UNIVERSIDADE FEDERAL DE LAVRAS
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO

GCC 108 – Teoria da Computação
Professor: Rafael S. Durelli

Observações

- O trabalho deve ser feito em grupos de no máximo 2 componentes
- Trabalhos entregues após a data limite não serão aceitos.
- Data limite de entrega: **3 de Dezembro de 2021 : 23h59m**
- Enviar para o email: **rafae.durelli@ufla.br**, o link de um repositório no *Github* que contenha a implementação do trabalho.

Trabalho Prático

Nesse trabalho, o grupo deve implementar uma Máquina de Turing Universal que implemente heurísticas para o Problema da Parada. Chamaremos essa máquina de UH (*Universal with Heuristics ☺)* daqui para frente. As decisões de projeto, implementação e simulações serão apresentadas pelo grupo durante as aulas reservadas para apresentação do trabalho prático.

Implementação

O programa deve ser implementado em Python ou Java, ou caso a dupla tenha conhecimento com outra linguagem de programação deverá conversar comigo e informar qual linguagem foi escolhida. De tal forma que sua chamada se dê por linha de comando, seguindo o padrão a seguir:

Python

```
Python programa.py argumento1
```

Java

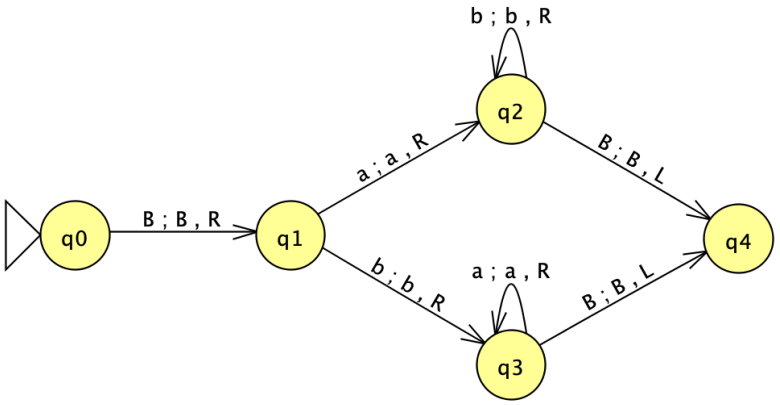
```
java -jar programa.jar argumento1
```

Na definição acima, `argumento1` é o caminho para um arquivo texto, codificado em UTF-8, que contém a representação de uma máquina de Turing M qualquer ($R(M)$) seguida de uma entrada w.

O arquivo referenciado em `argumento1` conterá $R(M)$ como definido em sala de aula (representação binária de uma máquina de Turing M) seguido de uma representação da entrada w. A entrada w terá sua representação com cada símbolo concatenado por um zero (0). Assim, seja uma máquina de Turing M, com alfabeto de entrada $\Sigma = \{a, b\}$, e representações $a = 1$, $b = 11$ e $B = 111$ então, uma entrada $w = aab$ terá sua representação ($R(w)$) igual a 1110101011000. Os três zeros no final indicam o fim da entrada e um símbolo branco ($B = 111$) é posto no início da entrada para o processo de 'inicialização da máquina'.

Vamos, também, formalizar as representações dos movimentos, a saber, $R = 1$ e $L = 11$. Consideraremos apenas as máquinas de Turing padrões, isto é, sem movimentos estáticos.

Considere, para clarificar a definição acima, o seguinte exemplo:



M:

Então (as quebras de linha foram apenas inseridas para facilitar a interpretação),

$$R(M) = 000$$
$$1011101101110100$$
$$11010111010100$$
$$11011011110110100$$
$$11101101110110100$$
$$1110111011111011101100$$
$$11110101111010100$$
$$11110111011111110111011$$
$$000$$

Considerando uma entrada $w = abbb$, então,

$$R(w) = 11101011011011000$$

A entrada para UH será a concatenação de $R(M)$ com $R(w)$ (novamente, as quebras de linhas servem apenas para melhor interpretação):

$$000$$
$$1011101101110100$$
$$11010111010100$$
$$11011011110110100$$
$$11101101110110100$$
$$1110111011111011101100$$
$$11110101111010100$$
$$11110111011111110111011$$
$$000$$
$$11101011011011000$$

As únicas restrições com relação à implementação do trabalho é que se deve manter, para cada fita utilizada, uma estrutura de dados sequencial e indexável (exemplos: em Python, uma lista e, em Java, um *Vector* ou *ArrayList*), **que possa ser impresso a cada passo da simulação de** M. Por se tratar de uma simulação de uma máquina de Turing Universal com algumas heurísticas de verificação de loop infinito, pelo menos 3 fitas devem ser utilizadas.

As heurísticas terão a finalidade de parar a simulação de M, tão logo UH detecte que M pode tenha entrado em loop infinito. Obviamente, heurísticas podem falhar e parar uma execução que iria terminar mais à frente.

Todas as heurísticas devem ser descritas apresentando exemplos que em funcionam e, opcionalmente, exemplos em que falham. Considere como falha de uma heurística quando ela para a execução de uma máquina que não entraria em loop infinito.

Apresentação

O grupo apresentará para a sala tanto a sua implementação como as definições das heurísticas. Para isso, devem se certificar que conseguem executar o código durante a aula e exibir no projetor. Caso precisem, podem solicitar ao professor a execução do código no computador do mesmo.

O grupo pode preparar slides para as definições das heurísticas e apresentação geral do projeto. Inclusive, é incentivada a preparação de slides para isso.

Durante a apresentação, o grupo deve formular exemplos em que suas heurísticas funcionam e, opcionalmente, exemplos em que suas heurísticas falham. Devem exibir a simulação apresentando o conteúdo das fitas de UH em momentos propícios para melhor entendimento da simulação.

O professor fornecerá exemplos de entradas para execução no momento da apresentação. Nesses casos, não será descontado ponto caso a heurística falhe ou deixe de identificar um possível loop infinito. Será descontado ponto caso a simulação de UH falhe, em algum ponto.

Será reservado um período para a sala realizar perguntas.

É importante que todos os componentes do grupo participem da apresentação. Caso contrário, os componentes que não participarem da apresentação terão dois pontos descontados de sua nota do trabalho.

Pontuação:

A pontuação máxima do trabalho será de **60 pontos**, distribuídos nos itens a seguir:

- **30 pontos** para qualidade de código:
  - legibilidade;
  - modularização;
  - documentação.
- **30 pontos** para apresentação do trabalho:
  - clareza;
  - apresentação geral da solução;
  - definições de cada heurística;
  - simulações da UH;
  - respostas às perguntas realizadas.

Em caso de dúvidas, enviar email para rafael.durelli@ufla.br.