

## PCS2056 – Linguagens e Compiladores

**Assunto:** Exercício – Projeto de um meta-analisador sintático a partir de gramáticas de Wirth.

**Objetivo:** Exercitar a criação de autômatos diretamente a partir do conhecimento de uma linguagem, sem passar por todas as etapas formais de sua obtenção.  
Automatizar o processo de obtenção de reconhecedores sintáticos a partir de gramáticas denotadas na notação tradicional de Wirth.

**Palavras-chave:**

autômato de pilha estruturado	atribuição de estados às expressões na notação de Wirth modificada
notação de Wirth modificada	expressões de Wirth modificadas numeradas
auto-recursões à direita/esquerda e centrais	mapeamento em transições do autômato
grafo de dependência dos não-terminais	eliminação de não-determinismos
não-terminais essenciais e não-essenciais	minimização do autômato
expressões de Wirth modificadas	

**Questões:**

- 1) Desenhar um autômato de pilha estruturado que reconheça gramáticas na notação de Wirth tradicional.
- 2) Acrescentar ao autômato da questão anterior rotinas semânticas que o transformem em um transdutor capaz de gerar como saída uma gramática com as expressões numeradas conforme o algoritmo de atribuição de estados estudado nas aulas anteriores.
- 3) Desenhar outro autômato de pilha estruturado que reconheça as gramáticas com expressões numeradas, obtidas como saídas na questão anterior.
- 4) Acrescentar-lhe rotinas semânticas que gerem como saídas as transições de estados do autômato correspondente à gramática de Wirth fornecida.
- 5) Construir um interpretador para o conjunto de regras de transição de estados gerado na questão anterior.
- 6) Testar o conjunto de programas construído partindo de uma gramática de Wirth, e gerando o autômato correspondente, e alimentando esse autômato com sentenças da linguagem definida pela gramática. Se for o caso, use, como parte do interpretador das transições do autômato, o analisador léxico construído anteriormente.

**Referências:**

J. J. Neto, C. B. Pariente and F. Leonardi, **Compiler Construction: A Pedagogical Approach**. *Proceedings of the ICIE*. Buenos Aires. 1999. [disponível para download em [www.pcs.usp.br/~lta](http://www.pcs.usp.br/~lta)]

## PCS2056 – Linguagens e Compiladores

**Assunto:** Semântica informal - declarações

**Objetivo:** Estudo da semântica das declarações em uma linguagem imperativa.

**Palavras-chave:**

Escopos: nomes locais e globais, estrutura de blocos aninháveis

Declarações de variáveis simples:

Declarações de vetores e matrizes: linearização do espaço de endereçamento

Notação para selecionar elementos de matrizes e vetores

Declarações de registros

Notação para selecionar campos de registros

Declarações de novos tipos

Declarações de variáveis ou agregados usando os novos tipos

Declarações de procedimentos e funções com parâmetros

Chamadas de procedimentos e funções paramétricas

**Questões:**

- 1) Explique como deve ser interpretada uma linguagem que apresente estrutura de blocos aninháveis:
  - a) o que significam os diversos níveis de aninhamento dos blocos?
  - b) qual é o âmbito em que têm significado os nomes de variáveis, matrizes, etc., que tenham sido declarados em um determinado bloco?
  - c) o que acontece aos elementos declarados em um dado bloco quando o controle da execução do programa não se encontra dentro do escopo ao qual eles pertencem?
  - d) o que deve acontecer quando se declara um nome já declarado no mesmo escopo?
  - e) e se o mesmo nome pertencer ao conjunto de nomes de algum escopo envolvente?
  - f) o que ocorre se forem declarados nomes iguais em blocos paralelos (não aninhados)?
- 2) Deduza uma fórmula de linearização para calcular, em relação ao endereço do seu primeiro elemento, os endereços virtuais de um elemento genérico de uma matriz cujos elementos possam ser bytes, palavras de 32 bits ou agrupamentos de 64 bits conforme o tipo do elemento (char, int ou double, respectivamente).
- 3) Em um cadastro, cada um dos 200 registros do arquivo correspondente guarda as seguintes informações: (a) o nome do cliente (máximo 128 caracteres); (b) data de nascimento (três inteiros - para dia, mês e ano); endereço [campos: logradouro (máximo 128 caracteres), número da casa (inteiro), CEP (inteiro), bairro (máximo 32 caracteres), cidade (máximo 32 caracteres)]. Reservando-se para os registros a área suficiente para armazenar cada campo em seu formato máximo, estabeleça uma fórmula geral para calcular o endereço de cada um dos campos específicos para cada um dos registros que compõem o cadastro.
- 4) Para cada um dos tipos de dados apresentados acima, levante as informações junto com as quais os dados devem ser armazenados para que seja possível executar os respectivos algoritmos de endereçamento.
- 5) Em relação a novos tipos de dados, definidos pelo usuário, faça um estudo de como deve ser efetuado o seu tratamento, de tal modo que possam ser declaradas variáveis ou agregados contendo elementos dos novos tipos. Como podem ser eles endereçados no momento de seu uso? É possível estipular uma fórmula geral para esse endereçamento?