

Expressões e Instruções de Atribuição

George Darmiton da Cunha Cavalcanti
(gdcc@cin.ufpe.br)

Tópicos



- Introdução
- Expressões Aritméticas
- Operadores Sobrecarregados
- Conversões de Tipo
- Expressões Relacionais e Booleanas
- Avaliação Curto-Circuito
- Instruções de Atribuição
- Atribuição de Modo Misto

Introdução



- Expressões são o meio fundamental de especificar computações em uma linguagem de programação
- Para entender a avaliação de expressões é necessário estar familiarizado com as ordens de avaliação de operadores e de operandos
- A essência das linguagens de programação imperativas é dado pelo papel predominantes das instruções de atribuição, cujo objetivo é mudar o valor de uma variável



Expressões Aritméticas



- A avaliação automática de expressões aritméticas foi uma das principais metas das primeiras linguagens de programação
- Expressões aritméticas consistem de operadores, operandos, parênteses e chamadas a função



Expressões Aritméticas



- Questões de projeto
 - Regras de precedência de operadores
 - Regras de associatividade dos operadores
 - Ordem de avaliação dos operandos
 - Efeitos colaterais da avaliação dos operandos
 - Sobrecarga de operadores
 - Mesclagem de modos nas expressões

Expressões Aritméticas



- Operadores
 - Unário
 - Binário
 - Ternário



Ordem de avaliação dos operadores



- As regras de **precedência** para avaliação de expressões definem a ordem na qual operadores adjacentes de diferentes níveis de precedência são avaliados
- Níveis de precedência comuns
 - parênteses
 - operadores unários
 - ** (se a linguagem o suporta)
 - *, /
 - +, -

Regras de associatividade



- Regras de **associatividade** para avaliação de expressões definem em qual ordem operadores adjacentes de mesma precedência são avaliados
- Regras de associatividade comuns
 - Da esquerda para a direita, exceto **, o qual é da direita para a esquerda
- Em APL todos os operadores têm precedência iguais e são associativos da direita para a esquerda
- Regras de precedência e de associatividade podem ser substituídas pelo uso de parênteses



Expressões Condicionais



- Operador ternário ?
- C, C++ e Java

– Um exemplo:

```
average = (count == 0) ? 0 : sum/count
```

– if-then-else equivalente

```
if (count == 0)
    average = 0
else average = sum /count
```



Ordem de avaliação dos operandos



1. Variáveis

- São avaliadas buscando seus valores na memória

2. Constantes

- Algumas vezes são avaliadas da mesma forma;
- Outras vezes podem fazer parte da instrução em linguagem de máquina

3. Expressões entre parênteses

- Todos os operadores devem ser avaliados antes que seu valor possa ser usado como um operando

Ordem de avaliação dos operandos: Efeitos Colaterais



- Quando uma função altera um de seus parâmetros ou uma variável global
- Exemplo
 - Quando uma função é chamada em uma expressão e altera outro operando da expressão:

```
a = 10;
```

```
/* assume that fun changes its parameter */
```

```
b = a + fun(a);
```



Ordem de avaliação dos operandos: Efeitos Colaterais



- Duas soluções possíveis
 1. O projetista da linguagem poderia impedir que a avaliação da função afetasse o valor das expressões
 - **Vantagem:** funciona
 - **Desvantagem:** inflexibilidade
 2. Declarar, na definição da linguagem, que os operandos devem ser avaliados em uma ordem particular
 - **Desvantagem:** limita alguns procedimentos de otimização dos compiladores

Sobrecarga de Operadores



- Usar um operador para mais do que um propósito
- Alguns são comuns
 - + para `int` e para `float`
- Alguns representam problemas em potencial
 - Perda da capacidade de detectar erros
 - Omissão de um operador
 - Podem ser evitados pela introdução de novos símbolos
 - Exemplo: `div` para divisão de inteiros no Pascal

Conversões de Tipo



- Uma **conversão de estreitamento** transforma um valor para um tipo que não pode armazenar todos os valores do tipo original
 - `float` para `int`
- Uma **conversão de alargamento** transforma um valor para um tipo que pode incluir, pelo menos, aproximações de todos os valores do original
 - `int` para `float`

Conversões de Tipo: Coerção para Expressões



- Uma **expressão de modo misto** é aquela que possui operandos de tipos diferentes
- Uma coerção é uma conversão de tipo implícita
- Desvantagem de coerção:
 - Diminui poder de detectar erros do compilador
- Na maioria da linguagem, todos os tipos numéricos são convertidos (*coerced*) em expressões, usando coerção de alargamento
- Em Ada, praticamente, não é permitida coerção em expressões

Conversão de Tipo Explícita



- Chamada de *casting* em linguagens baseadas em C
- Exemplos
 - C: `(int) angle`
 - Ada: `Float (sum)`

Note que a sintaxe em Ada é similar a chamada de funções



Conversões de Tipo: Erros em Expressões



- Causas
 - Limitações inerentes à aritmética
 - Divisão por zero
 - Limitações da aritmética computacional
 - *overflow*



Expressões Relacionais e Booleanas



- Expressões Relacionais
 - Possui dois operandos e um operador relacional
 - Seu valor é booleano
 - Os símbolos de operadores variam bastante entre linguagens
 - \neq
 - \neq
 - \neq
 - \neq
 - \neq

Expressões Relacionais e Booleanas



- Expressões Booleanas
 - Operandos são booleanos e seu resultado é booleano
 - Exemplos de operadores

| FORTRAN 77 | FORTRAN 90 | C | Ada |
|--------------------|-------------------|-------------------------|------------------|
| <code>.AND.</code> | <code>and</code> | <code>&&</code> | <code>and</code> |
| <code>.OR.</code> | <code>or</code> | <code> </code> | <code>or</code> |
| <code>.NOT.</code> | <code>not</code> | <code>!</code> | <code>not</code> |

Expressões Relacionais e Booleanas:

C não possui um tipo booleano



- C não dispõe de um tipo booleano
 - Usa o tipo `int` com valor 0 para falso e valores não-zero para verdadeiro
- Características de expressões em C
 - `a < b < c`
 - é um expressão legal
 - mas o resultado não é o esperado
 - Operador mais a esquerda é avaliado produzindo 0 ou 1
 - O resultado da avaliação é então comparado com o terceiro operando (i.e., `c`)

Avaliação Curto-Circuito



- Uma expressão que tem seu resultado determinado sem avaliar todos os operandos e/ou operadores
 - Exemplo: $(13 * a) * (b / 13 - 1)$
 - Se a é zero, não existe necessidade de avaliar $(b / 13 - 1)$

Avaliação Curto-Circuito



- Problema se a avaliação não for realizada com curto-circuito

```
index = 0;  
while (index < length) && (LIST[index] != value)  
    index++;
```

- Quando `index=length`, `LIST [index]` causará um problema de indexação (assumindo que `LIST` tem `length - 1` elementos)



Avaliação Curto-Circuito



- C, C++ e Java
 - Usam avaliação curto-circuito para operadores booleanos comuns (`&&` e `||`)
 - Mas os operadores booleanos *bitwise* não são avaliados curto-circuito (`&` e `|`)
- Ada:
 - O programador pode especificar (curto-circuito é especificado com `and then` e `or else`)
- Avaliação curto-circuito expõe efeitos colaterais em expressões
 - `(a > b) || (b++ / 3)`

Instruções de Atribuição



- Sintaxe geral

`<target_var> <assign_operator> <expression>`

- Operadores de atribuição

`=` FORTRAN, BASIC, PL/I, C, C++, Java

`:=` ALGOLs, Pascal, Ada

- `=` pode ser inadequado quando é sobrecarregado para o operador relacional de igualdade

Instruções de Atribuição: Alvos Condicionais



- Alvos condicionais (C, C++ e Java)

```
(flag)? total : subtotal = 0
```

Que é equivalente a

```
if (flag)
    total = 0
else
    subtotal = 0
```



Instruções de Atribuição: Operadores Compostos



- É um método abreviado de especificar uma forma de atribuição
- Introduzido em ALGOL; adotado por C
- Exemplo

`a = a + b`

É escrito como

`a += b`

Instruções de Atribuição: Operadores Unários



- Em linguagens baseadas em C, combinam operações de incremento e de decremento com atribuição

- Exemplos

`sum = ++count` (count incremented, added to sum)

`sum = count++` (count incremented, added to sum)

`count++` (count incremented)

`-count++` (count incremented then negated)

A Atribuição como uma Expressão



- Em C, C++ e Java, a instrução de atribuição produz um resultado, que é o mesmo valor atribuído ao alvo

- Um exemplo:

```
while ((ch = getchar()) != EOF) {...}
```

`ch = getchar()` é obtido; o resultado (atribuído a `ch`) é usado como um valor condicional na instrução `while`

Atribuição de Modo Misto



- Instruções de atribuição podem ser de modo misto, por exemplo

```
int a, b;  
float c;  
c = a / b;
```

- Em Pascal, variáveis inteiras podem ser atribuídas a variáveis reais, mas variáveis reais não podem ser atribuídas a variáveis inteiras
- Em Java, apenas conversão de alargamento são permitidas

Resumo



- Expressões
- Precedência e associatividade de operadores
- Sobrecarga de operadores
- Expressões de modo misto
- Várias formas de atribuição