

Universidade Federal de Roraima
Departamento de Ciência da Computação
Linguagens de Programação
DISCIPLINA: Linguagens de Programação – DCC206
1ª Lista

Aluno(a): Débora Bianca Taveira de Moura

1) Conceitue e descreva as diferenças entre (i) linguagem de alto-nível e (ii) linguagem de baixo-nível. Descreva o papel e a importância do compilador no processo de criação de programas de computador.

Alto-nível: A linguagem se assemelha à linguagem humana, possuindo um nível de abstração que facilita a compreensão do código.

Baixo-nível: A linguagem se assemelha à linguagem da máquina, diferentemente da linguagem de alto-nível pois sua sintaxe exige que você entenda registradores ou instruções do processador.

O compilador converte um programa em uma linguagem de alto-nível para um programa equivalente em código de máquina em um processador, isso que o torna tão essencial no processo de criação de programas, pois caso ocorra algum erro na compilação, é possível verificar o erro e corrigi-lo, contribuindo assim para a expansão no nicho de desenvolvedores, pois isso facilita a vida do desenvolvedor, além de contribuir positivamente para a vida útil do programa.

2) Faça um programa escrito na linguagem C, C++, Python e Perl que aceite como entrada um número inteiro n maior ou igual a 1 e retorne como saída o valor da série.

Executa o arquivo questao2.(c, ccp, sh, pm)

Input: 1;

Output: O resultado obtido e 0.25

3) Faça um programa escrito na linguagem C, C++, Python e Perl que leia uma quantidade indeterminada de strings na entrada a partir de um arquivo e identifique quais são as duas strings que, respectivamente, são o primeiro e o segundo na ordenação lexicográfica.

4) Faça um programa escrito na linguagem Shell Script para efetuar o download das imagens do tipo .jpg, .jpeg ou .png de um site por meio da utilização de sua URL. Adicionalmente, deve-se ao final do download de todas as imagens informar o tempo total para todas as imagens. Exemplo de execução do programa:
./downscript.sh http://ufrr.br

Executa o arquivo: ./questao4.exe

Input: <http://ufrr.br>

Output:

(ARQUIVOS BAIXADOS)

Tempo total decorrido: (tempo)s

Baixados: (valor) arquivos, (valor)M em (tempo) ((valor) KB/s)

5) Descreva as seguintes categorias de linguagens de programação e apresente o nome de duas linguagens de programação com seus respectivos exemplos.

(A)Imperativas: é definida por executar tarefas em série, manipulando variáveis baseando-se em comandos, onde variáveis são lidas num processo de entrada, manipuladas ou alteradas e por fim atualizadas, gerando a saída. Ex: Fortran e Pascal

(B)Funcionais: baseia-se no conceito matemático de função, em que para cada elemento do seu conjunto domínio (entrada) há apenas um elemento no seu conjunto contradomínio (saída). Além disso, as funções são normalmente expressas por meio de outras funções - de modo que obter o valor da função para um determinado conjunto de parâmetros envolve não só aplicar as regras daquela função, mas também fazer uso de outras funções. Ex: LISP e Miranda

(C)Lógicas: é baseada na utilização de sentenças lógicas, utilizando-se da lógica simbólica como linguagem para programação. Ex: QLISP e Mercury

(D)Marcação/Híbrida: é uma linguagem portátil, pois desde que tenha o interpretador, seu código intermediário pode ser executado em diferentes arquiteturas. Ex: Java e C#

6) Escreva uma gramática no formato BNF e não ambígua para expressões envolvendo as variáveis A, B, C e os operadores * (multiplicação) e ^ (exponenciação). A precedência deve seguir as regras usuais da matemática.

`<program> ::= begin <stmt_list> end`

`<stmt_list> ::= <stmt> | <stmt> <stmt_list>`

`<stmt> ::= <expression>`

`<expression> ::= <var><operator><expression> | <var><operator><var><expression> |`

`<var><operator><var> | <var>`

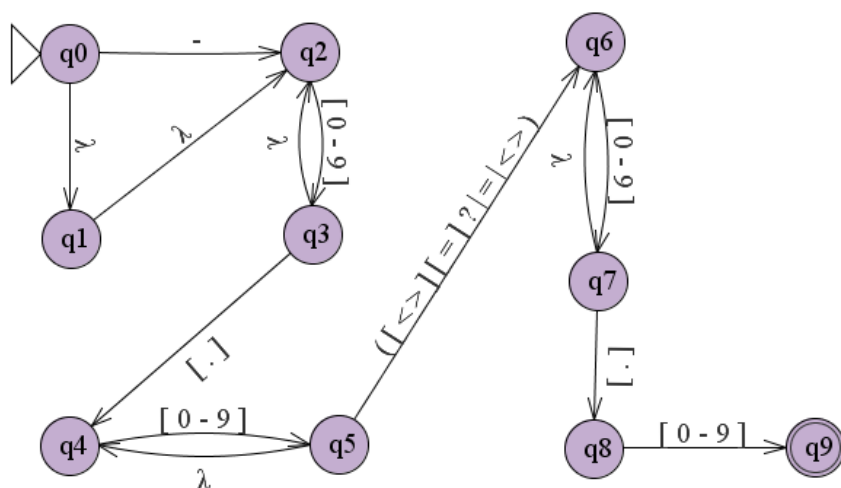
`<var> ::= A|'B'|'C'`

`<operator> ::= '*' | '^'`

7) Descreva o que é um paradigma de programação.

Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características comuns. A classificação de linguagens em paradigmas é uma consequência de decisões de projeto que tem impacto na forma segundo a qual uma aplicação real é modelada do ponto de vista computacional.

8) Apresente um autômato para reconhecer uma comparação entre números reais. A comparação deve ser feita utilizando os seguintes símbolos: < (menor), > (maior), >= (maior igual), <= (menor igual), = (igual), <> (diferente). Exemplo: 1.2 <= 59.13



9) Defina análise semântica e descreva: Semântica Operacional; Semântica Axiomática; e Semântica Denotacional.

Semântica Operacional: descreve o significado de um programa pela execução de suas instruções em uma máquina, seja ela real ou simulada; As mudanças no estado da máquina

(memória, registradores, etc.) definem o significado da instrução; Para usar a semântica operacional em linguagens de alto nível, uma máquina virtual é necessária;

Semântica Axiomática: baseada em lógica (cálculo de predicados), tem como propósito original a verificação formal de programas. Axiomas ou regras de inferência são definidas para cada tipo de instrução na linguagem (para permitir transformações de expressões para outras expressões). As expressões são chamadas de asserções (predicados). Pré-condição é uma asserção que precede imediatamente uma instrução de programa e descreve as variáveis dela nesse ponto.

Pós-condição é uma asserção que segue imediatamente a uma instrução e descreve as novas restrições a essas variáveis (e possivelmente a outras) depois da execução da instrução.

Semântica Denotacional: baseado na teoria de funções recursivas. O método mais abstrato de descrição da semântica.

O processo de construção da especificação denotacional para uma linguagem define um objeto matemático para cada entidade da linguagem; Define uma função que mapeia instâncias das entidades da linguagem em instâncias dos correspondentes objetos matemáticos;

A ideia baseia-se no fato de que há maneiras rigorosas de manipular objetos matemáticos, mas não construções de linguagens de programação

10) Utilizando a semântica axiomática apresente a pré-condição para os seguintes programas.

PROGRAMA – A

$x > 122 * y - 144$

$\{x > 144\}$

$122*y - 144 > 144$

$y > (144+144)/122$

$y > 2,36$

PROGRAMA – B

$y = 5 * x - 5$

$x = y + 5$

$\{x < 45\}$

$y + 5 < 45$

$y < 45/5$

$y < 9$

$5 * x - 5 < 9$

$5 * (y+5) - 5 < 9$

$5y + 25 - 5 < 9$

$5y < 29$

$y < 5$

PROGRAMA – C

if ($x < 200$)

$y = y + 2;$

else

$y = y - 2;$

$\{y > 2\}$

$y + 2 > 2$

$y > 0$

$y - 2 > 2$

$y > 4$

$\{y > 0\} \Rightarrow \{y > 4\}$

$\{y > 4\}$

PROGRAMA – D

Caso a condição não seja atendida a pré-condição que se validará é que $i == n$. Se ela for atendida uma vez, i obrigatoriamente dever ser $i = n - 1$. Para que ela seja atendida mais de uma vez $i < n - 1$. Logo, se i dever ser $<$ ou $=$ à n , a pré-condição deve ser definida como: $\{i \leq n\}$.