Laboratório 3: Animação de Processos no MWB

Luís Soares Barbosa

Sumário

O objectivo deste trabalho é introduzir o MOBILITY WORKBENCH, *uma ferramenta de modelação e análise de processos suportando o* π-calculus. *Após uma rápida habituação ao ambiente de trabalho, é convidado a utilizar o* MWB *para resolver alguns problemas menos triviais de modelação em* π-calculus.

O texto que se segue deverá ser lido com o MWB *activado de forma a poder responder às questões e exercícios propostos. As suas respostas deverão ser dadas de forma clara, sucinta e manuscrita. Complementarmente poderá anexar algum* printout *do* MWB.

Disciplina: Métodos de Programação IV (2005-06)
Docente *Luís Soares Barbosa,* Departamento de Informática, Universidade do Minho

---

1 Familiarização

1.1 O MWB

O MWB(acrónimo do *Mobility Workbench*) é uma ferramenta de domínio público, que incorpora um conjunto de técnicas para modelação e verificação de sistemas concorrentes em π-*calculus*. Permite, em particular, testar a bissimilaridade e a equivalência observacional (na versão *open*) e, tal como o CWB-NC, verificar a satisfação de propriedades expressas numa lógica adequada.

O MWB, e documentação associada, pode ser obtido para diversas plataformas computacionais no endereço

`/www.it.uu.se/research/group/mobility/mwb`

1.2 Aquecimento

Após instalar o sistema, faça *downlaod* do *User's Guide*, familarize-se com a sintaxe utilizada, os comandos disponíveis e anote as diferenças relativamente ao CWB-NC. Convém, antes de mais, atentar nas formas de definição de processos.

Exercício 1

Defina

```
MWB> agent P(a) = a.0
MWB> agent Q(a) = a.0 | 'a.Q<a>
```

e invoque o comando `env`, que fornece as definições existentes no ambiente. O resultado é

```
MWB>env
agent P = (\a)a.0
agent Q = (\a)(a.0 | 'a.Q<a>)
MWB>
```

Interprete este resultado à luz da discussão nas aulas sobre a definição de processos como funções de nomes para processos. Tente agora o comando `step`, que permite a simulação de execução passo a passo. O resultado será

```
MWB> step Q(z)
* Valid responses are:
  a number N >= 0 to select the Nth commitment,
  <CR> to select commitment 0,
  q to quit.
0: |>t.(z.0 | 'z.Q<z>)
1: |>z.'z.Q<z>
2: |>'z.(z.0 | z.0 | 'z.Q<z>)
Step>
```

requerendo, depois, a participação interactiva do experimentador.

No *User's Guide* do MWB encontra diversos exemplos de processos simples, na sua quase totalidade variantes de diferentes tipos de *buffers*, que deverá testar. Poderá, igualmente, formular no MWB exemplos das Fichas Laboratoriais anteriores. Teste os comandos `eq`, `eqd`, `weq`, e `weqd` sobre esses processos sem mobilidade. Que pode concluir sobre as equivalências usadas em π-*calculus* quando aplicadas a processos CCS?

Considere, agora, a codificação em MWB de uma versão simplificada do conhecido modelo do *JobShop* proposto por R. Milner. Consideram-se apenas dois trabalhadores e duas ferramentas:

```
agent JobShop = (^ getT,putT,getR,putR) Line

agent Line = Worker1 | Worker2 | ToolT | ToolR

agent Worker1(getT,getR,putT,putR,workW1)
      = 'getT.'getR.'workW1.'putR.'putT.Worker1<getT,getR,putT,putR,workW1>
agent Worker2(getT,getR,putT,putR,workW2)
      = 'getR.'getT.'workW2.'putR.'putT.Worker2<getT,getR,putT,putR,workW2>

agent ToolT(getT,putT) = getT.putT.ToolT<getT,putT>
agent ToolR(getR,putR) = getR.putR.ToolR<getR,putR>
```

Exercício 2

Experimente simular este modelo no MWB. Deliene uma estratégia para verificar que a exclusão mútua é efectivamente garantida.

1.3 Processos Móveis no MWB

Comece por definir o seguinte processo

```
agent T(a) = (^ m) a(x) . 'x<m> . T<a>
```

e considere o resultado de aplicar a `T` o comando `sort`:

```
MWB>sort T
Object sort: (a)
Sorting:
 {[a]->([x]),[x]->([m]),[m]->?}
MWB>
```

Experimente ainda simular alguns passos da sua execução:

```
MWB>step T(u)
* Valid responses are:
  a number N >= 0 to select the Nth commitment,
  <CR> to select commitment 0,
  q to quit.
0: |>u.(\x)(^~v)'x<~v>.T<u>
Step>
Abstraction (\~v1)
0: |>'~v1.(^~v)[~v](^m)u(x).'x<m>.T<u>
Step>
Concretion (^~v1)[u]
[Circular behaviour detected]
0: |>u.(\x)(^~v)'x<~v>.T<u>
Step>
```

Exercício 3

Interprete os resultados que obteve e relacione-os com as aulas sobre π-*calculus* (recorde, em particular, a noção de *abstracção* e *instância*). Defina um outro processo que coloque em paralelo com `T` de forma a ilustrar a passagem de ligações entre processos e alteração dinâmica do escopo de nomes.

Defina, de seguida, o processo

```
agent Y(b,c,d)  =  (^ a) ('b<a>.0 | a(x).x.0)  |  b(c).'c<d>.0
```

Exercício 4

Simule este processo no MWB e comente os resultado que obtiver.

Considere agora o seguinte modelo de um *buffer* 'elástico' em π-*calculus*.

```
agent ElasticBuf(in,out) = in(x) . (^ m) ( LB<in,m> | RC<x,m,out> )

agent LB(in,m) = in(x) . (^ n) (( LB<in,n> | C<x,n,m> )  +  m(out) . ElasticBuf<in,out> )

agent C(x,n,m) = m(out) . RC<x,n,out>

agent RC(x,n,out) = 'out<x> . RCT<n,out>
agent RCT(n,out) = 'n<out> . 0
```

Exercício 5

Experimente simular este processo no MWB. Explique a razão para o qualificativo 'elástico' usado acima e mostre que este comportamento *não* pode ser expresso em CCS.

Suponha que, ao introduzir o exemplo, se enganou e escreveu

```
agent LB(in,m) = in(x) . (^ m) ( LB<in,m> | RC<x,m,out> )  +  m(out) . ElasticBuf<in,out>
```

Interprete as mensagens de erro reportadas pela ferramenta (há, pelo menos, dois erros detectáveis!).

---

2 Estruturas de Dados Distribuídas

Nas aulas teóricas foi já abordada a representação de estruturas de dados através de processos, com o exemplo muito simples dos valores booleanos. Mesmo antes de o estudo do π-*calculus*, abordamos a modelação de sequências em que cada valor era guardado num processo autónomo (ver Lição 5). O π-*calculus*, porém, oferece potencialidades bem mais poderosas, como, de resto, o exemplo do *buffer* 'elástico' acima evidencia.

Consideremos, agora, alguns casos mais.

2.1 De $\mathbb{B}$ a $\mathbb{N}$

O esquema de representação dos valores booleanos discutido nas aulas é facilmente adaptável a qualquer tipo definido por enumeração. Por exemplo, cada valor do tipo enumerado `Cores`, que em HASKELL se declararia

```
data Cores = Azul | Verde | Rosa | Castanho
```

origina um processo próprio similar ao que se fez para true e false. Assim,

$$Azul(e) \triangleq !\, e(a,v,r,c) \cdot \overline{a} \quad \text{ou} \quad Castanho(e) \triangleq !\, e(a,v,r,c) \cdot \overline{c}$$

Exercício 6

Comece por experimentar no MWB as representações que estudou para os valores booleanos e para este tipo enumerado. Qual o principal problema de representação que necessita controlar?

Pense, agora, no problema seguinte: os números naturais $\mathbb{N}$ podem ser vistos como um tipo enumerado em que cada número é entendido como um construtor próprio. Seguindo o método adoptado até aqui as representações de, por exemplo, 0, 3 ou 5 seriam, respectivamente,

$$\begin{aligned} 0(e) &\triangleq !\, e(z,d) \cdot \overline{z} \\ 3(e) &\triangleq !\, e(z,d) \cdot \overline{d} \cdot \overline{d} \cdot \overline{d} \cdot \overline{z} \\ 5(e) &\triangleq !\, e(z,d) \cdot \overline{d} \cdot \overline{d} \cdot \overline{d} \cdot \overline{d} \cdot \overline{d} \cdot \overline{z} \end{aligned}$$

Pretende-se verificar que a definição seguinte representa a função *sucessor* sobre $\mathbb{N}$:

$$Succ(e,n) \triangleq !\, e(z,d) \cdot \overline{d} \cdot \overline{n}\langle z,d\rangle$$

Para isso mostre que

$$4\langle a\rangle \approx \mathsf{new}\, x\ (Succ\langle a,x\rangle \mid 3\langle x\rangle \tag{1}$$

recorrendo, primeiro, ao MWB e, depois, apenas a papel e lápis.

---

a seguinte implementação de sequências ni π-*calculus*:

2.2 Representação de Sequências

Recorde a declaração do tipo de dados *sequências de x* em HASKELL:

```
data Seq x = Nil | Cons (x , Seq x)
```

Vejamos como representar sequências de nomes (que outra coisa não temos no π-*calculus*...) por dois processos correspondentes aos seus construtores. Assim,

$$\begin{aligned} Nil(k) &\triangleq k(n,c) \cdot \overline{n} \\ Cons(k,v,l) &\triangleq k(n,c) \cdot \overline{c}\langle v,l\rangle \end{aligned}$$

O processo $Nil(k)$ denota a lista vazia localizada no endereço (ou acessível via ligação) k. Por outro lado, $Cons\langle k,v,l\rangle$ é entendido como um processo, acessível via k, que gere a agregação de um processo que representa a cabeça da lista (endereçável em v) com processo que detém o acesso à lista que constitui a cauda (endereçável em l). Assim, a expressão

$$\mathsf{new}\ \{v,l\}\ Cons\langle k,v,l\rangle \mid H\langle v\rangle \mid T\langle l\rangle$$

representa a construção de uma sequência a partir de um valor gerido pelo processo H e de outra sequência gerida por T.

Exercício 7

Desenrole, por expansão das definições, as expressões seguintes

$$\mathsf{new}\ \{v,l\}\ Cons\langle k,v,l\rangle \mid 5\langle v\rangle \mid Nil\langle l\rangle$$
$$\mathsf{new}\ \{v,l\}\ Cons\langle k,v,l\rangle \mid 5\langle v\rangle \mid (\mathsf{new}\ \{v1,l1\}\ Cons\langle k1,v1,l1\rangle \mid 2\langle v1\rangle \mid Nil\langle l1\rangle)$$

Recorra ao MWB para ganhar confiança neste tipo de definições.

Considere, agora, a seguinte definição de um processo que copia uma sequência acessível através do nome l para outra acessível de m:

$$Copy(l,m) \triangleq \mathsf{new}\ \{n,c\}\ \overline{k}\langle n,c\rangle \cdot (n \cdot Nil\langle m\rangle + c(v,l') \cdot (\mathsf{new}\ m'\ (Cons\langle m,v,m'\rangle \mid Copy\langle l',m'\rangle)))$$

Note que, quando composto em paralelo com um processo acessível em l que represente uma lista, o processo $Copy\langle l,m\rangle$ construirá uma cópia da mesma lista, mas agora acessível via m. Ao percorrer a lista para a copiar, a lista original $L\langle l\rangle$ é destruida (porquê?).

Exercício 8

Comprove a sua intuição, recorrendo ao simulador de processos do MWB para converter

$$Nil\langle l\rangle \mid Copy\langle l,m\rangle$$

em $Nil\langle m\rangle$. Faça o mesmo mas substituindo Nil por um processo L que represente uma lista não vazia.

Ainda recorrendo ao MWB, verifique que, para qualquer processo $L(l)$ representando uma sequência de nomes, se tem

$$L\langle l\rangle \mid Copy\langle l,m\rangle \approx L\langle m\rangle \tag{2}$$

Repita a prova manualmente.

Exercício 9

No contexto da questão anterior, e supondo que tanto $L(l)$ como $Q(l)$ representam sequências de nomes, indique, auxiliado pelo MWB, *todas* as sequências de transições por τ da expressão

$$L\langle l\rangle \mid Q\langle l\rangle \mid Copy\langle l, m\rangle \tag{3}$$

Será possível controlar o não determinismo emergente? Será desejavel?

Exercício 10

A representação de sequências aqui discutida é volátil: cada lista é consumida ao ser percorrida. Altere a especificação de forma a introduzir permanência dos dados.

Exercício 11

Faça uma estudo similar da representação em π-*calculus*

1. Dos números naturais definidos pelos construtores $Zero$ e $Succ$;
2. Do tipo de dados *árvore binária*.
