

CS3005D Compiler Design

Winter 2024

Lecture #26

Intermediate Code Generation - Arithmetic Expressions, Assignment

Saleena N
CSED NIT Calicut

March 2024

Generation of 3-address code - expressions

Production	Semantic Rules
$E \rightarrow E_1 + E_2$	$E.addr = newTemp()$ $E.code = E_1.code \parallel E_2.code \parallel$ $gen(E.addr \text{ ' = ' } E_1.addr \text{ ' + ' } E_2.addr)$
$E \rightarrow \mathbf{id}$	$E.addr = \mathbf{id}.entry$ $E.code = \text{' '}$

- $E.code$: 3-address code for E
- $E.addr$: the address¹ that will hold the value of E
- $\mathbf{id}.entry$: pointer to the symbol table entry for \mathbf{id} (denoted by the name of the \mathbf{id} for convenience)
- $newTemp()$: returns a distinct temporary name
- $gen()$: generates and returns a 3-address instruction, concatenating its arguments (quoted arguments as such, expressions replaced by value)

¹the address can be a name, a constant or a compiler-generated temporary

SDD for translating Assignment statement

Production	Semantic Rules
$S \rightarrow id = E$	$S.code = E.code \parallel gen(id.entry \text{ ' = ' } E.addr)$
$E \rightarrow E_1 + E_2$	$E.addr = newTemp()$ $E.code = E_1.code \parallel E_2.code \parallel$ $gen(E.addr \text{ ' = ' } E_1.addr \text{ ' + ' } E_2.addr)$
$E \rightarrow -E_1$	$E.addr = newTemp()$ $E.code = E_1.code \parallel$ $gen(E.addr \text{ ' = ' 'minus' } E_1.addr$
$E \rightarrow (E_1)$	$E.addr = E_1.addr$ $E.code = E_1.code$
$E \rightarrow id$	$E.addr = id.entry$ $E.code = \text{ ' ' }$

Exercise

1. Write the 3-address code generated for the following instructions, as per the previous SDD:

$$x = y$$

$$x = a + b$$

$$x = a + -(c + d)$$

2. Extend the previous SDD by adding production $E \rightarrow E * E$ with the required semantic rule. Write the 3-address code generated for $x = (a + b) * (c + d)$

References

References:

- Aho A.V., Lam M.S., Sethi R., and Ullman J.D. Compilers: Principles, Techniques, and Tools (ALSU). Pearson Education, 2007.

Further reading:

- ALSU Chapter 6 - 6.4.1