goto } p_{\text{end}} \\
p_{\text{false}} : & t = 0 \\
p_{\text{end}} : &
\end{array}$$

$$p_2 = p + |B_1|, p_{\text{right}} = p_2 + |B_2| + 4, p_{\text{false}} = p_{\text{right}} + 3, p_{\text{end}} = p_{\text{false}} + 1$$

$$\frac{p \vdash \langle a_1 \rangle \Downarrow (t_1, A_1) \qquad p_2 \vdash \langle a_2 \rangle \Downarrow (t_2, A_2)}{p \vdash \langle a_1 = a_2 \rangle \Downarrow (t, \textsc{GenEquals})} \ \text{Equals}$$

$$\begin{array}{rl}
\textsc{GenEquals} \equiv & \\
 & A_1 \\
 & A_2 \\
 & t_s = t_1 - t_2 \\
 & \texttt{if } t_s = 0 \texttt{ goto } p_{\text{true}} \\
 & t = 0 \\
 & \texttt{goto } p_{\text{end}} \\
p_{\text{true}} : & t = 1 \\
p_{\text{end}} : &
\end{array}$$

$$p_2 = p + |A_1|, p_{\text{true}} = p_2 + |A_2| + 4, p_{\text{end}} = p_{\text{true}} + 1$$

The While3Addr language only has two relational operators, `<` and `=`. In order to transform While programs, which supports `<`, `>`, `>=`, and `<=`, we can use the following equivalences to help us translate to While3Addr.

$$a < b \Rightarrow a - b < 0 \tag{1}$$

$$a > b \Rightarrow b < a \Rightarrow b - a < 0 \tag{2}$$

$$a \geq b \Rightarrow \neg(a < b) \Rightarrow \neg(a - b < 0) \tag{3}$$

$$a \leq b \Rightarrow \neg(a > b) \Rightarrow \neg(b < a) \Rightarrow \neg(b - a < 0) \tag{4}$$

Notice that there are two techniques used, i.e.,

1. Swap a and b to transform $<$ to $>$
2. Negate the result to swap $\leq$ to $>$ or $\geq$ to $<$

We can create a single generic code generator to support all four cases.

$$\begin{array}{rl} \textsc{GenRel}(t_a, t_b, v_{\text{true}}, v_{\text{false}}) \equiv & \\ & A_1 \\ & A_2 \\ & t_s = t_a - t_b \\ & \texttt{if}\ t_s < 0\ \texttt{goto}\ p_{\text{true}} \\ & t = v_{\text{false}} \\ & \ \texttt{goto}\ p_{\text{end}} \\ p_{\text{true}}: & t = v_{\text{true}} \\ p_{\text{end}}: & \end{array}$$

$$p_2 = p + |A_1|, p_{\text{true}} = p_2 + |A_2| + 4, p_{\text{end}} = p_{\text{true}} + 1$$

Then we can use GenRel to define translations for each of the four relational operators in the While language.

$$\frac{p \vdash \langle a_1 \rangle \Downarrow (t_1, A_1) \qquad p_2 \vdash \langle a_2 \rangle \Downarrow (t_2, A_2)}{p \vdash \langle a_1 < a_2 \rangle \Downarrow (t, \textsc{GenRel}(t_1, t_2, 1, 0))}\ \text{Less Than}$$

$$\frac{p \vdash \langle a_1 \rangle \Downarrow (t_1, A_1) \qquad p_2 \vdash \langle a_2 \rangle \Downarrow (t_2, A_2)}{p \vdash \langle a_1 > a_2 \rangle \Downarrow (t, \textsc{GenRel}(t_2, t_1, 1, 0))}\ \text{Greater Than}$$

$$\frac{p \vdash \langle a_1 \rangle \Downarrow (t_1, A_1) \qquad p_2 \vdash \langle a_2 \rangle \Downarrow (t_2, A_2)}{p \vdash \langle a_1 >= a_2 \rangle \Downarrow (t, \textsc{GenRel}(t_1, t_2, 0, 1))}\ \text{Greater Than or Equals}$$

$$\frac{p \vdash \langle a_1 \rangle \Downarrow (t_1, A_1) \qquad p_2 \vdash \langle a_2 \rangle \Downarrow (t_2, A_2)}{p \vdash \langle a_1 \Leftarrow a_2 \rangle \Downarrow (t, \textsc{GenRel}(t_2, t_1, 0, 1))} \text{ Less Than or Equals}$$

1.3.3 Arithmetic Expressions

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

$$\frac{}{p \vdash \langle x \rangle \Downarrow (x, \emptyset)} \text{ Var}$$

$$\frac{p \vdash \langle a_1 \rangle \Downarrow (t_1, A_1) \qquad p_2 \vdash \langle a_2 \rangle \Downarrow (t_2, A_2)}{p \vdash \langle a_1 \mathtt{op}_a a_2 \rangle \Downarrow (t, \textsc{GenArithmetic})} \text{ Arithmetic}$$

$$\begin{aligned} \textsc{GenArithmetic} \equiv \\ & A_1 \\ & A_2 \\ & t = t_1 \mathtt{op}_a t_2 \end{aligned}$$

$$p_2 = p + |A_1|$$