

PCS2056 – Linguagens e Compiladores

Assunto: Sintaxe formal

Objetivo: Exercitar as três notações mais usadas como metalinguagens para com elas definir formalmente uma linguagem de programação.

Palavras-chave:

meta-linguagens

Notação de Wirth

BNF

Diagramas de sintaxe

Questões:

1) Formalize em BNF, usando a recursão para a definição de construções repetitivas, a linguagem de programação abaixo:

Uma Mini-linguagem

(descrição informal e incompleta de uma pequena linguagem de programação, adaptada de Ledgard & Marcotty *The programming Language Landscape*, cap.2)

programa consiste da seqüência

declarações: seqüência de ocorrências da forma

comando de atribuição:

comando condicional: primeira forma:

segunda forma:

comando iterativo:

comando de entrada:

comando de saída:

program *declarações begin comandos end ;*

declare *lista de identificadores ;*

posição de memória *:= expressão;*

if *condição then comandos else comandos end if;*

if *condição then comandos end if ;*

while *condição loop comandos end loop ;*

input *lista de posições de memória;*

output *lista de expressões;*

A linguagem deverá conter:

- uma seção de declarações e outra de comandos executáveis
 - declarações de variáveis simples, de tipos aritmético e booleano
 - comandos: atribuição, condicionais, iterações, entrada e saída
 - expressões aritméticas com parênteses aninháveis, envolvendo variáveis e constantes aritméticas, chamadas de funções aritméticas com parâmetros, as operações aritméticas de soma, subtração, multiplicação, divisão, potenciação, e o sinal negativo unário
 - expressões booleanas com parênteses aninháveis, envolvendo variáveis e constantes booleanas, comparações de ordem (maior, menor, igual, diferente) entre expressões aritméticas, comparações de igualdade (igual, diferente) entre expressões booleanas, e as operações booleanas usuais (AND, OR, NOT)
 - os comandos de entrada efetuam uma leitura de dados no dispositivo padrão de entrada (por exemplo, o teclado) e armazenam o resultado de tal leitura, devidamente convertido para formato interno correspondente, em uma variável, indicada pelo comando. No comando podem ser indicadas diversas variáveis, as quais serão seqüencialmente preenchidas dessa maneira
 - os comandos de saída devem permitir imprimir no dispositivo padrão de saída (por exemplo, a tela do computador) constantes opcionais (números, valores booleanos ou cadeias de caracteres), acompanhados do valor obtido como resultado da avaliação de uma expressão aritmética ou booleana. Mais de um valor pode ser especificado desta maneira para ser impresso por esse comando
- Reescreva a gramática obtida na questão anterior usando a Notação de Wirth, na qual sejam substituídas, sempre que for conveniente, as definições recursivas por definições iterativas equivalentes.
 - Use diagramas de sintaxe para representar a mesma linguagem, partindo da versão gramatical em notação de Wirth iterativa
 - Exercite novamente o uso das três meta-linguagens repetindo os três exercícios acima para definir a linguagem representada pela sintaxe das expressões regulares. A propósito, esta sintaxe é regular? Justifique a sua resposta.

PCS2056 – Linguagens e Compiladores

Assunto: Sintaxe formal - 2a. parte

Enunciado da **parte 2** do trabalho: Definição formal da sintaxe de uma linguagem de programação simples, de alto nível

Data de entrega: **13 de outubro de 2016**

Objetivo: Exercitar a definição sintática de linguagens de programação imperativas em seus diversos aspectos, e definir a linguagem a ser adotada no projeto.

Palavras-chave:

Declarações de variáveis simples	Escopos, variáveis locais e globais
Declarações de agregados homogêneos	Parâmetros e argumentos
Declarações de agregados heterogêneos	Sintaxe de comandos de múltipla escolha
Declarações de tipos	Rótulos e Comandos de desvio explícito
Declarações de procedimentos e funções	Macros simples e macros paramétricas

Conceitos:

1. Formalize a sintaxe das construções acima mencionadas para uma linguagem de programação imperativa típica, representando-a através das notações metalingüísticas estudadas nas aulas anteriores.
2. Proponha uma alteração no analisador léxico que você implementou de tal modo que ele passe a reconhecer a definição de novas macros, e seja capaz de expandir todas as suas ocorrências no texto-fonte.

Questões:

- 1) Complete os formalismos desenvolvidos nos exercícios da aula anterior, especificando, no mínimo, as seguintes novas construções sintáticas:
 - a) construções sintáticas adicionais capazes de permitir ao programador a definição de escopos estáticos aninhados (estrutura de blocos)
 - b) a definição (declaração) e o uso (em expressões) de agregados homogêneos – vetores e matrizes – em todos os pontos em que for aceita uma variável simples do mesmo tipo ou de tipo compatível
 - c) a declaração e o uso (chamada) de procedimentos e funções tipadas, com parâmetros. Notar que o uso de chamadas de procedimentos exige a criação de um comando correspondente, e que o uso de chamadas de funções deve prever uma alteração da sintaxe das expressões do tipo compatível com o tipo da função.
- 2) Levante o conjunto definitivo das palavras reservadas dessa linguagem, e adapte o seu analisador léxico para reconhecê-las e classificá-las corretamente. Faça o mesmo acerca de símbolos compostos de mais de um caractere ASCII.
- 3) Defina formalmente a linguagem assim especificada, usando cada uma das três notações metalingüísticas apresentadas anteriormente: BNF, notação de Wirth e Diagramas de Sintaxe. Muita atenção aos detalhes da notação: presença de $\langle \rangle$ em tomo do nome dos não-terminais em BNF, aspas envolvendo os terminais na notação de Wirth, o ponto final depois de cada regra na notação de Wirth, a ausência de notação para fechamentos em BNF, etc.
- 4) Crie, a partir da gramática, um conjunto de arquivos contendo pequenos programas corretos, escritos na nova linguagem, os quais deverão ser utilizados para as atividades de teste do futuro reconhecedor sintático da linguagem. Por ora, utilize-os para alimentar o analisador léxico desenvolvido na primeira parte do projeto.