Análise Lexical

Compiladores, Aula Nº 5

João M. P. Cardoso

Linguagens Formais

- Linguagem natural
  - Ambígua
    - problema no processamento das linguagens
    - dependência do contexto permite mensagens mais curtas
- Linguagem (artificial) formal
  - Obedece a regras apresentadas rigorosamente através do recurso a formalismos apropriados
  - Regras garantem a não ambiguidade da linguagem

Linguagens formais e definição da linguagem

- Necessidade de definir precisamente uma linguagem
- Estrutura por camadas na definição da linguagem
  - Começar por um conjunto de símbolos da linguagem
  - **Estrutura lexical** - identifica "palavras" na linguagem (cada palavra é uma sequência de símbolos)
  - **Estrutura Sintáctica** - identifica "frases" na linguagem (cada frase é uma sequência de palavras)
  - **Semântica** – significado do programa (especifica que resultados deverão ser obtidos para as entradas)

Especificação Formal de Linguagens

- Expressões regulares (método generativo)
  - Existem casos que não se podem descrever por expressões regulares
- Autómatos finitos (método por reconhecimento)
  - Não deterministas (NFAs)
  - Deterministas (DFAs)
  - Implementam qualquer expressão regular

Especificação de Estruturas Lexicais utilizando Expressões regulares

- Dado um vocabulário/alfabeto ? = conjunto de símbolos
- Expressões regulares são construídas com:
  - ? - string vazia
  - Qualquer símbolo do alfabeto ?
  - $r_1 r_2$ – expressão regular r_1 seguida de r_2 (sequência): concatenação (às vezes utiliza-se o **.**)
  - $r_1 | r_2$ – expressão regular r_1 ou r_2 (selecção)
  - r* - sequência iterativa ou selecção ? | r | rr | …
  - Parêntesis para indicar precedências
    - Prioridade: *, **.**, |

Expressões regulares

- Reescrever a expressão regular até se obter apenas uma sequência de letras (string)

Regras gerais

1) $r_1 | r_2$? r_1
2) $r_1 | r_2$? r_2
3) r* ? rr*
4) r* ? ?

Exemplo

(0 | 1)*”.”(0 | 1)*
(0 | 1)(0 | 1)*”.”(0 | 1)*
1(0 | 1)*”.”(0 | 1)*
1”.”(0 | 1)*
1”.”(0 | 1)(0 | 1)*
1”.”(0 | 1)
1”.”0

Não determinismo na geração

- Diferente aplicação de regras pode conduzir a resultados finais diferentes

Exemplo 1

(0 | 1)*”.”(0 | 1)*
(0 | 1)(0 | 1)*”.”(0 | 1)*
1(0 | 1)*”.”(0 | 1)*
1”.”(0 | 1)*
1”.”(0 | 1)(0 | 1)*
1”.”(0 | 1)
1”.”0

Exemplo 2

(0 | 1)*”.”(0 | 1)*
(0 | 1)(0 | 1)*”.”(0 | 1)*
0(0 | 1)*”.”(0 | 1)*
0”.”(0 | 1)*
0”.”(0 | 1)(0 | 1)*
0”.”(0 | 1)
0”.”1

Linguagem gerada por expressões regulares

- Conjunto de todas as strings geradas pela expressão regular é uma linguagem de expressões regulares
- Em geral, uma linguagem pode ser infinita
- String na linguagem é chamada de token

Linguagens e Expressões Regulares

- Exemplos:
- ? = {0, 1, "." }
  - (0 | 1)*"."(0 | 1)* - números binários de vírgula flutuante
  - (00)* - strings de zeros com comprimento par
  - (1*01*01*)* - strings com um número ímpar de zeros
- ? = {a, b, c, 0, 1, 2 }
  - (a | b | c)(a | b | c | 0 | 1 | 2)* - identificadores alfanuméricos
  - (0|1|2)* - números ternários

Expressões Regulares

- Outras construções:
  - r^+ - uma ou mais ocorrências de r: r | rr | rrr ...
    - Equivalente a: $r.r^*$
  - $r^?$ – zero ou uma ocorrência de r: (r | ?)
  - [ ] – classes de símbolos:
    - [ac] o mesmo que: (a | c)
    - [a-c] o mesmo que: (a | b | c)

Expressões Regulares

- Especifique a linguagem de Inteiros
- Especifique a linguagem de identificadores (uma letra seguida de sequências de letras e algarismos)
- Enumere propriedades algébricas das expressões regulares
- Dê exemplos de linguagens que não podem ser especificadas por expressões regulares