



Universidade do Minho
Escola de Ciências

Interação e Concorrência (3^o ano de LCC)
Trabalho Prático
Sistemas reativos

Pedro Manuel Pereira dos Santos (A100110)

3 de junho de 2025

Conteúdo

1	Introdução	3
2	Problema 1	4
2.1	Exercício 1	4
2.2	Exercício 2	4
3	Problema 2	6
3.1	Exercício 1	6
3.2	Exercício 2	7
4	Conclusão	10

Lista de Figuras

3.1	Diagrama de sincronização de C	7
3.2	Código do processo C em mCRL2	7
3.3	Diagrama gerado pelo mCRL2	8
3.4	Opção de simulação no mCRL2	9
3.5	Opção de diagrama reduzido no mCRL2	9
3.6	Opções de redução do diagrama no mCRL2	9

Capítulo 1

Introdução

No âmbito da unidade curricular de Interação e Concorrência da Licenciatura em Ciências da Computação, foram propostos vários exercícios relacionados a sistemas reativos. De seguida, temos a resolução dos exercícios propostos.

Capítulo 2

Problema 1

Este problema aborda o conceito de *palavra bissimilar* e pede-nos que o exploremos.

2.1 Exercício 1

Para definir formalmente a relação \xrightarrow{s} para $s \in N^*$, consideremos que s é uma sequência de ações (palavra) em N^* . A relação \xrightarrow{s} é definida recursivamente da seguinte forma:

Caso Base (palavra vazia ε):

$$p \xrightarrow{\varepsilon} p \quad (\text{para qualquer estado } p)$$

Caso Indutivo (para $s = a \cdot s'$ onde $a \in N$, $s' \in N^*$):

$$p \xrightarrow{a \cdot s'} p' \quad \text{se e somente se} \quad \exists p'' \text{ tal que } p \xrightarrow{a} p'' \text{ e } p'' \xrightarrow{s'} p'.$$

Sendo assim, temos a seguinte definição formal:

$$\xrightarrow{s} = \left\{ (p, p') \in S \times S \mid \exists n \geq 0, \exists p_0, \dots, p_n \text{ tais que } p_0 = p, p_n = p', \text{ e } p_i \xrightarrow{a_{i+1}} p_{i+1} \text{ para } 0 \leq i < n \right\},$$

considerando que $s = a_1 a_2 \dots a_n$ é a sequência de ações que leva de p a p' .

2.2 Exercício 2

Queremos demonstrar a equivalência:

$$p \sim q \Leftrightarrow p \text{ e } q \text{ são palavras bissimilares}$$

(\Rightarrow) **Bisimilaridade forte implica palavras bissimilares**

Assuma que $p \sim q$. Então existe uma bisimulação forte R com (p, q) . Mostremos que R é também uma palavra bissimilar:

1. **Definição de \xrightarrow{s} :** Para $s = a_1 \dots a_n \in N^*$, a relação $p \xrightarrow{s} p'$ significa que existe uma sequência:

$$p \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} p'$$

2. Prova por indução sobre $|s|$:

- **Caso base** ($n = 1$): Segue diretamente da definição de bisimulação forte.
- **Método indutivo**: Para $p \xrightarrow{a_1} p_1 \xrightarrow{a_2 \dots a_n} p'$, como R é bissimulação forte, existe q_1 tal que:

$$q \xrightarrow{a_1} q_1 \quad \text{com} \quad (p_1, q_1) \in R$$

Pelo método de indução, existe q' com:

$$q_1 \xrightarrow{a_2 \dots a_n} q' \quad \text{e} \quad (p', q') \in R$$

(\Leftarrow) **Palavras bissimilares implica bisimilaridade forte**

Assuma que p e q são palavras bissimilares R com $(p, q) \in R$. Mostramos que R é bisimulação forte:

- Para qualquer relação base $a \in N$ (caso $|s| = 1$):
 - Se $p \xrightarrow{a} p'$, como R é palavra bissimilar (para $s = a$), existe q' tal que:

$$q \xrightarrow{a} q' \quad \text{e} \quad (p', q') \in R$$

- E analogamente o mesmo acontece para transições a partir de q .

Portanto, temos que:

$$p \sim q \text{ se e somente se } p \text{ e } q \text{ são palavras bissimilares}$$

Capítulo 3

Problema 2

Este problema demonstra um sistema de controle de um cruzamento ferroviário. O sistema inclui processos para carros (Road), comboios (Rail) e um semáforo (Signal), com eventos como car, train, up/dw e green/red. O processo global C sincroniza estas componentes.

O objetivo é analisar o seu comportamento e representar as interações num diagrama e, posteriormente, aplicar isso em mCRL2.

3.1 Exercício 1

Este processo inicialmente pode detectar a chegada de um carro ou de um comboio. O controlador (Signal) permanece em um estado de espera até que ocorra um evento de up ou green, ativando depois a sequência de operações mencionadas a seguir.

Primeiramente, quando o processo do carro é iniciado (Road) e assume a entrada de carros (car), é levantada a barra de proteção (up) - ação sincronizada entre o processo Road e Signal. O carro avança então pelos trilhos, registrando a sua passagem através do sensor ccross. Após a passagem, é enviado um comando para baixar a barra (dw), completando assim o ciclo rodoviário e devolvendo o sistema ao estado inicial.

No caso do comboio, quando este processo é iniciado (Rail) e assume a chegada de um comboio (train), o Signal ativa o sinal verde (green) através da ação sincronizada entre o processo Train e Signal, para autorizar a passagem. Quando o comboio atravessa o sensor tcross, é enviada uma mensagem para mudar os semáforos para vermelho (red), garantindo que o cruzamento fica seguro antes de qualquer possível interação com o tráfego rodoviário. Esta sincronização garante que nunca ocorrem situações de conflito onde ambos os meios de transporte poderiam estar simultaneamente no cruzamento.

Caso tenhamos uma situação em que um carro e um comboio venham ao mesmo tempo, o primeiro processo que for ativado avança, enquanto o outro espera para eventualmente começar o seu processo após o primeiro acabar e o Signal voltar ao seu estado inicial.

Finalmente, a seguir apresento um diagrama de sincronização desenhado correspondente ao processo C :

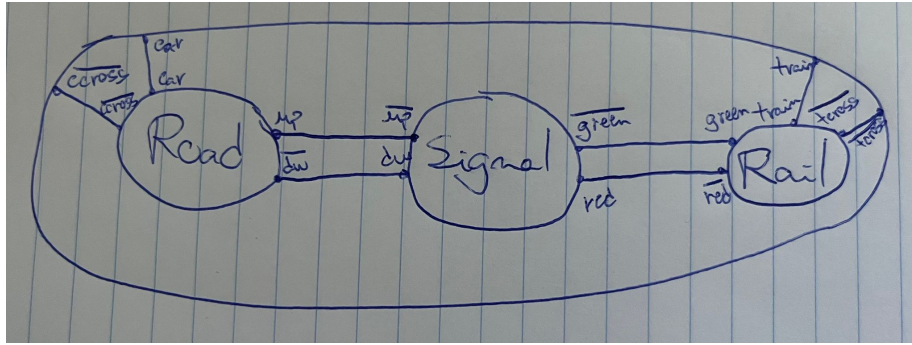


Figura 3.1: Diagrama de sincronização de C

3.2 Exercício 2

Para traduzir o processo C para o mCRL2 com [1], temos algo semelhante ao que está no enunciado, mantemos os processos e o nome das variáveis iguais, sendo apenas necessário atribuir novas variáveis para as transições em sincronização (para restringir as transições apenas ao estado de sincronização), que são:

- **sgreen** (sincronização entre o **green** de Rail e **green** de Signal)
- **sred** (sincronização entre o **red** de Rail e **red** de Signal)
- **_sup** (sincronização entre o **up** de Road e **up** de Signal)
- **sdw** (sincronização entre o **dw** de Road e **dw** de Signal)

Com isto, obtemos o código em mCRL2 [2]:

```

1 act car, train, up, dw, ccross, tcross, green, red,
2 sgreen, sred, _sup, sdw;
3
4 proc Road = car . up . ccross . dw . Road;
5      Rail = train . green . tcross . red . Rail;
6      Signal = green . red . Signal + up . dw . Signal;
7
8      C = allow({car,ccross, train, tcross, sgreen, sred, _sup, sdw},
9      comm({green|green -> sgreen,
10          red|red -> sred,
11          up|up -> _sup,
12          dw|dw -> sdw},
13      Road || Rail || Signal));
14
15 init C;
16

```

Figura 3.2: Código do processo C em mCRL2

Ao executar a opção de representar o diagrama, obtemos o seguinte:

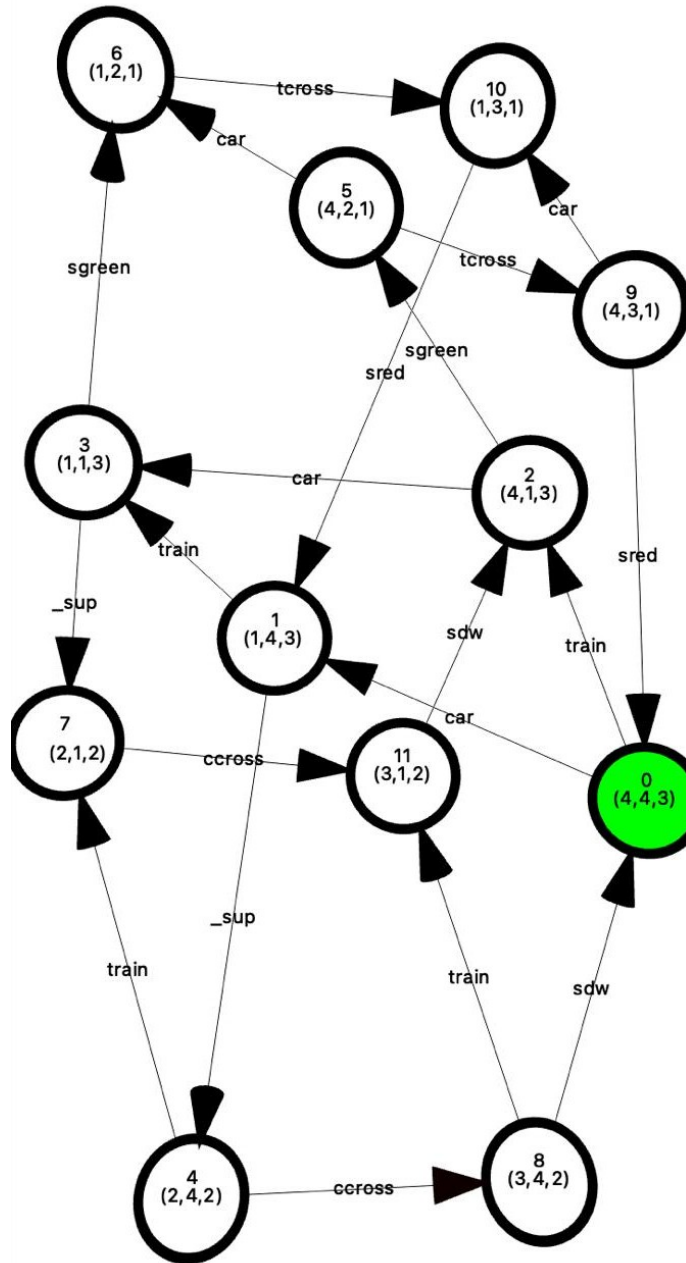


Figura 3.3: Diagrama gerado pelo mCRL2

Com este grafo, podemos analisar em que ação os 3 processos podem estar simultaneamente e que ações podem realizar a partir desse estado.

Por exemplo, podemos ver que em $(4,4,3)$ todos estão na ação inicial, e após a ação car vai para $(1,4,3)$, e etc.

Também podemos ver que, quando há uma ação de sincronização (sgreen, sred, _sup ou sdw), obrigatoriamente o signal tem de estar disponível para realizar essa ação. Caso contrário, é impossível fazê-lo.

Para analisar melhor o comportamento do processo C , o mCRL2 disponibiliza uma opção de simulação com o seguinte símbolo:



Figura 3.4: Opção de simulação no mCRL2

Nesta opção, é possível simular o processo C realizando ações à nossa escolha. Resumidamente, pode aparecer um `car` ou um `train`, mas o que escolhermos sinalizar primeiro ao `Signal`, este avança, fazendo o outro esperar. Para o funcionamento do resto do processo, a descrição no exercício anterior é igual à que seria descrita aqui.

Por fim, existe mais uma opção de representar o diagrama com o seguinte símbolo:

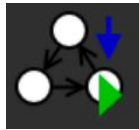


Figura 3.5: Opção de diagrama reduzido no mCRL2

Isto gera um gráfico semelhante ao da Figura 3.3, mas com a possibilidade de escolher reduções, sendo elas:

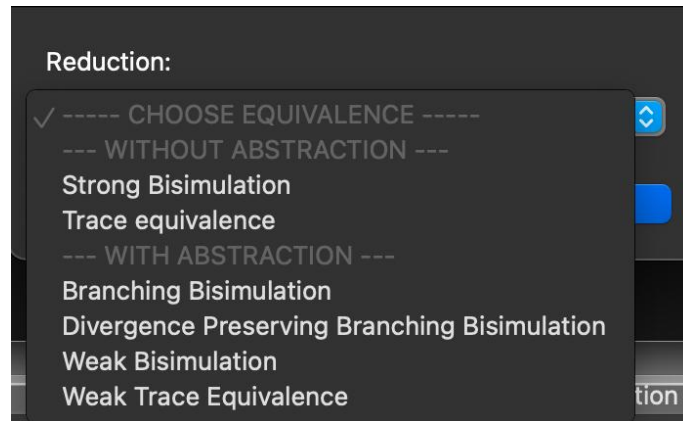


Figura 3.6: Opções de redução do diagrama no mCRL2

Capítulo 4

Conclusão

Este trabalho proporcionou um aprofundamento dos conhecimentos em sistemas reativos, explorando conceitos fundamentais como a bissimulação de palavras e a modelação de sistemas concorrentes utilizando a ferramenta MCRL2.

Através da resolução dos problemas, consolidou-se a compreensão sobre equivalência comportamental entre estados em sistemas de transição, bem como a importância da sincronização e coordenação de processos em sistemas críticos, como o controle de um cruzamento entre estrada e linha férrea.

A utilização do MCRL2 revelou-se uma ferramenta valiosa para análise formal, permitindo visualizar o sistema e verificar propriedades de forma rigorosa.

Estas experiências reforçaram a relevância dos métodos formais no desenvolvimento de sistemas fiáveis e seguros, destacando a sua aplicação prática em contextos reais.

Referências

- [1] Technische Universiteit Eindhoven. mCRL2 user manual, 2025. URL: https://mcr12.org/web/user_manual/language_reference/index.html.
- [2] Technische Universiteit Eindhoven. mCRL2: A formal specification tool, 2025. URL: <https://mcr12.org>.