

PCS2056 – Linguagens e Compiladores

Assunto: Reconhecimento. Reconhecedores descendentes (top-down)

Objetivo: Conceituação de reconhecedores e analisadores sintáticos. Analisadores sintáticos determinísticos descendentes e sua implementação pelos métodos clássicos recursivos descendentes e dirigidos por tabelas de decisão LL(1). Preparação de gramáticas para a aplicação de métodos descendentes: eliminação de regras gramaticais com prefixos comuns, e eliminação de regras recursivas à esquerda.

Palavras-chave:

sintaxe livre de contexto	análise sintática descendente (top-down)
reconhecimento e análise sintática	recursão à esquerda
reconhecimento determinístico e não-determinístico	conflitos por prefixo comum
Compiladores dirigidos pela sintaxe	método descendente recursivo
gramáticas LL(k)	tabelas de decisão para análise descendente
	método LL(k)

Questões:

- 1) Quais são as funções associadas ao assunto *sintaxe* em um compilador?
- 2) Qual é a diferença entre reconhecedor e analisador sintático?
- 3) Qual é a diferença entre determinismo e ambigüidade?
- 4) Quais são os tipos usuais de reconhecedores determinísticos canônicos? Como operam?
- 5) Que é um compilador dirigido pela sintaxe? Como funciona?

Gramática de expressões aritméticas simples

os terminais +, -, * representam operadores; **a** representa identificadores; **n** representa números; **()** e **[]** delimitam expressões e seqüências de índices de matriz, respectivamente; ϵ representa a cadeia vazia.

- | | | |
|------------------------|-----------------------------|------------------------------|
| 1. $E \rightarrow E+T$ | 6. $F \rightarrow (SE)$ | 11. $X \rightarrow \epsilon$ |
| 2. $E \rightarrow T$ | 7. $S \rightarrow -$ | 12. $Y \rightarrow I$ |
| 3. $T \rightarrow T*F$ | 8. $S \rightarrow +$ | 13. $Y \rightarrow I, Y$ |
| 4. $T \rightarrow F$ | 9. $S \rightarrow \epsilon$ | 14. $I \rightarrow a$ |
| 5. $F \rightarrow a X$ | 10. $X \rightarrow [Y]$ | 15. $I \rightarrow n$ |

Questões:

- 1) Quais são as cadeias mais curtas que se podem construir com esta gramática?
- 2) Construa ao menos 5 expressões diferentes que exercitem pelo menos 10 diferentes regras dessa gramática cada uma. Tome o cuidado de utilizar todas as regras da gramática, cada uma das quais em todas as possíveis situações que a gramática permite.
- 3) A gramática acima é LL(k)? Se for, quanto vale k? Por quê? Se não for, seria possível construir uma gramática LL(k) equivalente? Como se faz isso?
- 4) Utilize uma gramática LL(k) para a linguagem definida acima, com o menor k possível. Com base no exemplo de construção do analisador LL(1) visto em classe na aula anterior, construa um reconhecedor recursivo descendente, codificando-o como um conjunto de procedimentos booleanos mutuamente recursivos. Pressuponha a disponibilidade de um analisador léxico para a extração dos átomos das sentenças.
- 5) Exercite o reconhecedor assim construído aplicando-o sobre as expressões construídas na questão 2. Atenção ao conteúdo da pilha da linguagem de implementação ao longo do reconhecimento sintático.
- 6) Construa também uma tabela de decisão para a montagem de um analisador sintático determinístico descendente dirigido por tabelas de decisão, para essa gramática.
- 7) Exercite essa tabela com os exemplos construídos na questão 2, e construa para cada expressão a correspondente árvore de derivação, com base na aplicação do analisador dirigido por tabelas.

PCS2056 – Linguagens e Compiladores

Assunto: Reconhecedores e analisadores determinísticos ascendentes (bottom-up)

Objetivo: Estudo dos métodos clássicos de reconhecimento/análise sintática (parsing) determinística ascendente.

Palavras-chave:

reconhecimento e análise sintática

reconhecimento determinístico

ambigüidade

árvore de derivação (parsing)

análise sintática ascendente (bottom-up)

método LR(k)

tabelas de decisão para análise ascendente

produções marcadas (itens pontilhados)

conjuntos característicos de produções marcadas

(estados do autômato / tabela de decisão)

Gramática de expressões aritméticas simples

os terminais +, -, * representam operadores; **a** representa identificadores; **n** representa números; **()** e **[]** delimitam expressões e seqüências de índices de matriz, respectivamente; ϵ representa a cadeia vazia.

1. $E \rightarrow E+T$

2. $E \rightarrow T$

3. $T \rightarrow T*F$

4. $T \rightarrow F$

5. $F \rightarrow a X$

6. $F \rightarrow (SE)$

7. $S \rightarrow -$

8. $S \rightarrow +$

9. $S \rightarrow \epsilon$

10. $X \rightarrow [Y]$

11. $X \rightarrow \epsilon$

12. $Y \rightarrow I$

13. $Y \rightarrow I, Y$

14. $I \rightarrow a$

15. $I \rightarrow n$

Questões:

- 1) Conceitue reconhecimento sintático ascendente. Compare-o tecnicamente com o reconhecimento descendente.
- 2) O que é uma gramática LR(k)?
- 3) A gramática acima é LR(k)? Se for, quanto vale k? Por quê?
- 4) Empregue a técnica de construção direta do modelo de autômato por tabela de decisão, para a montagem de um analisador sintático ascendente a partir dessa gramática. Exercite-o com algumas sentenças da linguagem e com cadeias que não fazem parte da linguagem.
- 5) Repita a questão anterior usando a técnica das produções marcadas e conjuntos característicos. Compare o autômato assim obtido com o construído na questão anterior. Exercite-o também com as mesmas cadeias usadas no teste proposto na questão anterior.
- 6) Compare as árvores de derivação obtidas nas duas questões anteriores com as obtidas na questão 7 da última aula.