

Diferença entre Templates de Autômatos Celulares Unidimensionais Binários

Zorandir Soares Jr.
zorandir@gmail.com

Universidade Presbiteriana Mackenzie
Programa de Pós-Graduação em Engenharia Elétrica e Computação

Orientador: Prof. Dr. Pedro Paulo Balbi de Oliveira

16 de agosto de 2016



Sumário

Objetivos

Autômatos Celulares

Templates

Diferença entre Templates

Discussões e Testes

Conclusão



Objetivos

- ▶ Apresentar a operação de diferença entre templates
- ▶ Apresentar a operação geradora de templates de exceção
- ▶ Apresentar exemplos da utilização dos templates no problema de paridade



Autômatos Celulares



Figura 1 : Tapetes expostos em uma exposição de arte realizada na “Maison Salvan”, em Carjac, França, em junho de 2008. Eles foram criados com o simulador FiatLux CA.



Famílias de autômatos celulares

Uma família (ou espaço) de autômatos celulares é definida pelo raio r e pelo número de estados k .

O tamanho de uma família é definido pela expressão abaixo:

$$k^{k^{2r+1}} \quad (1)$$



Propriedades Estáticas

- ▶ Confinamento
- ▶ Simetria Interna Máxima
- ▶ Simetria Interna Arbitrária
- ▶ Totalidade e Semi-totalidade
- ▶ Conservabilidade da soma de estados
- ▶ Conservabilidade da soma modular de estados



Templates

Template é uma generalização de tabelas de transições de ACs.

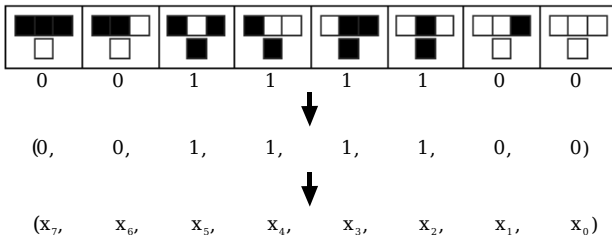


Figura 2 : Exemplo de tabelas de transições



Expansão

Expansão é o processo no qual se obtêm todas as tabelas de transição R_k associadas a um template T .

$$E(T) = R_k \quad (2)$$



Exemplo - Expansão

Tabela 1 : Processo de expansão do template $(1, 1, 1, 1, 1, x_0 + x_1, x_1, x_0)$

i	x_1	x_0	tabela k -ária resultante
0	0	0	$(1, 1, 1, 1, 1, 0, 0, 0)$
1	0	1	$(1, 1, 1, 1, 1, 1, 0, 1)$
2	1	0	$(1, 1, 1, 1, 1, 1, 1, 0)$
3	1	1	$(1, 1, 1, 1, 1, 2, 1, 1)$



Intersecção

$$I(T_1, T_2) = T_3 \Leftrightarrow E(T_3) = E(T_1) \cap E(T_2) \quad (3)$$

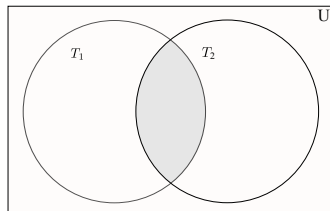


Figura 3 : Os círculos T_1 e T_2 são templates que representam dois conjuntos de regras. Em cinza, T_3 é o template que representa o conjunto de regras de intersecção entre T_1 e T_2 .



Geradoras de Templates e Número de Estados

Tabela 2 : Compatibilidade entre algoritmos geradores de templates e número de estados

Algoritmos Geradores de Templates	Número de Estados	
	$k = 2$	$k > 2$
Totalidade e Semi-Totalidade	●	●
Confinamento	●	●
Color Blind	●	● ¹
Simetria Máxima	●	●
Simetria Arbitrária	●	
Conservabilidade da soma de estados	●	●
Conservabilidade da soma modular de estados	● ¹	

¹Template Modular



Geradoras de Templates e Pós-processamento

Tabela 3 : Pós-processamento necessário por templates

Template	Pós-processamento
Template de regras confinadas	Filtrar regras com restrições inválidas
Template de conservabilidade de estados	Filtrar regras inválidas
Template de conservabilidade de paridade	Aplicar $\text{mod } 2$ e filtrar regras inválidas
Template de totalidade e semi-totalidade	Nenhum pós-processamento necessário
Templates de simetria	Nenhum pós-processamento necessário

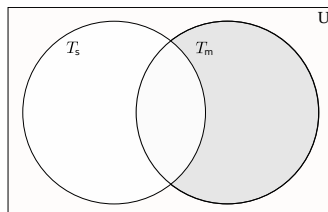


Diferença entre Templates

- ▶ Diferença entre Templates
- ▶ Templates de exceção



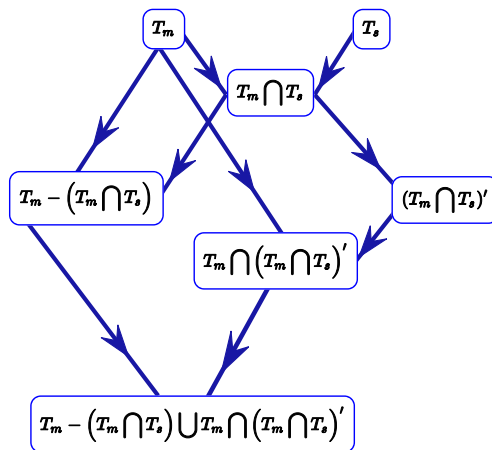
Operação de Diferença entre Templates Binários



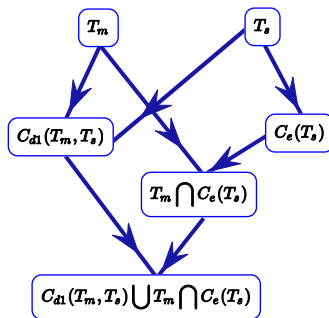
$$\begin{aligned} D_i(T_m, T_s) = C_{di} &\Leftrightarrow E(C_{di}) = E(T_m) \setminus E(T_i) \\ T_i &= I(T_m, T_s) \\ C_{di} &= \{T_1, T_2, \dots, T_n\} \end{aligned} \quad (4)$$



Operação de Diferença entre Templates Binários



Operação de Diferença entre Templates Binários



Exemplo - Operação de Diferença entre Templates

Considere:

$$T_m = (x_7, x_6, x_5, x_4, x_3, x_1, x_1, x_0)$$

$$T_s = (x_7, x_6, x_3 + x_1, 1 - x_1, x_3, x_1, x_1, 0)$$

$$\left\{ \begin{array}{lcl} x_7 & = & x_7 \\ x_6 & = & 0 \\ x_5 & = & x_3 + x_1 \\ x_4 & = & 1 - x_1 \\ x_3 & = & x_3 \\ x_1 & = & x_1 \\ x_1 & = & x_1 \\ x_0 & = & x_0 \end{array} \right. \quad (5)$$



Exemplo - Operação de Diferença entre Templates

Considere:

$$T_m = (x_7, x_6, x_5, x_4, x_3, x_1, x_1, x_0)$$

$$T_s = (x_7, x_6, x_3 + x_1, 1 - x_1, x_3, x_1, x_1, 0)$$

$$\begin{array}{rcl} x_7 & = & x_7 \quad \wedge \\ x_6 & = & 0 \quad \wedge \\ x_5 & = & x_3 + x_1 \quad \wedge \\ x_4 & = & 1 - x_1 \quad \wedge \\ x_3 & = & x_3 \quad \wedge \\ x_1 & = & x_1 \quad \wedge \\ x_1 & = & x_1 \quad \wedge \\ x_0 & = & x_0 \end{array} \quad (6)$$



Exemplo - Operação de Diferença entre Templates

Considere:

$$T_m = (x_7, x_6, x_5, x_4, x_3, x_1, x_1, x_0)$$

$$T_s = (x_7, x_6, x_3 + x_1, 1 - x_1, x_3, x_1, x_1, 0)$$

$$\begin{aligned} x_6 &= 0 \quad \wedge \\ x_5 &= x_3 + x_1 \quad \wedge \\ x_4 &= 1 - x_1 \end{aligned} \tag{7}$$



Exemplo - Operação de Diferença entre Templates

Considere:

$$T_m = (x_7, x_6, x_5, x_4, x_3, x_1, x_1, x_0)$$

$$T_s = (x_7, x_6, x_3 + x_1, 1 - x_1, x_3, x_1, x_1, 0)$$

$$\begin{aligned} x_6 &= 1 - 0 & \vee \\ x_5 &= 1 - (x_3 + x_1) & \vee \\ x_4 &= 1 - (1 - x_1) \end{aligned} \tag{8}$$



Exemplo - Operação de Diferença entre Templates

Considere:

$$T_m = (x_7, x_6, x_5, x_4, x_3, x_1, x_1, x_0)$$

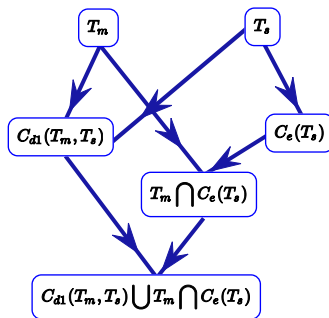
$$T_s = (x_7, 0, x_3 + x_1, 1 - x_1, x_3, x_1, x_1, x_0)$$

$$S = \{\{x_6 \rightarrow 1\}, \{x_4 \rightarrow x_1\}, \{x_5 \rightarrow 1 - x_1 - x_3\}\} \quad (9)$$

$$C_{d1} = \{ \begin{aligned} &(x_7, 1, x_5, x_4, x_3, x_1, x_1, x_0), \\ &(x_7, x_6, x_5, x_1, x_3, x_1, x_1, x_0), \\ &(x_7, x_6, 1 - x_1 - x_3, x_4, x_3, x_1, x_1, x_0) \end{aligned} \} \quad (10)$$



Operação de Diferença entre Templates Binários



Templates de exceção

$$\begin{aligned} X(T_o) &= C_e \\ C_e &= \{T_1, T_2, \dots, T_n\} \end{aligned} \tag{11}$$



Exemplo - Templates de exceção

Considere:

$$T_s = (x_7, 0, x_3 + x_1, 1 - x_1, x_3, x_1, x_1, x_0)$$

O primeiro passo do algoritmo é encontrar um conjunto com as posições que não apresentem apenas constantes ou variáveis livres, obtendo-se assim $\{\{x_3 + x_1\}, \{1 - x_1\}\}$.



Templates de exceção

Considere:

$$T_s = (x_7, 0, x_3 + x_1, 1 - x_1, x_3, x_1, x_1, x_0)$$

Tabela 4 : Expansão do campo

$x_3 + x_1$.

x_3	x_1	Expansão do campo
0	0	$x_3 + x_1 = 0$
0	1	$x_3 + x_1 = 1$
1	0	$x_3 + x_1 = 1$
1	1	$x_3 + x_1 = 2$

Tabela 5 : Expansão do campo 1

$- x_1$.

x_1	Expansão do campo
0	$1 - x_1 = 1$
1	$1 - x_1 = 0$



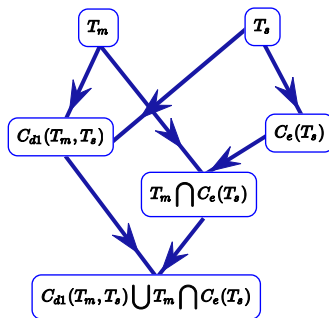
Templates de exceção

Para finalizar a operação, o algoritmo seleciona as substituições que geram valores inválidos, e aplicam ela ao template base, gerando assim o conjunto de templates C_e , mostrado na Eq. 12.

$$C_e = \{(x_7, x_6, x_5, x_4, 1, x_2, 1, x_0)\} \quad (12)$$



Operação de Diferença entre Templates Binários



Finalizando a Operação de diferença

$$C_e = \{(x_7, x_6, x_5, x_4, 1, x_2, 1, x_0)\}$$

$$C_{d2} = C_e \cap T_m$$

$$C_{di} = C_{d1} \cup C_{d2}$$

$$C_{di} = \{$$

$$(x_7, 1, x_5, x_4, x_3, x_1, x_1, x_0),$$

$$(x_7, x_6, x_5, x_1, x_3, x_1, x_1, x_0),$$

$$(x_7, x_6, 1 - x_1 - x_3, x_4, x_3, x_1, x_1, x_0),$$

$$(x_7, x_6, x_5, x_4, 1, 1, 1, x_0)\}$$
(13)



Discussões e Testes

- ▶ Número de templates gerados
- ▶ Aplicação no Problema de Paridade
- ▶ Metodologia de testes



Função de Teste

- ▶ A função recebe dois templates como parâmetro, um template minuendo e um subtraendo
- ▶ Expande as dois templates obtendo dois conjuntos de regras
- ▶ Realiza a diferença entre os dois conjuntos de regras
- ▶ Verifica se as regras obtidas são equivalentes a expansão dos templates obtidos pela operação de diferença



Testes

Tabela 6 : Propriedades estáticas utilizadas para o teste da operação de diferença em cada raio

Template subtraendo (T_s)	Template minuendo (T_m)				
	totalidade	semi-totalidade	conservabilidade de estados	invariância à troca de cor	confinamento
totalidade	Raio 1, 2 e 3	Raio 1, 2 e 3	Raio 1 e 2	Raio 1 e 2	Raio 1
semi-totalidade	Raio 1, 2 e 3	Raio 1, 2 e 3	Raio 1 e 2	Raio 1 e 2	Raio 1
conservabilidade de estados	Raio 1 e 2	Raio 1 e 2	Raio 1 e 2	Raio 1 e 2	Raio 1
invariância à troca de cor	Raio 1 e 2	Raio 1 e 2	Raio 1 e 2	Raio 1 e 2	Raio 1
confinamento	Raio 1	Raio 1	Raio 1	Raio 1	Raio 1



Exemplo: template de semi-totalidade - template de totalidade

$$\begin{aligned}
 & (x_{31}, x_{15}, x_{15}, x_7, x_{27}, x_{11}, x_{11}, x_3, x_{15}, x_7, x_7, x_5, x_{11}, x_3, x_3, x_1, x_{15}, x_7, x_7, x_5, x_{11}, x_3, x_3, x_1, x_7, x_5, x_5, x_4, x_3, x_1, x_1, x_0) - \\
 & (x_{31}, x_{15}, x_{15}, x_7, x_{27}, x_{11}, x_{11}, x_3, x_{15}, x_7, x_7, x_5, x_{11}, x_3, x_3, x_1, x_{15}, x_7, x_7, x_5, x_{11}, x_3, x_3, x_1, x_7, x_5, x_5, x_4, x_3, x_1, x_1, x_0) = \\
 & \{ (x_{31}, x_{15}, x_{15}, x_7, x_{27}, 1 - x_7, 1 - x_7, x_3, x_{15}, x_7, x_7, x_5, 1 - x_7, x_3, x_3, x_1, x_{15}, x_7, x_7, x_5, 1 - x_7, x_3, x_3, x_1, x_7, x_5, x_5, x_4, x_3, x_1, x_1, x_0), \\
 & (x_{31}, x_{15}, x_{15}, x_7, 1 - x_{15}, x_{11}, x_{11}, x_3, x_{15}, x_7, x_7, x_5, x_{11}, x_3, x_3, x_1, x_{15}, x_7, x_7, x_5, x_{11}, x_3, x_3, x_1, x_7, x_5, x_5, x_4, x_3, x_1, x_1, x_0), \\
 & (x_{31}, x_{15}, x_{15}, x_7, x_{27}, x_{11}, x_{11}, x_3, x_{15}, x_7, x_7, x_5, x_{11}, x_3, x_3, x_1, x_{15}, x_7, x_7, x_5, x_{11}, x_3, x_3, x_1, x_7, x_5, x_5, 1 - x_1, x_3, x_1, x_1, x_0), \\
 & (x_{31}, x_{15}, x_{15}, x_7, x_{27}, x_{11}, x_{11}, x_3, x_{15}, x_7, x_7, 1 - x_3, x_{11}, x_3, x_3, x_1, x_{15}, x_7, x_7, 1 - x_3, x_{11}, x_3, x_3, x_1, x_7, 1 - x_3, 1 - x_3, x_4, x_3, x_1, x_1, x_0) \}
 \end{aligned}$$

(14)



Problema de Paridade

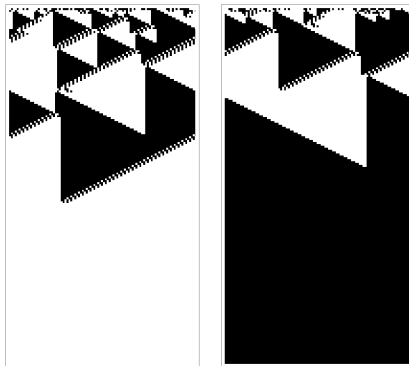


Figura 4 : Exemplo de regra de paridade. A imagem a esquerda contém em sua entrada um número par de 1s. A da direita contém um número ímpar.



Problema de Paridade

$$T_{paridade} = (T_{conservaparidade} \cap T_{confinado}) - T_{conservaestados} \quad (15)$$

A regra que solucione o problema de paridade:

- ▶ deve ser **confinada**
- ▶ deve **conservar a paridade**
- ▶ não deve **number conserving**



Problema de Paridade

Tamanho do espaço:

$$2^{2^{2 \times 3 + 1}} = 3,4 \times 10^{38} \quad (16)$$

Número máximo de regras conservativas de paridade:

$$2^{63} \approx 9.2 \times 10^{18} \quad (17)$$



Problema de Paridade - Pós-processamento

Template Core:

$$\{1, 1 + x_2 - x_3, 1 - x_2, 1 - x_1 - x_2, x_3, x_2, x_1, 0\} \quad (18)$$

Regra *number-conservative*
FilterOutOfRange

Regra conservativa de paridade
TemplateMod 2



Conclusão

- ▶ Diferença entre templates
- ▶ Templates de exceção
- ▶ Aplicação no problema de paridade



Agradecimentos

À Capes, CNPq, MackPesquisa e ao Laboratório de Computação Natural (LCoN).

