Análise Semântica e Representação Intermédia

Compiladores, Aula Nº 21

João M. P. Cardoso

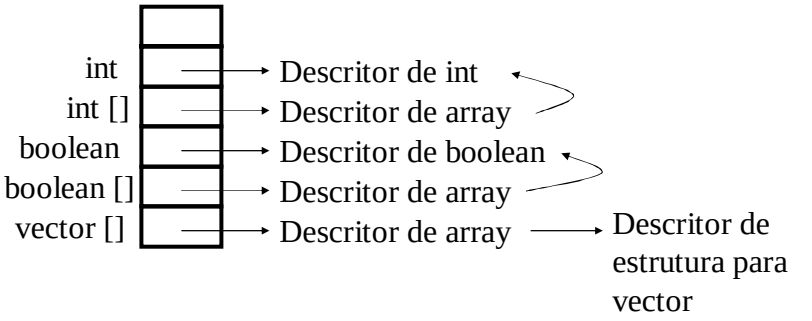
Descritores

- O que contêm?
- Informação utilizada para geração de código e análise semântica
  - Descritores locais - nome, tipo, offset na pilha
  - Descritores de funções
    - assinatura (tipo do valor retornado, e parâmetros)
    - Referência à tabela de símbolos local
    - Referência ao código para a função

Parâmetros, Local, e Descritores de Tipos

- Parâmetros, Local referem a descritores de tipo
  - Descritor de tipo base: int, boolean, etc.
  - Descritor de tipo de array, que contém referência ao descritor de tipo para os elementos do array
  - Descritor de estrutura, etc.

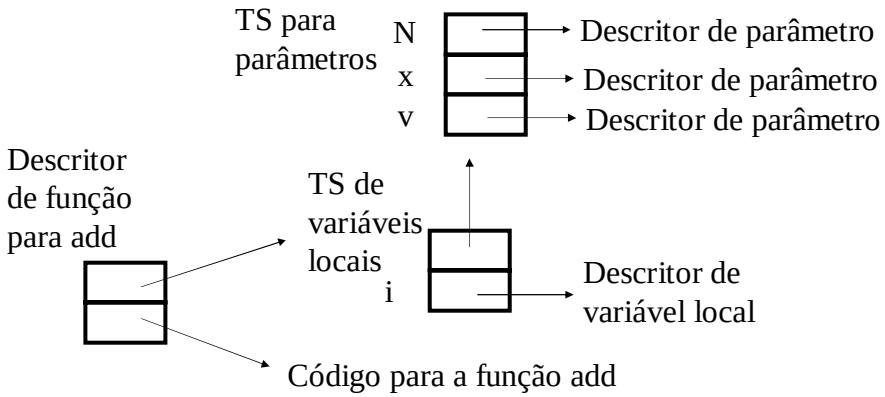
Exemplo: Tabela de Símbolos para Tipos



Descritores de funções

- Contêm referência para o código da função
- Contêm referência para a tabela de símbolos local (para as variáveis locais da função)
- Na hierarquia de tabelas de símbolos, a TS para os parâmetros é mãe da TS para as variáveis locais

Descritor de função para add



O que é uma árvore sintáctica?

- Árvore sintáctica guarda resultados da análise sintáctica
- Nós externos são terminais/tokens
- Nós internos são não-terminais

Árvores abstractas versus concretas

- Relembrar modificações à gramática
  - Factorização à esquerda, eliminação de ambiguidade, precedências dos operadores
- Modificações levam a uma árvore que não reflecte uma interpretação do programa intuitiva e clara
- Pode ser mais conveniente trabalhar com a AST (pode ser vista como a árvore sintáctica representativa da gramática sem as modificações modificações)

Construções alternativas para Representações Intermédias

- Construir a árvore sintáctica concreta, traduzir para AST, traduzir para representação intermédia
- Construir a árvore sintáctica abstracta, traduzir para representação intermédia
- Incluir a construção da representação intermédia durante a análise sintáctica
  - Elimina a construção intermédia da árvore sintáctica – melhora performance do compilador
  - Menos código a escrever

Tabela de Símbolos

- Dada uma árvore sintáctica (abstracta ou concreta)
  - Atravessar recursivamente a árvore
  - Construir a tabela de símbolos enquanto a travessia da árvore decorre

Escopos aninhados

- Várias formas de aninhamento
  - TS das funções aninhadas na TS dos globais
  - TS de locais aninhada dentro da TS da função
- Aninhamento resolve ambiguidade em possíveis conflitos
  - Mesmo nome utilizado para uma variável global e uma variável local
  - Nome refere uma variável local dentro da função

Escopos aninhados de código

- TS podem ter profundidade arbitrária com base no aninhamento do código:

```
boolean x;
int foo(int x) {
      double x = 5.0;
      { float x = 10.0;
                  { int x = 1; ... x ...}
        ... x ...
      }
      ... x ...
}
```

Nota: Conflitos de nomes com aninhamento podem reflectir erros no programa. Os compiladores geram mensagens de aviso em presença de conflitos deste tipo.

Representação de código em nível alto

- Ideia básica
  - Movimento em direcção à linguagem assembly
  - Preservar a estrutura de nível alto
    - Formato de objectos
    - Fluxo de controlo estruturado
    - Distinção entre parâmetros, variáveis locais, e campos
  - Abstracção de nível alto da linguagem assembly
    - Nós load e store
    - Acesso a armazenamento local abstracto, parâmetros e campos, e não posições de memória directamente

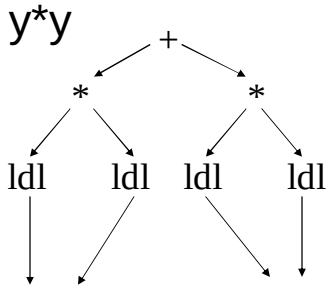
Representação de expressões

- Árvores de expressões representam as expressões
  - Nós internos – operações como: +, -, etc.
  - Folhas – Nós Load representam acesso a variáveis
- Nós Load
  - **ldl** para acesso a variáveis locais – descritor de locais
  - **ldp** para acessos a parâmetros – descritor de parâmetros
  - **lda** para acesso a arrays
    - Árvore da expressão para o valor
    - Árvore de expressão para o índice
  - Para acesso a atributos de uma classe ou campos de estruturas...

Exemplo

x e y são variáveis locais

x*x + y*y



Descritor de local para x
Na tabela de símbolos locais

Descritor de local para x
Na tabela de símbolos locais

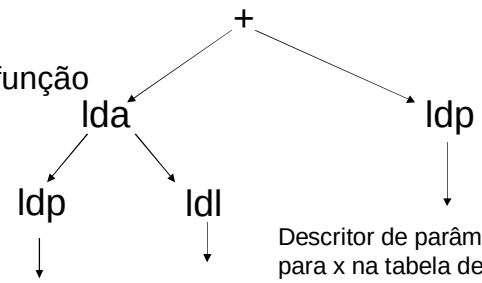
Exemplo

v é uma array passado como argumento da função add

i é uma variável local

x é um argumento da função

v[i]+x



Descritor de parâmetro para v na tabela de símbolos de parâmetros da função add

Descritor local para I na tabela de símbolos locais da função add

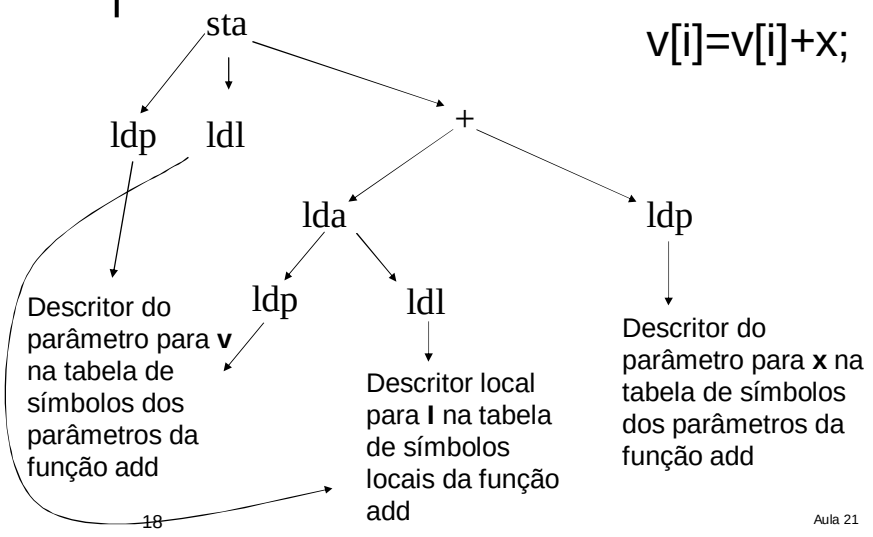
Descritor de parâmetro para x na tabela de símbolos de parâmetros da função add

Representação de enunciados de atribuição

- Nós Store
  - **stl** para *stores* em variáveis locais
    - Descritor local
    - Árvore da expressão para o valor a guardar
  - **sta** para *stores* em elementos de arrays
    - Árvore da expressão para o array
    - Árvore da expressão para o índice
    - Árvore da expressão para o valor a guardar
  - Para stores em atributos de classes ou campos de estruturas...

Exemplo

v[i]=v[i]+x;



Orientação

- Representações intermédias
  - Movimento em direcção à linguagem máquina
  - Suporte para análises do programa e acções de transformação
- IR (*intermediate representation*) de nível alto
  - Preserva estruturas de objectos e de arrays
  - Tabelas de símbolos
  - Descritores