

Processos e Arquiteturas de Software

Nuno Oliveira

26 janeiro de 2012, quinta-feira

1. Modelo de Cores – Uma Introdução Informal

Modelo semântico que usa cores (ou símbolos visuais) para descrever e determinar o fluxo que os dados tomam num circuito Reo.

O fluxo de dados depende do contexto de operações I/O nos nodos externos dos canais e (por composição) dos circuitos Reo.

Oferece uma semântica precisa para o fluxo de dados num circuito Reo.

Uma cor é um elemento do seguinte conjunto COR de 3 cores: { --- , -->-- , --<-- }

A cor ---- significa que há fluxo de dados no canal

As cores -->-- e --<-- significam que não há fluxo de dados no canal porque a ação I/O que se propaga na direção da seta não pode ser executada (por falta de pedidos, p.e.).

Estas cores “explicam” o motivo pelo qual não há fluxo ou uma ação I/O deve ser atrasada.

Coloração $c : N \rightarrow COR$, é uma função que mapeia uma cor para cada nodo de uma canal ou circuito.

Tabela de Colorações é uma tabela que contém todas as colorações que um canal ou circuito pode observar.

Observações (Resposta a perguntas levantadas na aula)

- A função de coloração efetivamente mapeia uma cor a cada nodo do canal. No entanto, quando as cores são iguais em ambos os nodos, nota-se todo o canal com apenas uma cor. Por exemplo, acontece este caso com o canal *Sync*, em qualquer uma das suas colorações, e.g., --->---- --->---- = ---->---
- As tabelas de coloração são sempre reduzidas ao menor número possível de colorações. Deste modo, cores repetidas são eliminadas tendo em conta a **regra Flip** que diz:

“Se uma tabela de coloração T tem uma entrada c que mapeia o nodo n à cor --->---- n , isto é, dando uma razão para atrasar ação em n , então

coloração c' , que é igual a c mas mapeia n à cor $---$, isto é, dando uma razão para não ocorrer o fluxo no canal, é redundante e pode ser eliminada de T ."

As colorações a eliminar numa tabela dependem ainda da composição que se pode fazer entre duas colorações. Ou, a um nível mais alto, entre duas tabelas.

Em resumo, duas colorações podem ser compostas para um nodo n (um nodo misto) seguindo a ideia de que ao nodo n (misto) tem que ser dado pelo menos uma razão para ele se atrasar nas suas ações. Este nodo **nunca** pode ser o motivo pelo qual há uma falha no fluxo. Então apenas as seguintes composições de colorações (de falha de fluxo) são aceites:

$--->--- n --->---$
 $---<--- n ---<---$
 $--->--- n --->---$

No entanto as colorações $---<--- n --->---$ não podem ser compostas, porque repare-se n (um nodo misto) não pode ser causador de atrasos ou falhas no fluxo.

RESUMO: uma coloração que dá razão para uma ação num nodo se atrasar ($--->--- n$) é sempre escolhida em detrimento de uma coloração cujo nodo é que dá a razão para não haver fluxo no canal ($---<--- n$).

Exemplos:

Synchronous Channel ($a--->b$)

i) $-----$ ii) $--->---$ iii) $---<---$

- i) Há ação I/O tanto em a como em b . Então $c(a) = ----$ e $c(b) = ----$. Deste modo a coloração será $a----- b$, ou usando a simplificação: $a----- b$.
- ii) Há ação I/O em b , mas não em a . Então $c(a) = --->---$ e (por propagação da razão) $c(b) = --->---$. Usando a mesma convenção, $a--->---$ b . Vejamos que no canal *Sync* se não há ação I/O no nodo source, o fluxo não ocorre e este nodo dá uma razão ao nodo sink para a ação aí presente ficar pendente.
- iii) Há ação I/O em a mas não em b . Então $c(b) = ---<---$ e (por propagação da razão) $c(a) = ---<---$. Logo $a---<---$ b . A explicação é semelhante à dada no ponto (ii).

Synchronous Drain Channel ($a>----<b$)

i) $-----$ ii) $--->---$ iii) $---<---$

- i) Há ação I/O em a e b . Então $c(a) = ----$ e $c(b) = ----$. Logo a coloração para o canal neste contexto é: $a----- b$.

- ii) Há ação I/O em b , mas não em a . Então $c(a) = \text{--->---}$ e (por propagação da razão) $c(b) = \text{--->---}$. Logo a coloração final é: $a \text{ --->--- } b$. Nota: a explicação da propagação segue do facto do canal ser síncrono e é semelhante à dada em (ii) do canal *Sync*.
- iii) Há ação I/O em a , mas não em b . Então $c(b) = \text{---<---}$ e (por propagação da razão) $c(a) = \text{---<---}$. Logo a coloração final é: $a \text{ ---<--- } b$. Nota: a explicação da propagação segue do facto do canal ser síncrono e é semelhante à dada em (ii) do canal *Sync*.

Lossy Synchronous Channel (a - - - - > b)

- i) ----- ii) ----- ---<--- iii) --->---
- i) Há ação I/O em a e b . Então $c(a) = \text{-----}$ e $c(b) = \text{-----}$. Logo a coloração final é: $a \text{ ----- } b$.
- ii) Há ação I/O em a mas não em b . Então $c(a) = \text{-----}$ e $c(b) = \text{---<---}$. Neste caso a coloração não pode ser resumida a uma única cor, portanto a coloração final é: $a \text{ ----- ---<--- } b$. Repare-se que este canal aceita sempre que informação seja escrita no seu nodo source (ao contrário do canal *Sync*) no entanto a informação é perdida quando não há ação em no nodo sink.
- iii) Há ação I/O em b mas não em a . Então $c(a) = \text{--->---}$ e (por propagação da razão) $c(b) = \text{--->---}$. Neste caso a coloração já pode ser resumida a uma única cor, portanto a coloração final é: $a \text{ --->--- } b$.

Fifo1 Channel (a --[]--> b)

- i) ----- --->--- ii) --->--- --->--- iii) ---<--- ---<--- iv) ---<--- -----
- i) Há ações I/O em ambas as pontas dos canais e o buffer está vazio. Então $c(a) = \text{-----}$ e $c(b) = \text{--->---}$. A ação de escrita em a ocorre porque o buffer está vazio, mas a ação de leitura em b não ocorre porque não é possível o buffer *escrever na segunda metade do canal* (a razão de a leitura não ocorrer propaga-se desde a sua origem que é: o buffer está vazio) Portanto a coloração final é a seguinte: $a \text{ ----- --->--- } b$.
- ii) Há ação I/O em b e o buffer está vazio. Então $c(a) = \text{--->---}$ e $c(b) = \text{--->---}$. Repare-se que neste contexto, não se dá a unificação da coloração pois a coloração em a deve-se ao facto de não haver ação I/O em a , no entanto a coloração em b já se deve ao facto de o buffer estar vazio. Logo não há propagação da razão mas sim razões diferentes que (por acaso) oferecem a mesma coloração. Nestes casos, e porque o modelo das cores oferece a semântica aos canais baseada em contexto e razões de atrasos no fluxo de dados, devem ser mantidas as duas cores. Logo a coloração final do canal

FIFO vazio, para este contexto será: $a \rightarrow \dots \rightarrow b$.

iii) Há ação I/O em a mas não em b e o buffer está cheio. Então $c(a) = \dots \leftarrow \dots$ e $c(b) = \dots \leftarrow \dots$. Novamente não pode ocorrer a unificação das cores porque as razões são diferentes, i.e., a razão não se propaga. A coloração para a deve-se ao facto de o buffer estar cheio, e portanto esta é uma razão para a ação em a ficar pendente. Em b , não há ação, e por esta razão, o buffer não pode *escrever na segunda metade do canal*. A coloração final para este contexto é: $a \dots \leftarrow \dots \leftarrow \dots b$

iv) Há ação I/O em a e b e o buffer está cheio. Então $c(a) = \dots \leftarrow \dots$ e $c(b) = \dots \leftarrow \dots$. Como o buffer está cheio, o fluxo de dados ocorre porque b tem ação I/O. A razão para o atraso da ação I/O em a deve-se ao facto de o buffer estar cheio, daí a cor usada. A coloração final para este contexto é, então, a seguinte: $a \dots \leftarrow \dots \leftarrow \dots b$.

NOTA

Para este caso do buffer podemos pensar em mais dois contextos: Buffer vazio, ação I/O em a mas não em b . Ou Buffer cheio, ação I/O em b mas não em a . *Vejamos para o primeiro caso, o segundo é semelhante.*

Neste caso $c(a) = \dots \leftarrow \dots$, claramente porque o buffer está vazio, e $c(b) = \dots \rightarrow \dots$ ou então $c(b) = \dots \leftarrow \dots$.

Aqui surge a dúvida. Claramente, o fluxo não ocorre porque não há ação I/O em b . No entanto também não poderia haver fluxo pois o buffer está vazio. Esta última é a razão mais geral. Mesmo que houvesse ação em b não poderia haver fluxo. No entanto a escolha da cor deve-se não ao facto da razão ser mais geral, mas sim ao facto relacionado com a composição de cores visto em RESUMO (acima).

Para mais informação acerca deste tema, consultar a bibliografia:

- Dave Clarke, David Costa, and Farhad Arbab. Connector colouring I: Synchronisation and context dependency. *Electron. Notes Theor. Comput. Sci.*, 154:101-119, May 2006.
- David Costa. *Formal Models for Component Connectors*. PhD thesis, Vrije University, Amsterdam, October 2010.

2. Exercícios com Padrões de Business Process

Crie circuitos Reo que tenham o mesmo comportamento semântico que os padrões de busines process descritos de seguida.

a. Parallel Split:

The divergence of a branch into two or more parallel branches each of which execute concurrently;

e.g.: Once the customer has paid for the goods, issue a receipt and pack them for dispatch

b. Simple Merger:

The convergence of two or more branches into a single subsequent branch. Each enablement of an incoming branch results in the thread of control being passed to the subsequent branch.

e.g.: After the cash-payment or provide-credit activities, initiate the produce-receipt activity

c. Sequencer

An activity in a workflow process is enabled after the completion of a preceding activity in the same process.

e.g.: A receipt is printed after the train ticket is issued.

d. Synchronizer

The convergence of two or more branches into a single subsequent branch such that the thread of control is passed to the subsequent branch when all input branches have been enabled.

e.g.: The dispatch-goods activity runs immediately after both the check-invoice and produce-invoice activities are completed.

e. Exclusive Router

The divergence of a branch into two or more branches. When the incoming branch is enabled, the thread of control is immediately passed to precisely one of the outgoing branches based on the outcome of a logical expression associated with the branch.

e.g.: After the review election activity is complete, either the declare results

or the recount votes activity is undertaken.

f. Exclusive Router – Non-deterministic?

g. Sequencer – Wait for acknowledgement from the previous task?