TP1 - Hint: Interacção e Concorrência - 2019-20

Luís Soares Barbosa

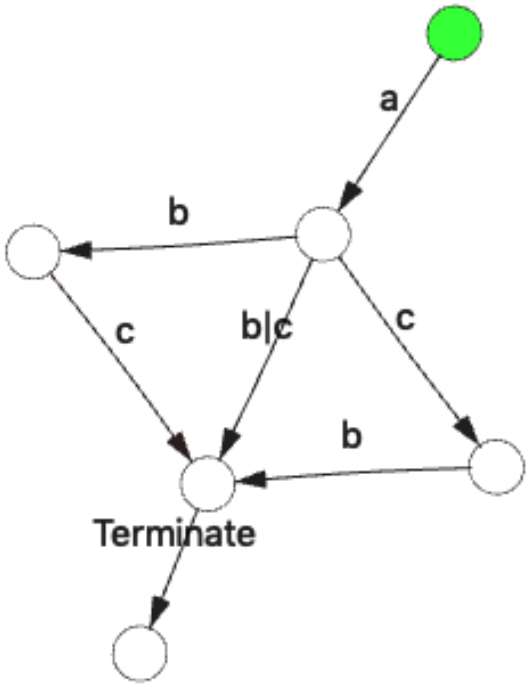
1 O problema

Como vários grupos perceberam, o o linearizador do mCRL2 não permite processos fora da *forma concorrente normal*. Isto significa, em particular, que não é possível aparecerem conectivos *estáticos* no escopo de conectivos *estáticos* (prefixo, escolha, recursividade, ...). Trata-se de uma limitação da implementação do mCRL2 razoavelmente documentada nos textos oficiais da ferramenta.

2 Tem solução?

No caso geral, não. Mas em algumas situações podemos recorrer a soluções mais engenhosas para obter o efeito pretendido. Vejamos duas situações:

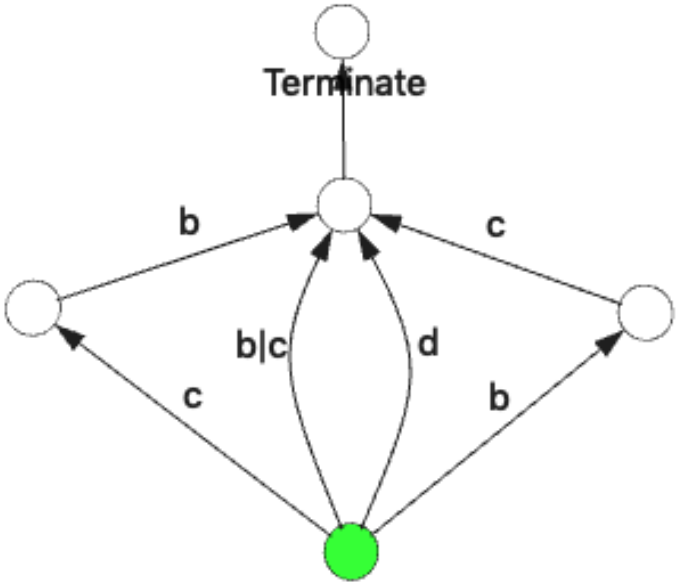
- O processo `a.(b||c)` não está na *forma concorrente normal* e por isso a ferramenta dá erro quando tenta linearizar. No entanto, o grafo que gostaríamos de obter seria:



e este pode ser obtido pela especificação seguinte (com recurso a variáveis dummy):

```
Z = block(d1,d2,comm(d1|d2 -> a,(d1.b)||(d2.c)))
```

- O mesmo se passa para o processo `(b||c)+d`, cujo grafo seria



... que foi obtido com

```
Z = block(d1,comm(d1|d1->d,(b+d1)||(c+d1)))
```

3 Este tipo de soluções aplicam-se ao processo C do enunciado?

Não, dada a presença de recursividade que vota a misturar operadores estáticos e dinâmicos. Por exemplo, o processo

```
Z = block(d1,comm(d1|d1->d,(b.Z+d1)||(c.Z+d1)))
```

já não lineariza.

4 O que fazer?

O meu primeiro objectivo era que o problema fosse detectado e encontradas soluções mesmo limitadas para o resolver parcialmente. As dadas acima já não contam ... mas se alguém tiver outras deve reporta-las.

Não será possível prototipar C^n em mCRL2 na sua versão mais geral. No entanto, recorrendo a técnicas como as acima descritas, talvez se possa fazer uma prototipagem parcial (i.e. de casos de tamanho fixo ...). Em todo o caso, o mCRL2 pode e deve ser usado sobre o processo sequencial que faz a especificação da versão concorrente dinâmica.