

Engenharia de Linguagens MEI

Ficha de Avaliação 2

Nuno Oliveira | Pedro Henriques

Considere o formato de texto .ma usado para a descrição de Autómatos de Markov. O conceito Autómatos de Markov, muito usado na modelação de sistemas interactivos e probabilísticos, refere-se a um autómato cujas transições podem ser probabilísticas (i.e., transita-se de um estado para outro por uma dada ação α com uma probabilidade ρ) ou markovianas (i.e., transita-se de um estado para outro por um determinado ritmo ou taxa (referente a uma distribuição exponencial)).

Tendo em conta algumas restrições impostas nestes autómatos, podemos obter uma Cadeia de Markov Interativa (Interactive Markov Chain – IMC). Neste modelo as transições entre estados continuam a ser de 2 tipos: markovianas e interativas (i.e., transita-se de um estado para outro por uma dada ação α). As transições interativas podem ser internas ou externas. As internas significam que são invisíveis do exterior e por convenção nomeiam-se τ (tau).

O seguinte exemplo mostra a descrição de um Autómatos de Markov usando a notação .ma.

```
#INITIALS
s0
#GOALS
s2
#TRANSITIONS
s0 a
* s1 0.3
* s0 0.7
s0 b
* s3 1
s1 !
* s2 2
s2 !
* s1 3
* s2 1
s3 tau
* s2 0.6
* s3 0.4
```

Algumas considerações sobre a representação “.ma”:

- #INITIALS, #GOALS e #TRANSITIONS definem o conjunto de estados iniciais, finais e de transições.
- As transições probabilísticas são fatorizadas por estado e ação. Por exemplo, do estado `s0` saem 2 transições pela ação `a`, para os estados `s1` e `s0` com probabilidades 0.3 e 0.7 respetivamente.
- As probabilidades devem estar sempre compreendidas entre 0 e 1.
- A soma das probabilidades nas transições de um estado por uma dada ação devem sempre somar 1 (indica que a probabilidade de sair desse estado é de 100%).
- As transições markovianas usam o símbolo ‘!’ (como nome da ação) para ser diferenciadas das probabilísticas. Contudo são fatorizadas da mesma maneira.
- O valor da taxa nas transições markovianas é um qualquer número real maior que zero.
- O identificador de um estado pode ser um número inteiro (e.g., 110), alfanumérico (e.g., `state`, `s0`, `0s_s2` ou `aindacompute()`).

Se, no entanto, quisermos tomar este exemplo como sendo a descrição de uma IMC, as probabilidades nas transições probabilísticas (denominadas por interativas nas IMCs) devem ser ou 0 ou 1 (ignorando o ponto 4 acima). Todas as restantes considerações feitas acima se mantêm válidas.

Neste contexto, resolva as seguintes alíneas:

1. Em AntLR crie uma gramática adequada à validação sintática de IMCs descritos no formato .ma.
2. Desenvolva um processador, adicionando atributos à gramática criada, de modo a que este avise (notando linha e posição):
 - (a) (através de warnings) quando uma transição interativa está a ser declarada como probabilística (ou seja, o valor da probabilidade não é 0 ou 1).
 - (b) (através de erros) quando para um mesmo estado existem várias transições pela mesma ação (incluindo o '!') e estas não estão fatorizadas.
 - (c) (através de warnings) quando existe um *loop markoviano* num estado com mais do que uma transição markoviana. Considere um *loop markoviano* como sendo uma transição markoviana de um estado para si próprio.
 - (d) (através de warning) quando existe uma transição markoviana partindo de um estado instável. Um estado instável é um estado origem para transições internas (i.e., pela ação `tau`).
 - (e) (através de erros) quando a taxa de uma transição markoviana é 0.
3. O processador pedido deve contar ainda (i) o número de estados iniciais, finais, o total de estados e de transições, (ii) o número de estados com não determinismo (atenção que o não determinismo só acontece em transições interativas pela mesma ação!) e (iii) o número de estados instáveis.
4. Para auxiliar este trabalho, é ainda fornecido **um package** com classes Java para a criação de objetos IMC. Neste contexto, o processador de ficheiros .ma deve criar um objeto IMC. Este objeto, assim como uma lista de erros e outra de warnings (ou apenas uma com distinção clara do tipo de erros), deve ser posteriormente acessível via métodos get do Java.

Ainda no contexto desta alínea, e para colocarem no relatório, devem utilizar o método `toDOT-Format()` da classe IMC para gerarem uma imagem com o grafo correspondente a um texto .ma de entrada.