

Trabalhos Compiladores (grupos de 1 ou 2 alunos)

1º bimestre: (prova 60%, trabalhos 40%)

Especificação mínima da linguagem (30%):

Prazo: 11/02/2015

- Como regra geral usar a forma de análise dos trabalhos de CLP.
- Especificar a forma geral de um programa.
- A especificação de tipos deve ser estática, com no mínimo o seguinte conjunto de tipos de dados: (devem ser especificadas quais as operações de cada tipo):
 - inteiro;
 - ponto flutuante;
 - caractere;
 - booleano;
 - cadeia de caracteres;
 - arranjos unidimensionais.
- Conjunto mínimo de operadores: (especificar ordem de precedência e associatividade):
 - aritméticos:
 - aditivos, multiplicativos;
 - unário negativo;
 - relacionais, para numéricos, caracteres e cadeias de caracteres;
 - lógicos:
 - negação, conjunção e disjunção;
 - concatenação de cadeias de caracteres.
- Instruções: (especificar natureza e formas de controle)
 - estrutura condicional de uma e duas vias;
 - estrutura iterativa com controle lógico;
 - estrutura iterativa controlada por contador;
 - entrada e saída.
- Atribuição pode ser instrução ou operador (especificar).
- Funções
- Incluir os seguintes exemplos, que deverão ser testados usando o analisador léxico.
 - *alô mundo*;
 - série de Fibonacci até um termo limite usando iteração com controle lógico;
 - *shell sort* em um arranjo usando iteração controlada por contador.

Especificação dos tokens (30%):

Prazo: 25/02/2015

- Especificar a linguagem de programação em que os analisadores léxico e sintático serão implementados
- Especificar a enumeração com as categorias dos *tokens* a ser **obrigatoriamente** usada nos analisadores léxico e sintático, usando a sintaxe da linguagem escolhida para a implementação dos analisadores.
- Especificar em dois grupos **distintos**:
 - expressões regulares auxiliares, que não representam terminais da linguagem, e
 - lexemas, que representam os terminais da linguagem, especificando a categoria do *token* associado
- A especificação das expressões regulares devem seguir a simplificação do padrão EBNF ISO/IEC 14977: 1996(E) como usada em sala de aula (nomes sem espaços e sem vírgulas entre elementos das produções).

Analizador Léxico (40%):

Prazo: 10/03/2015

- Deve ser implementado para fazer a análise “*on the fly*”, devolvendo o *token* identificado via um método/função cuja assinatura seja

Token nextToken();

para o analisador sintático, e não como um passo em separado que faça a análise léxica toda antes da análise sintática, onde **Token** deverá ser uma **struct** ou **class** contendo os dados do token.

- Deve ser implementado um programa para teste que execute o analisador léxico, listando para cada *token* sua posição (linha e coluna) no programa exemplo, sua categoria (nome associado) e seu valor léxico, que deverá estar presente nos objetos **Token**.
- Devem ser apresentados os resultados dos testes para os três programas previamente solicitados.

SUGERE-SE FORTEMENTE que versões prévias sejam apresentadas ao professor para análise e discussão.

Trabalhos Compiladores (grupos de 1 ou 2 alunos)

2º bimestre: (prova 60%, trabalhos 40%)

Especificação da gramática da linguagem (40%):

Prazo: 29/04/2015

- Especificar qual o tipo de analisador sintático será implementado:
 - analisador descendente preditivo recursivo;
 - analisador descendente preditivo tabular;
 - analisador de precedência de operadores;
 - analisador ascendente SLR(1);
 - analisador misto.(especificar onde cada tipo será usado).
- A gramática deverá ser especificada de acordo com o que foi visto em sala de aula, **especificando precedência e associatividade para os operadores.**
- A especificação das produções devem seguir a simplificação do padrão EBNF ISO/IEC 14977: 1996(E) como usada em sala de aula (nomes sem espaços e sem vírgulas entre elementos das produções).
- **Não** são admitidas formas **condicionais** ou com **repetições** nas produções, já que as mesmas não são tratáveis pelos analisadores estudados.
- Dependendo do analisador sintático escolhido deverão ser especificados também:
 - analisador descendente preditivo recursivo ou tabular:
 - a especificação LL(1) correspondente da gramática, com as restrições necessárias;
 - analisador descendente preditivo tabular, analisador de precedência de operadores ou ascendente SLR(1):
 - tabela de análise, como vistas em sala de aula;
 - analisadores mistos:
 - cada parte de acordo com o analisador utilizado.

Analisador sintático(60%):**Prazo: 29/04/2015**

- Deverá ser implementado o analisador especificado no item anterior.
- Deverá ser gerada informação que permita desenhar a árvore de derivação:
 - numerar cada nó não terminal da árvore sequencialmente;
 - cada derivação/redução efetuada deverá ter sua produção listada, com os não terminais identificados pela sua ordem sequencial entre parênteses, e os terminais com seus lexemas entre parênteses, de acordo com os exemplos para a expressão:

a + b * c

- descendente: (apenas primeiros terminais com lexema)

Gramática:	Resultado esperado:
E = T Er Er = "opa" T Er Er = ε T = F Tr Tr = "opm" F Tr Tr = ε F = "id"	E(1) = T(2) Er(3) T(2) = F(4) F(4) = "id"(a) Er(3) = "opa"(+) T(5) Er(6) T(5) = F(7) Tr(8) F(7) = "id"(b) Tr(8) = "opm"(*) F(9) Tr(10) F(9) = "id"(c) Tr(10) = epsilon Er(6) = epsilon

- ascendente:

Gramática:	Resultado esperado:
E = E "opa" T E = T T = T "opm" F T = F F = "id"	F(1) = "id"(a) T(2) = F(1) E(3) = T(2) F(4) = "id"(b) T(5) = F(4) F(6) = "id"(c) T(7) = T(6) "opm"(*) F(6) E(8) = E(3) "opa"(+) T(7)

- no caso do analisador de precedência de operadores ou de analisador misto, adequar de acordo com os modelos acima, em caso de dúvida consultar o professor.
- Os exemplos dos trabalhos do 1º bimestre deverão ser analisados.

SUGERE-SE FORTEMENTE que versões prévias sejam apresentadas ao professor para análise e discussão.