Notas de aula - 20/02/2014

Análise Léxica (cont.)

1. Analisadores léxicos como autômatos finitos
2. Relembrando autômatos finitos
   (a) Alfabeto de entrada
   (b) Conjunto de estados
   (c) Estado inicial
   (d) Estados finais
   (e) Transições entre estados
3. Mapeando isso para código
   (a) Como construir a máquina de estados
   (b) Loop mais simples: exemplo-sem-lex

Análise Sintática

1. Classes de linguagens (Hierarquia de Chomsky)
   (a) Tipo 3: Linguagens regulares (expressões regulares, autômatos finitos)
   (b) Tipo 2:
      i. Linguagens livres de contexto determinísticas (GLC determinísticas, autômatos de pilha determinísticos)
      ii. Linguagens livres de contexto (GLC; não existe algoritmo para informar se uma GLC é ambígua, e algumas LLC são inerentemente ambíguas!)
   (c) Tipo 1: Linguagens primitivamente recursivas (máquina de Turing linearmente limitada, linguagem com loops limitados)
   (d) Tipo 0: Linguagens recursivamente enumeráveis (máquinas de Turing)
2. Linguagens livres de contexto
   (a) A classe das LLC contém a classe das linguagens regulares
   (b) Mais expressivas que as linguagens regulares, mas ainda longe de cobrir todas as linguagens
   (c) Porém, cobrem estruturas hierárquicas em geral, como parênteses balanceados, estruturas em bloco, etc.
   (d) linguagens livres de contexto são a base para a estrutura sintática de linguagens de programação

3. Nos interessam aqui gramáticas livre de contexto (em especial as determinísticas)
   (a) É uma gramática $G = (V, T, P, S)$
      i. V são "variáveis" (símbolos não-terminais)
      ii. T são símbolos não terminais (o alfabeto da linguagem que a gramática escreve)
      iii. P são regras de produção $P : V \times (V \cup T)^*$
      iv. $S \in V$ é um símbolo inicial
   (b) com a restrição de que qualquer regra de produção em P possui a forma:
   $A \to \alpha$
   onde A é uma variável de V (isto é, um não-terminal) e α é uma palavra de $(V \cup T)^*$ (isto é, não-terminais e terminais).
   (c) Ou seja, uma GLC é uma gramática no qual o lado esquerdo das produções contém exatamente uma variável
4. Por que o nome "livre de contexto":
   (a) Porque a expansão de um não-terminal em uma palavra não depende do contexto na palavra onde ele está inserido.
   (b) Isto é, para expandir um não-terminal A para γ em uma regra $\alpha A\beta$ (isto é, para fazer $\alpha A\beta \Rightarrow \alpha\gamma\beta$), só preciso de uma regra $A \to \gamma$, e esta expansão não depende do contexto onde A estava (ou seja, não depende nem de α nem de β).
5. Backus-Naur Form, ou BNF: Notação popular para gramáticas livres de contexto em computação (John Backus e Peter Naur trabalharam na criação de Algol; BNF foi criada por Backus; Naur chamou de Backus Normal Form; Donald Knuth mudou para Backus-Naur Form)
   (a) Identificadores entre $\langle\rangle$ são não-terminais, e os demais são terminais
   (b) Uso de | para alternativa, [] para trechos opcionais e {}$*$ para repetições (às vezes sem o asterisco).
   (c) <stat> ::= if <exp> then <block> {elseif <exp> then <block>} [else <block>] end | ...
6. Exemplos de gramáticas
   (a) expressões: $S \to x|y|z|S + S|S - S|S * S|S/S|(S)$
   (b) Mini-0!