

Projectos Práticos

Elementos Lógicos da Programação III

2003/2004

Resumo

Apresentam-se as propostas para os projectos práticos da disciplina de Elementos Lógicos da Programação III. Os alunos interessados devem comunicar ao docente da disciplina a escolha do projecto. A data limite para a entrega e discussão é a do exame da disciplina.

1 Introdução

Cada projecto é composto de duas partes:

- Um exercício de modelização (comum a todos os projectos) onde se deve definir uma rede de Petri que modele o problema descrito.
- A exploração de uma ferramenta para a animação/análise da rede definida no ponto anterior.

2 Descrição do Problema: *Locks hierárquicos*

Começemos por uma versão simplificada do problema.

Admita a existência de uma *base de dados* que contém unicamente dois *registos*. Dois clientes vão necessitar de aceder a essa bases de dados. Os acessos são regulados por *locks* que devem satisfazer os seguintes requisitos:

- O acesso a cada um dos registos (por parte de qualquer dos clientes) requer exclusividade — um cliente a aceder a um registo impede que o outro possa aceder a esse registo;
- Qualquer um dos clientes pode necessitar de aceder à base de dados com exclusividade (e.g. para efectuar uma trasacção que envolva ambos os registos). Tal acesso só deverá ser possível se nenhum dos registos estiver *locked*, e impede qualquer outro acesso (à base de dados ou a cada um dos registos).

2.1 Valorização

2.1.1 *Locks* para Escrita e Leitura

Considere que se discriminam as operações de leitura e de escrita. As questões fundamentais relativas a essa distinção surgem resumidas nos seguintes pontos:

- As operações de leitura não impedem a realização de outras operações de leitura sobre o mesmo objecto, mas impedem operações de escrita.
- Operações de escrita impedem a realização de qualquer outra operação (escrita ou leitura).

Ajuste a rede para distinguir os tipos de *locks* para qualquer um dos níveis (registo ou base de dados).

2.1.2 Redes Coloridas

Pretende-se generalizar o problema proposto relaxando a restrição do número de registos e do número de clientes. Para tal devem-se utilizar redes de Petri coloridas (uma cor para o conjunto de registos, outra para o conjunto de clientes).

3 Ferramentas

Uma segunda fase do projecto consiste em explorar uma ferramenta de análise de redes de Petri para verificar propriedades do modelo construído. São ferramentas de domínio pública cuja documentação está disponíveis na WEB (ver apontadores na página da disciplina).

3.1 PEP tool

PEP (Programming Environment based on Petri Nets) é um conjunto de aplicações que permite manipular diferentes formatos de descrições de redes de Petri. Utilize o PEP para desenhar e animar a(s) rede(s) de Petri definida(s). Pode ainda combinar a utilização do PEP com outra ferramenta de análise (como o INA ou o SMV) já que o PEP disponibiliza conversão para as linguagens aceites por esses sistemas.

3.2 INA

Integrated Net Analyser é uma ferramenta que realiza um vasto conjunto de verificações estáticas sobre redes de petri (e.g. cálculo de invariantes ou análise de acessibilidade). Pretende-se utilizar essa ferramenta para verificar as propriedades de exclusão mútua requeridas pelo problema proposto.

Obs.: As versões mais recentes do INA incluem funcionalidades de *model checking* de formulas CTL. No âmbito deste projecto é suficiente utilizar o INA na sua vertente de “analisador estático”.

3.3 SMV

O *SMV model checker* realiza verificação de modelos finitos sobre uma descrição de um sistema concorrente. Uma vez expressas as propriedades pretendidas em CTL, o model checker procurará um contra-exemplo que refute essas propriedades (se não encontrar, as propriedades são válidas).

Obs.: A linguagem de entrada aceite pelo SMV não corresponde directamente às redes de Petri (trata-se antes de uma descrição de máquinas de estados finitos). Assim, a utilização desta ferramenta pressupõe um passo adicional que consiste em converter a rede na linguagem requerida pelo SMV. Sugere-se uma das alternativas para quem escolher explorar esta aplicação:

1. Considerar uma versão simples da rede (e.g. a descrição “base” do problema).
2. Utilizar a ferramenta PEP para efectuar a conversão (em rigor, acaba por realizar dois projectos, pelo que pode ser entendido como uma valorização).

3.4 Animador Haskell

Nas aulas teóricas, foram desenvolvidos/demonstrados alguns programas Haskell que permitem animar e analisar redes de Petri.

Pretende-se que utilize esses programas na(s) rede(s) de Petri definida(s) e que os estenda por forma a permitir realizar o cálculo dos invariantes mínimos dessa rede. Com base nesses invariantes deve estabelecer as propriedades de exclusão mútua requeridas.