



Universidade Federal do Cariri
Centro de Ciências e Tecnologia
Ciência da Computação
Trabalho 2
CC0066 – Autômatos e linguagens formais

Quantidade máxima de integrantes: 4 estudantes

Prazo para envio do e-mail com os integrantes da equipe: 23:59 do dia 14/10/2024

Prazo para entrega do trabalho: 23:59 do dia 04/11/2024

Linguagem de programação permitida: C

Tamanho máximo do arquivo contendo o código fonte: 2 KB

Descrição

Neste trabalho, a equipe deverá desenvolver um programa que resolva o problema de membresia em gramáticas livres-de-contexto. O programa deve ser feito obrigatoriamente com utilização da linguagem C com a restrição de que nenhuma biblioteca pode ser utilizada, com exceção apenas da biblioteca *stdio.h*.

Os estudantes devem enviar um e-mail ao docente consistindo dos nomes completos dos integrantes da equipe. É necessário apenas um e-mail por equipe. Após o envio deste e-mail, os estudantes devem esperar a resposta do docente, que virá contendo uma gramática livre-de-contexto e um alfabeto. A gramática e alfabeto recebidos serão a gramática e alfabeto que a equipe irá trabalhar de forma que será uma gramática diferente para cada equipe. Após o recebimento da sua gramática e alfabeto, a equipe deve então desenvolver um programa que consiste em receber uma string do usuário, que usará apenas caracteres no alfabeto da equipe, e determinar se essa string faz parte ou não da linguagem da gramática da equipe.

Formato da entrada e saída

O programa deve receber a string através da entrada padrão (teclado) e deve imprimir a resposta na saída padrão (tela). A entrada consistirá apenas de caracteres contidos no alfabeto da equipe. A entrada terá tamanho de no mínimo zero caracteres (string vazia) e de no máximo 1000 caracteres. Esse limite desconsidera os caracteres de controle ao final da string da linguagem C, como o ‘\0’ ou o ‘\n’.

A saída deve ser apenas a string “sim\n” caso a string de entrada esteja na linguagem da gramática da equipe, ou “nao\n” caso contrário. Note que não há acentos ou espaços nas strings de saída. O programa não deve imprimir na saída padrão nada além de uma dessas duas strings. Após imprimir a saída, o programa deve finalizar a sua execução.

Entrega e avaliação

O código fonte em C, que deve estar em um único arquivo com extensão `.c`, deve ser enviado ao professor por e-mail até prazo supracitado, com uma tolerância de 10 minutos, como resposta ao email anterior do docente que especifica o alfabeto e a gramática livre-de-contexto da equipe. E-mails enviados dentro do prazo estipulado ou após o prazo estipulado, mas dentro da tolerância de 10 minutos, serão considerados normalmente para fins de correção e nota. Trabalhos enviados após a tolerância de 10 minutos, sofrerão um decréscimo de 1,0 ponto mais 1,0 ponto por cada hora completa de atraso.

Códigos fontes enviados em uma linguagem de programação diferente da linguagem C serão desconsiderados para fins de avaliação. Também serão desconsiderados códigos na linguagem C que incluam bibliotecas (diretiva `#include`) diferentes da `stdio.h`, que é a única biblioteca permitida, ou ainda que desrespeitem a restrição do tamanho máximo do arquivo. Em todos esses casos, a nota da equipe será zero. O código fonte será compilado no Linux com a linha de comando

```
gcc -O3 -Wall -pedantic arquivo.c
```

onde `arquivo.c` é o nome do arquivo que contém o código fonte. Se o arquivo não for compilado corretamente pelo compilador (erro de compilação), a nota da equipe será zero. Por isso, é aconselhado aos estudantes que testem os seus códigos fontes no Linux utilizando a linha de comando supracitada para que não haja surpresas ao terem os seus códigos compilados pelo docente. É aconselhado também aos estudantes que tenham atenção aos *warnings* apresentados pelo compilador, mas isso não será critério para avaliação e, caso haja *warnings* na compilação, a equipe não sofrerá nenhuma penalidade por isso. Não é necessário comentários no código.

Uma vez compilado com sucesso, os trabalhos serão avaliados da seguinte forma:

- O programa será testado através de 20 casos de testes, ou seja, o programa será executado 20 vezes, com uma entrada diferente em cada execução;
- O programa terá 1 segundo para apresentar a resposta de cada caso de teste;
- Para cada caso de teste, uma resposta correta dentro do tempo limite de 1 segundo valerá 0,5 pontos. Uma resposta errada ou fora do tempo limite valerá zero pontos, assim como um erro em tempo de execução;
- A nota do trabalho será a soma dos pontos obtidos nos 20 casos de testes;

Todos os casos de teste serão executados e verificados de forma automática. Por isso, é importante que cada equipe se certifique de que o seu programa respeite rigorosamente o formato de entrada e saída especificado na seção anterior.