

PCS2056 – Linguagens e Compiladores

Assunto: Recuperação de erros em autômatos finitos e em autômatos de pilha estruturados.

Objetivo: Completar o estudo da sintaxe de linguagens de programação com a inclusão de mecanismos de recuperação de erros de sintaxe. Estudar a recuperação absoluta de erros simples e a recuperação de erros múltiplos por meio da técnica de “panic mode”.

Palavras-chave:

ponto de ocorrência de erro
ponto de detecção de erro
região de influência do erro
erro simples

erros múltiplos
correção automática
recuperação absoluta
panic mode

Enunciado:

- 1) Quais são todos os erros simples que podem ocorrer no primeiro estado de um autômato que deve aceitar apenas a cadeia “abc”?
- 2) Como fica o autômato do exercício anterior se ele for modificado para aceitar também todos os casos incorretos previstos?
- 3) Repita o procedimento para os outros dois estados do reconhecedor da cadeia "abc". Como deve ser tratada a recuperação de erros no estado final do autômato finito?
- 4) Descreva o método para todos os estados de um autômato finito.
- 5) Estenda o procedimento para as sub-máquinas de um autômato de pilha estruturado. Como tratar as chamadas e os retornos de sub-máquinas? Sugestão: Usar look-ahead nas transições em vazio, inclusive nos estados finais.
- 6) “Panic Mode” é um método radical para recuperação de erros genéricos, e consta da remoção da parte da cadeia de entrada compreendida entre o ponto de detecção do erro e um ponto em que seja encontrado um símbolo de sincronização, ou seja, um elemento da cadeia de entrada que esteja ou possa ser associado com facilidade a um estado bem determinado do autômato. Nesse ponto força-se tal estado e prossegue-se o reconhecimento.

PCS2056 – Linguagens e Compiladores

Assunto: Formas intermediárias.

Objetivo: Apresentar formas intermediárias de programas como alternativas para a geração de código, e seu uso potencial em atividades de otimização.

Palavras-chave:

formas intermediárias	quádruplas
notação infixa	árvores de sintaxe
notação polonesa prefixa	sintaxe concreta e abstrata
notação polonesa reversa	código alinhavado
ênuplas	máquinas abstratas
triplas	p-code

Conceitos

formas intermediárias: linguagens de saída de um passo de compilação, e entrada do passo seguinte.

notação polonesa prefixa ou funcional operação operando1 operando2 ... operando n

notação polonesa reversa (operandos, operação)

ênuplas (operador, operando1, operando2, ... operando n)

triplas (operador, operando1, operando2)

quádruplas (operador, operando1, operando2, operando3)

árvores de sintaxe – similares às árvores de derivação, porém em geral codificadas

sintaxe concreta – inclui todos os símbolos usados na codificação do texto do programa

sintaxe abstrata – elimina do texto-fonte os símbolos que não contribuem para a análise/compilação

código alinhavado – corresponde a uma lista ligada de ponteiros para procedimentos

máquinas abstratas – hardware virtual com conjunto de instruções aderente à linguagem de alto nível

p-code – código intermediário para a linguagem Pascal.

Exercícios:

- 1) Partindo de um reconhecedor sintático de expressões aritméticas, já construído anteriormente, acrescente rotinas semânticas que gerem como saídas as expressões equivalentes em formato de notação polonesa prefixa.
- 2) Repita o exercício anterior para notação polonesa reversa.
- 3) Usando como linguagem de saída a forma intermediária de triplas, compile expressões aritméticas para essa forma. Lembre-se de que as triplas, por convenção, associam o resultado da operação ao número sequencial da tripla dentro da seqüência de triplas que compõem o código gerado.
- 4) Repita para quádruplas. Lembre-se de que o terceiro operando da quádrupla é o nome do local em que a quádrupla deposita o resultado da operação.
- 5) Codifique uma expressão relativamente longa usando as formas intermediárias citadas. Represente-a em forma de árvore. Compare a árvore com as demais notações e estabeleça uma correspondência entre essas formas alternativas de representação.
- 6) Procure na literatura ou na Internet descrições de máquinas abstratas importantes, como a máquina virtual da linguagem Java, a máquina P (cuja linguagem é o P-code), etc. Estude uma delas, e identifique as vantagens de sua utilização. Por que não se costuma usar algum hardware real que a interprete?