

Diagrama de Máquinas de Turing

Sobre:

O aplicativo permite a criação de Diagramas de Máquinas de Turing, com o objetivo de simplificar a criação de Máquinas de Turing complexas. O usuário pode criar módulos, definir regras e observar a execução do diagrama ao passar uma string como entrada.

Instruções:

Para executar um diagrama, passe por parâmetro o diretório do arquivo .dt contendo a definição do diagrama e o estado inicial da fita.

Exemplo:

.\TuringMachineSimulator diagrama.dt aabbb

Os arquivos .dt devem ter o seguinte formato:

```
modulo %R r.mt
modulo S s.mt
modulo T t.mt
R [a,b] S
S [*] T
R [#] T
```

Instrução de carregamento de módulo:

Deve conter a keyword “modulo”, o nome desejado para o módulo e o nome do arquivo .mt contendo a descrição do módulo. É possível utilizar o mesmo arquivo de módulo, mas com nomes diferentes. O módulo inicial pode ser especificado utilizando o símbolo “%” antes do nome do módulo.

Exemplo1:

```
modulo R r.mt
modulo %R2 r.mt
```

Carrega o módulo especificado pelo arquivo r.mt e atribui o identificador R a esse módulo. Ao carregar R2, a aplicação detecta que o arquivo “r.mt” já foi carregado e apenas adiciona uma nova referência para ele, com o nome R2. O módulo R2 será o primeiro módulo a ser iniciado.

Regras:

São definidas especificando o módulo inicial, o símbolo a ser lido após a execução do módulo inicial e o módulo seguinte a ser executado. É possível declarar uma lista de valores aceitos utilizando a seguinte notação: [a1,a2,...,an]. Também é possível utilizar o símbolo “*” para determinar que o módulo seguinte será executado, independente do símbolo lido.

Exemplo1:

```
R [a,b] S
```

Executa o módulo R e se, após a execução de R, a cabeça de leitura ler um “a” ou “b”, executa o módulo S em seguida.

Exemplo 2:

```
S [*] T
```

Executa o módulo S e, independente do símbolo lido em seguida, o módulo T será executado.

Arquivos .mt:

Os arquivos .mt devem conter as máquinas de turing correspondentes e estar no mesmo diretório do arquivo dt. O arquivo deve seguir o mesmo padrão já definido com o simulador feito pelo prof. Carlos. Também é possível executar um arquivo .mt diretamente, sem a necessidade de um diagrama.

Exemplo1:

```
q0 1000 1000
```

```
q0 a q0 >
```

```
q0 b q0>
```

```
q0 # q1 #
```

Avança a cabeça de leitura até encontrar um “#”.
O estado inicial deve ser definido no cabeçalho.

Diagramas e Módulos com variáveis:

É possível declarar variáveis nos diagramas e passar essas variáveis para os módulos, caso eles aceitem uma variável. Um módulo com variáveis deve ter o seguinte formato:

```
q0 1000 1000
var x
```

```
q0 * q1 x
Escreve o valor de x.
```

As variáveis devem ser declaradas no cabeçalho do módulo. O valor será passado pelo diagrama para o módulo. Um módulo pode ter no máximo uma variável.
Um diagrama com variáveis tem o seguinte formato:

```
modulo R r.mt
modulo R# r#.mt
modulo W wx.mt
```

```
R [x=a,b] R#
R# [#] W(x)
W [*] END
O diagrama copia o primeiro caractere para o final da string de entrada.
```

Não é necessário declarar variáveis no diagrama. Para atribuir um valor a uma variável, basta definir a regra de transição da seguinte forma, como definido no exemplo:

```
MóduloInicial [var=a,b] MóduloFinal
```

A variável “var” irá armazenar o valor “a” ou “b” lido na fita.

Para utilizar o valor de uma variável, basta adicionar um “(var)” após o nome do módulo final, na declaração de uma regra. Por exemplo:

```
MóduloInicial [a,b] MóduloFinal(var)
```

Se MóduloFinal suportar o recebimento de uma variável (ou seja, se tiver alguma declaração de var dentro do módulo), o valor armazenado em “var” será passado para o módulo.

Um diagrama pode utilizar diversas variáveis. Por exemplo:

```
R [x=a] R#
R [y=b] R#
R# [*] L
L [a] W(y)
L [b] W(x)
W [*] END
```

Caso uma variável seja passada para um módulo, mas nenhum valor tenha sido atribuído a ela, a aplicação deverá mostrar essa variável com o valor NULL e o símbolo “#” será passado para o módulo.

Observações:

Não é necessária nenhuma biblioteca adicional.
Dúvidas críticas e sugestões, procure-nos durante as aulas de teoria.

Links:

SVN: <http://code.google.com/p/turing-machine-simulator-ufc-2012-1/>