

Compiladores

Análise Semântica

Bruno Lopes

```
fie(a, b, c, d) {  
    int a, b, c, d;  
    ...  
}
```

```
fee() {  
    int f[3], g[0], h, i, j, k;  
    char *p;  
    fie(h, i, "ab", j, k);  
    k = f * i + j;  
    h = g[17];  
    printf("<%s,%s>.\n", p, q);  
    p = 10;  
}
```

O que está errado?

Nível de corretude além da gramática!

```
fie(a, b, c, d) {  
    int a, b, c, d;  
    ...  
}
```

```
fee() {  
    int f[3], g[0], h, i, j, k;  
    char *p;  
    fie(h, i, "ab", j, k);  
    k = f * i + j;  
    h = g[17];  
    printf("<%s,%s>.\n", p, q);  
    p = 10;  
}
```

O que está errado?

Nível de corretude além da gramática!

Além da sintaxe

Para gerar o código, o compilador precisa responder várias questões:

- Uma variável “x” é um escalar, array ou função? Ela foi declarada?
- Existem variáveis não declaradas?
- Declaradas mas não usadas?
- A expressão “ $x * y + z$ ” tem tipos consistentes?

Além da sintaxe

- Ao referenciar “a[i,j,k]”, a variável “a” tem três dimensões?
- Onde a variável “z” será armazenada? (registrador, local, global, heap, estática)
- Quantos argumentos uma função “fie()” tem? E “printf()”?
- A expressão “*p” referencia o resultado de um “malloc()”?

Além da sintaxe

- Essas questões fazem parte da análise sensível ao contexto (ou semântica)
- As respostas dependem dos valores e não da gramática
- E envolvem informação não local
- Também podem envolver resultados de computações

Solução. . .

Formal gramáticas sensíveis ao contexto

Formal gramáticas de atributos

Ad-hoc tabelas de símbolos

Ad-hoc código *ad-hoc*

Gramática de Atributos

- Gramática livre de contexto aumentada com um conjunto de regras
- Cada símbolo na derivação tem um conjunto de valores nomeados (atributos)
- As regras especificam como computar o valor de cada atributo

Gramática de Atributos

number -> number digit

number -> digit

digit -> 0

digit -> 1

.

.

.

digit -> 9

number.val = digit.val

digit.val = 0

digit.val = 1

.

.

.

digit.val = 9

345

Árvore sintática?

Gramática de Atributos

number -> number digit

number -> digit

digit -> 0

digit -> 1

.

.

.

digit -> 9

number.val = digit.val

digit.val = 0

digit.val = 1

.

.

.

digit.val = 9

345

Árvore sintática?

Gramática de Atributos

number -> number digit

number -> digit

digit -> 0

digit -> 1

.

.

.

digit -> 9

number.val = digit.val

digit.val = 0

digit.val = 1

.

.

.

digit.val = 9

345

Árvore sintática?

$\text{exp} \rightarrow \text{exp} + \text{term} \mid \text{exp} - \text{term} \mid \text{term}$
 $\text{term} \rightarrow \text{term} * \text{factor} \mid \text{factor}$
 $\text{factor} \rightarrow (\text{exp}) \mid \text{number}$

$(34 - 3) * 42$

Árvore sintática?

exp \rightarrow exp + term | exp - term | term
term \rightarrow term * factor | factor
factor \rightarrow (exp) | number

(34 - 3) * 42

Árvore sintática?

```
decl -> type var-list
type -> int
type -> float
var-list -> id, var-list
var-list -> id
```

float x, y

Árvore sintática?

```
decl -> type var-list  
type -> int  
type -> float  
var-list -> id, var-list  
var-list -> id
```

float x, y

Árvore sintática?

num-base -> num basechr

basechr -> o

basechr -> d

num -> num digit

345o

Árvore sintática?

num-base -> num basechr

basechr -> o

basechr -> d

num -> num digit

345o

Árvore sintática?

Métodos de avaliação de atributos

Métodos dinâmicos, baseados em dependências

- constroem a árvore sintática
- constroem um grafo de dependências
- ordenam topologicamente o grafo de dependências
- definem os atributos em ordem topológica

Métodos de avaliação de atributos

Métodos baseados em regras

- analisam as regras em tempo de geração do compilador
- determinam uma ordenação fixa (estática)
- avaliam nós naquela ordem

Métodos de avaliação de atributos

Métodos com abstração

- ignoram regras e árvore sintática
- escolhem uma ordem conveniente em tempo de projeto e a usam

Grafo de dependências

- Atributos: associados com nós da árvore sintática
- Regras: atribuições de valores associadas com produções
- O atributo é definido uma vez, usando informação local
- O grafo deve ser acíclico
- Atributos sintetizados: dependem dos valores dos filhos
- Atributos herdados: dependem dos valores dos pais e/ou irmãos