
Análise Semântica

Sandro Rigo
sandro@ic.unicamp.br

Introdução

- **Análise Léxica:**
 - Quebra a entrada em palavras conhecidas como *tokens*
- **Análise Sintática:**
 - Analisa a estrutura de frases do programa
- **Análise Semântica:**
 - Calcula o “significado” do programa

Tipos de Dados

- Dois propósitos principais:
 - Fornecer um contexto implícito
 - Pascal:
 - $a+b$: real ou integer?
 - `new p`: alocação no heap de tamanho correto
 - C++/Java:
 - `New m_type()`: chamada automática de construtor

Tipos de Dados

- Dois propósitos principais:
 2. Limitar o conjunto de operações semanticamente válidas
 - Registros não podem ser somados a caracteres, etc
 - Tipos de parâmetros de funções
 - Não conseguem prevenir qualquer tipo de erro
 - Pegam um número de erros suficientes para ser de grande valor prático

Sistema de Tipos

- Podemos dizer que consiste de:
 1. Um mecanismo para definição de tipos e sua associação a certas construções da linguagem
 2. Um conjunto de regras para:
 - Equivalência
 - Compatibilidade
 - Inferência

Sistema de Tipos

- Equivalência

- Determinar quando os tipos de dois valores são os mesmos

- Compatibilidade

- Determinar quando um valor de um dado tipo pode ser usado em um certo contexto

- Inferência

- Determina o tipo de uma expressão a partir dos tipos de seus componentes ou (as vezes) do contexto em volta

Sistema de Tipos

- Verificação de Tipos
 - Assegurar que o programa obedece as regras de compatibilidade da linguagem
- Linguagem fortemente tipada (strongly typed)
 - Proíbe de maneira que a implementação pode forçar, a aplicação de qualquer operação a qualquer objeto que não suporta esta operação

Sistema de Tipos

- Linguagem estaticamente tipada (statically typed)
 - É fortemente tipada
 - Verificação de tipos feita em tempo de compilação
 - Poucas linguagens satisfazem, se encarado estritamente
 - Na prática, termo é aplicado quando a maioria da verificação é feita estaticamente

Sistema de Tipos

- Ada:
 - Fortemente tipada e em sua maioria estaticamente tipada
- Pascal:
 - Maior parte estaticamente tipada
 - Não é fortemente tipada: *untagged variant records*
- C:
 - N

Sistema de Tipos

- C:
 - Não é fortemente tipada
 - Unions
 - Subrotinas com número variável de parâmetros
 - Interoperabilidade entre arrays e ponteiros
 - Implementações de C não costumam checar qualquer coisa em tempo de execução

Sistema de Tipos

- **Assembly:**
 - Não tem sistema de tipos
 - Operações de qualquer tipo podem ser aplicadas a valores em locais arbitrários (memória ou registradores)
- **Verificação Dinâmica**
 - Operações verificam em tempo de execução se os tipos de seus operandos são aceitáveis
 - LISP, Smalltalk, Scheme

Sistema de Tipos

- Verificação Dinâmica

- Maioria das verificações feitas em tempo de execução
- Operações verificam em tempo de execução se os tipos de seus operandos são aceitáveis
 - LISP, Smalltalk, Scheme, Python, Perl, ...
- Podem ser mais flexíveis e sofisticados por terem informação de tempo de execução
- Só asseguram que uma dada execução do programa está correta
- É repetida para todas as execuções

Analizador Semântico

- Checa os tipos de cada expressão
- Relaciona declarações de variáveis com seus usos
- É caracterizado pela manipulação de tabelas de símbolos
 - Mapeiam identificadores a seus tipos e localizações
- Declarações geram inclusões nas tabelas de símbolos

Analizador Semântico

- Usos geram consultas nas tabelas
- Escopo
 - Determina a visibilidade
 - Ex: variáveis locais de um método são visíveis apenas dentro do método
- Tabelas são também chamadas de ambientes
- Ambientes são conjuntos de “amarrações” do tipo:
 - {g \mapsto string, a \mapsto int}

Analizador Semântico

```
1 class C {
2   int a; int b; int c;
3   public void m(){
4     System.out.println(a+c);
5     int j = a+b;
6     String a = "hello";
7     System.out.println(a);
8     System.out.println(j);
9     System.out.println(b);
10  }
11 }
```

- σ_0
- $\sigma_1 = \sigma_0 + \{a \mapsto \text{int}, b \mapsto \text{int}, c \mapsto \text{int}\}$ (2)
- $\sigma_2 = \sigma_1 + \{j \mapsto \text{int}\}$ (5)
- $\sigma_3 = \sigma_2 + \{a \mapsto \text{String}\}$ (6)
 - Qual o tipo de a após esta adição?
 - Para ambientes
 - X+Y é diferente de Y+X
- Diga em qual ambiente é realizada a consulta para cada uso das variáveis do programa

Analizador Semântico

```
structure M = struct
  structure E = struct
    val a = 5;
  end
  structure N = struct
    val b = 10
    val a = E.a + b
  end
  structure D = struct
    val d = E.a + N.a
  end
end

package M;
class E {
  static int a = 5;
}
class N {
  static int b = 10;
  static int a = E.a + b;
}
class D {
  static int d = E.a + N.a;
}
```


Analizador Semântico

$$\sigma_1 = \{a \mapsto int\}$$

$$\sigma_2 = \{E \mapsto \sigma_1\}$$

$$\sigma_3 = \{b \mapsto int, a \mapsto int\}$$

$$\sigma_4 = \{N \mapsto \sigma_3\}$$

$$\sigma_5 = \{d \mapsto int\}$$

$$\sigma_6 = \{D \mapsto \sigma_5\}$$

$$\sigma_7 = \sigma_2 + \sigma_4 + \sigma_6$$

- nML
 - N: $\sigma_0 + \sigma_2$
 - D: $\sigma_0 + \sigma_2 + \sigma_4$
 - Result : $\{ M \mapsto \sigma_7 \}$
- Java
 - E, N e D em σ_7
 - Forward reference