DCC045 - Teoria dos Compiladores
Departamento de Ciência da Computação — DCC - UFJF
Professor: Leonardo Vieira dos Santos Reis — lvsreis@ice.ufjf.br
Trabalho Prático: Análise Léxica

1 Instruções Importantes

Nessa seção são apresentadas diversas informações relevantes referentes a entrega do trabalho e orientações a serem seguidas durante a implementação do mesmo. Leia atentamente antes de começá-lo.

1.1 Equipe de Desenvolvimento

O trabalho será desenvolvido individualmente ou em dupla.

1.2 Escolha da Linguagem de Programação

O trabalho deverá ser desenvolvido na linguagem Java. No que conserne a implementação, todas as estruturas de dados referente a modelagem e desenvolvimento do trabalho deverão ser implementadas. Poderão ser usadas apenas bibliotecas de estruturas de dados elementares (listas, filas, pilhas, árvores, etc.) e leitura de strings.

1.3 Artefatos a Serem Entregues

Os artefatos a serem entregues são:

- código fonte do programa;
- arquivo de *build* para compilação do fonte (makefile, ant, shell script, etc);
- documentação do trabalho em formato pdf.

Antes de enviar seu trabalho para avaliação, assegure-se que:

1. seu código compila e executa em ambiente Unix / Linux. Programas que não compilam receberão nota zero;
2. todos os fontes a serem enviados têm, em comentário no início do arquivo, nome e matrícula do autor do trabalho;
3. arquivo de documentação tenha a identificação do autor do trabalho;
4. arquivo compactado com os artefatos estão devidamente identificados com nome e matrícula.

1.4 Critérios de Avaliação

A avaliação será feita mediante análise do código fonte, documentação e apresentação do trabalho (entrevista). Os seguintes fatores serão observados na avaliação do código fonte: corretude do programa, estrutura do código, redigibilidade e legibilidade. A corretude se refere à implementação correta de todas as funcionalidades especificadas, i.e., se o programa desenvolvido está funcionando corretamente e não apresenta erros. Os demais fatores avaliados no código fonte são referentes a organização e escrita do trabalho. Nesse quesito, será observado:

- estruturas de dados bem planejadas;
- código modularizado em nível físico (separação em arquivos) e lógico;
- legibilidade do código, i.e., nomes mnemônicos de variáveis, funções e comentários úteis no código;

A documentação do código deve conter informações relevantes para compilar, executar e auxiliar no entendimento do código fonte. Ressalta-se que na documentação não deve conter cópias do fonte – afinal o seu fonte é um dos artefatos enviado, mas deve apresentar as decisões de projetos tomadas: expressões regulares e autômatos para cada *token* da linguagem, estruturas de dados usadas e estratégia de implementação do analisador léxico e de "bufferização" da entrada, dentre outras informações.

O trabalho deverá ser apresentado ao professor da disciplina e, só serão avaliados após a realização da entrevista, i.e., trabalhos que não forem apresentando não terão nota. Na entrevista, o discente deverá elucidar, ao menos, como modelou e resolveu o problema, os resultados e conclusões obtidas. A entrevista também tem a finalidade de avaliar a confiabilidade e segurança do autor do código em explicar pontos relevantes do trabalho desenvolvido.

Assim, a entrevista influenciará na avaliação dos artefatos entregues. Portanto, a nota final será dada a partir da avaliação do conjunto do código fonte, documentação e entrevista. **É de responsabilidade do discente solicitar a marcação do dia e horário da entrevista com o professor da disciplina**.

Atrasos serão penalizados por uma função exponencial de dias de atrasos, i.e., será reduzido do percentual da nota a exponencial na base 2 dos dias de atraso. A tabela a seguir mostra a nota em função dos dias de atraso:

Dias de Atraso	Nota
1	n*0.98
2	n*0.96
3	n*0.92
4	n*0.84
5	n*0.68
6	n*0.36
7	0

Observe que a partir do 7º dia de atraso seu trabalho não será mais avaliado.

2 Especificação Técnica do Trabalho

Ao longo do semestre será construído um compilador para a linguagem *lang*. Assim, nesse primeiro trabalho pede-se que seja implementado um analisador léxico.

Deve-se implementar um programa de teste que usa o analisador léxico implementado. Tal programa irá receber como entrada recebe um arquivo contendo um programa da linguagem e imprime, na saída padrão, a sequência de *tokens* produzido pelo analisador léxico, um *token* por linha. Por exemplo, para o programa da Figura 1 a saída será:

```
ID:main
(
)
{
PRINT
ID:fat
(
INT:10
...
```

```
main() {
  print fat(10);
}

fat(num :: Int) : Int {
  if (num < 1)
    return 1;
  else
    return num * fat(num-1);
}

divmod(num :: Int, div :: Int) : Int, Int {
  q = num / div;
  r = num % div;
  return q, r;
}
```

Figura 1: Exemplo de programa na linguagem

3 Entrega do Trabalho

A data da entrega do trabalho será até o dia **05 de outubro de 2020**.