

Análise semântica

Ivan Ricarte

2008

Sumário

Tabela de símbolos

- Aplicação

- Decoração de nomes

- Informação de escopo

Heurísticas para a análise semântica

- Verificação de tipos

- Verificação de fluxo de controle

- Verificação de unicidade

Análise semântica

Etapas da análise do código apoiada por heurísticas

- ▶ Difícil de ser formalizada por meio de gramáticas
- ▶ Associada a inter-relacionamentos entre partes distintas do código

Atividades usuais da análise semântica: verificação de...

declarações todas as variáveis usadas foram declaradas?

tipos o tipo da variável é o correto para o operador?

fluxo de controle o comando é válido nesse contexto?

unicidade o identificador (variável, rótulo) é único no escopo?

Análise semântica

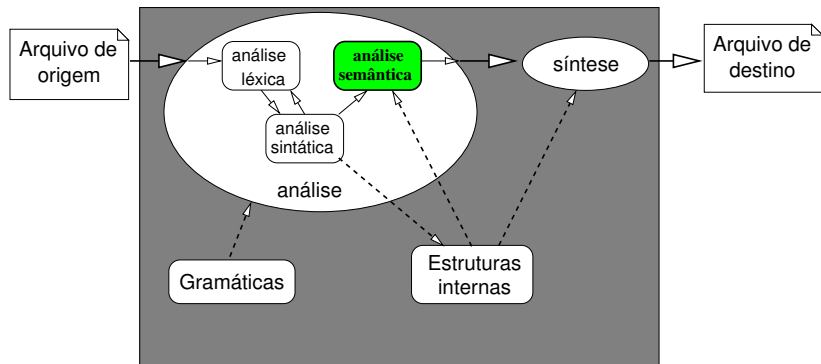


Tabela de símbolos

Estrutura auxiliar utilizada para apoiar análise semântica

- ▶ Quais símbolos foram definidos?
- ▶ O que o símbolo representa?
- ▶ Qual escopo de validade do símbolo?

Aplicação da tabela de símbolos

Variável foi declarada?

```
#include <iostream>
using namespace std;

int main() {
    a = 10;
    cout << "Valor de a: " << a << endl;
}
```

Mensagem de erro

```
...cpp: In function 'int main()':
...cpp:5: error: 'a' was not declared in this scope
```

Aplicação da tabela de símbolos

Tipo da variável está correto?

```
#include <iostream>
using namespace std;

int main() {
int a = 9;
float b = 5;
cout << "a%b: " << a%b << endl;
}
```

Mensagem de erro

```
...cpp: In function 'int main()':
...cpp:7: error: invalid operands of types 'int' and 'float' to binary
'operator%'
```

Aplicação da tabela de símbolos

Escopo da variável está correto?

```
#include <iostream>
using namespace std;
int main() {
    int a = 9;
    void mostra();
    mostra();
}
void mostra() {
    cout << "a: " << a << endl;
}
```

Mensagem de erro

...cpp: In function 'void mostra()':

...cpp:11: error: 'a' was not declared in this scope

Decoração de nomes

- ▶ Mecanismo usado pelo compilador para permitir que uma função tenha diferentes implementações para diferentes tipos de argumentos
- ▶ O nome armazenado na Tabela de Símbolos combina nome declarado pelo programador com um prefixo para identificar a decoração e com um sufixo para identificar os tipos dos argumentos

Exemplos

Declarado	Decorado
mostra()	H_mostra_v
mostra(int x)	H_mostra_i
mostra(double y, int x)	H_mostra_di

Decoração de nomes

Integração com objetos externos

- ▶ Como o mecanismo de decoração de nomes é particular de cada compilador, é preciso sinalizar quando uma aplicação integra código gerado por outros compiladores
- ▶ Por exemplo, integração entre C++ e C:

```
extern "C"
{
    int  yyparse(void);
    int  yylex(void);
    ...
}
```

Informação de escopo

Qual escopo de cada variável no seguinte fragmento?

```
int a, b;  
... // escopo 1  
void f() {  
    float a, c;  
    ... // escopo 2  
}  
... // escopo 1  
void g() {  
    int c, d;  
    ... // escopo 3  
}
```

Informação de escopo

Estratégias para manter informação de escopo na tabela de símbolos:

Múltiplas tabelas uma tabela de símbolos para cada escopo

Pilha de tabelas a cada novo escopo, um novo fragmento de tabela é empilhado na tabela geral e desempilhado ao encerrar o escopo

Heurísticas para a análise semântica

Compilador realiza verificação estática de código

- ▶ Aquela que pode ser realizada pela análise do código-fonte
- ▶ Por meio da aplicação de heurísticas relacionando fragmentos do código

Há situações nas quais problemas só podem ser detectados durante a execução do programa

- ▶ Dependem da verificação dinâmica do código

Verificação de tipos

Objetivo Verificar coerência entre declaração de variável e seu uso

Exemplos de situações com erros em C++

- ▶ Operando real para operador %
- ▶ Conflito entre variável escalar e variável ponteiro
- ▶ Conflito na conversão de variável para outro tipo

Verificação de fluxo de controle

Objetivo Detectar se comandos que controlam o fluxo da execução são usados de forma correta

Exemplos de situações com erros em C++

- ▶ Uso inadequado do comando `break`
- ▶ Uso inadequado do comando `continue`

Verificação de unicidade

Objetivo Detectar se identificadores são únicos dentro de um mesmo escopo

Exemplos de situações com erros em C++

- ▶ Nome de variável duplicado
- ▶ Nome de função com mesma decoração resultante
- ▶ Rótulo duplicado

Sugestões de leitura (Web)

Artigos na Wikipedia sobre

- ▶ Tabela de símbolos:

http://en.wikipedia.org/wiki/Symbol_table

- ▶ Decoração de nomes:

http://en.wikipedia.org/wiki/Name_mangling