

Ferramentas de suporte ao estudo da Disciplina de Sistemas Digitais – Carta de Implicações

Sérgio Barbosa, Manuel Bernardo Cunha, Augusto Silva

Resumo – A Carta de Implicações é um algoritmo de simplificação de Máquinas de Estados Finitos e faz parte do conteúdo programático da disciplina de Sistemas Digitais leccionada no Departamento de Engenharia Electrónica, Telecomunicações e Informática.

O trabalho que aqui se apresenta resultou na conquista de uma ferramenta eficaz na fase de projecto de dispositivos sequenciais e útil no suporte à aprendizagem.

Desenvolvida para plataformas Microsoft, utiliza o Excel e o Visual Basic para Aplicações, como suporte e linguagem de programação respectivamente.

Ao código do algoritmo foi dada dupla função: utilizado na ferramenta aqui apresentada e serve na plataforma do Projecto Matemática Ensino como motor resolvente de exercícios.

No fim, espera-se que este documento consiga transmitir a ideia de que, é possível recriar cenários úteis à aprendizagem, a partir de ferramentas comuns.

Abstract – The Implication Chart is an algorithm used to simplify Finite State Machines which is a lectured part of the Digital Systems course, under the Engineering Electronics, Telecommunication and Informatics Department responsibility.

The result of all the work presented here is an efficient tool used at the project stage of sequential components and useful in the learning support.

Developed for Microsoft platforms, it uses Excel and Visual Basic for Applications, as the tool support and programming language respectively.

The programming code of the algorithm engine has been given double function: used in the tool presented here and integrates the Mathematical Education Project platform as an engine of exercises resolution.

At the end, we hope that this document will transmit the idea that, it's possible to build useful scenarios in the learning process, with common tools.

I. INTRODUÇÃO

Um dos tópicos do conteúdo programático da disciplina de Sistemas Digitais (SD) é Máquinas de Estados Finitos (MEF). No âmbito deste tema são ensinadas formas de representar o comportamento de uma máquina sequencial, quer através de tabelas de transições, quer por intermédio de diagramas de estados. O grande objectivo neste tipo de exercício é simplificar o número de estados redundantes, que ocorrem com maior frequência em sistemas de maior complexidade.

Existem formas de simplificar o número de estados, uma das quais (conhecida por “*row matching*”) resulta da análise das transições e valores de saída do próprio estado. Esta forma, apesar de simples e ajudar na simplificação, nem sempre conduz à melhor solução. Contudo, existe uma ferramenta que permite determinar o número mínimo de estados do sistema sequencial com grande rigor: Carta de Implicações.

Por ser uma ferramenta que assenta num processo iterativo, que integra um conjunto de regras bem definidas, surgiu a ideia de desenvolver um software o algoritmo de simplificação, a partir de uma tabela de transições de estados. A motivação inicial foi a de construir mais um motor gerador de exercícios que pudesse integrar a plataforma PmatE, mas rapidamente expandiu-se a sua utilização, construindo uma nova ferramenta de suporte ao estudo da disciplina de SD.

Nas secções seguintes são apresentados o enquadramento teórico e o modo de funcionamento da ferramenta.

II. APLICAÇÃO DA FERRAMENTA

A. Área científica

As MEF fazem parte de cursos de iniciação aos Sistemas Digitais, é um dos tópicos finais do programa, apresentado logo a seguir às unidades básicas de memória.

No desenho destas máquinas é essencial não só assegurar o funcionamento, como garantir que estas tenham o mesmo desempenho utilizando o menor número de recursos possível. Para que isto seja garantido é necessário uma inspecção da tabela de transições e recorrendo a ferramentas como a Carta de Implicações (CI) garantir o menor número de estados.

A relação entre o número de estados e o número de células de memória (*flip-flops*) é dada pela expressão **i)** em que N_e corresponde ao número de estados e n ao número de *flip-flops*. Assim, e contando que n apenas admite valores inteiros, n será igual ao segundo membro sempre que este pertencer ao conjunto dos naturais e aproximado ao menor dos inteiros acima do valor obtido, nos restantes casos.

$$i) \quad - N_e \leq 2^n; \ln_2 N_e \leq \ln_2 2^n; n \geq \ln_2 N_e$$

Posto isto, é de todo interesse na fase de projecto conseguir o menor valor para N_e , minimizando o valor de n .

Para se poder minimizar o número de estados de uma máquina é necessário encontrar equivalência entre eles. Dois estados dizem-se equivalentes se, para a mesma excitação, transitam para o mesmo estado com as mesmas saídas. Por outro lado, se transitam com saídas diferentes não são equivalentes. Existem casos, ainda, em que a decisão é remetida para o futuro, o que acontece quando para as mesmas entradas se obtêm as mesmas saídas mas os estados seguintes não são os mesmos.

A ferramenta que é capaz relacionar todos os estados e encontrar aqueles que são equivalentes é, como já foi referido, a CI. O ponto de partida, normalmente, é uma tabela de transições de estados, como mostra o exemplo da Figura 1.

Present State	Next State				Output
	00	01	10	11	
S ₀	S ₀	S ₁	S ₂	S ₃	1
S ₁	S ₀	S ₃	S ₁	S ₅	0
S ₂	S ₁	S ₃	S ₂	S ₄	1
S ₃	S ₁	S ₀	S ₄	S ₅	0
S ₄	S ₀	S ₁	S ₂	S ₅	1
S ₅	S ₁	S ₄	S ₀	S ₅	0

Figura 1 - Tabela de transições de estados

A CI utiliza uma estrutura tabular para definir todas as associações entre estados. Sempre que uma associação não representa um estado equivalente, é colocada na célula correspondente uma cruz.

A carta fica reduzida a um triângulo uma vez que a representação quadrangular contém dupla associação para cada par distinto de estados – o que não faz sentido – e a diagonal (ver Figura 2) representa a associação de um estado com ele próprio (desprezável).

Portanto, aquilo que se pretende descobrir são estados equivalentes e para o fazer vamos por exclusão de partes: passo a passo encontrar os que não são equivalentes e analisar o que resta.

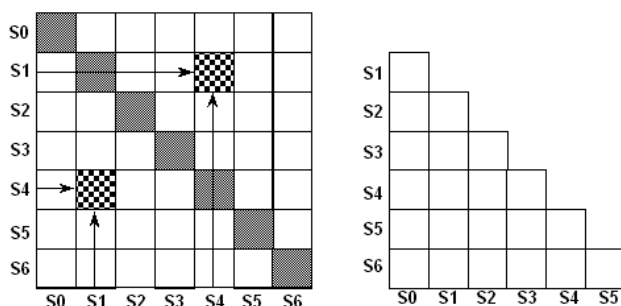


Figura 2 - Carta de Implicações

Cada uma das células da Figura 2 é preenchida de acordo com a Figura 3.

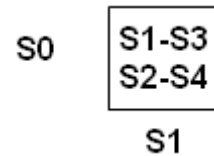


Figura 3 - Associação dos Estados S0 e S1

O Exemplo da Figura 3 ilustra o tipo de informação a registar, em cada uma das células, na CI.

S₁ e S₃ são os estados para os quais S₀ e S₁ transitam quando X=0, S₂ e S₄ os estados seguintes de S₀ e S₁ respectivamente, quando X=1. Será necessário preencher esta informação para todos os pares da CI, excepto aqueles que à partida não são simplificáveis (aqueles cujas saídas sejam diferentes), esses retiram-se de imediato do plano de simplificação colocando uma cruz, e representam a primeira acção sobre a carta.

Tratando-se de um processo iterativo é necessário voltar ao início e com base nas associações de estados não simplificáveis, perceber quais os pares que não conduzem a um único estado.

O fluxo do processo recomeça na célula de coordenadas (S₁, S₀), percorre na vertical até a célula (S₆, S₀), reposiciona-se na célula (S₂, S₁), voltando a percorrer todas as células até (S₆, S₁), e assim sucessivamente até à célula (S₆, S₅).

Ao longo deste trajecto é necessário ir eliminando associações não simplificáveis. De que forma?

Dentro de cada célula existem múltiplas associações, em número igual a 2^n , com n igual ao número de entradas. Se existir pelo menos uma delas, que na CI já tenha sido considerada não simplificável, então os estados envolvidos nesta associação, também se consideram não equivalentes.

O processo deve ser repetido até ao momento em que já não é possível tirar qualquer conclusão. Ou seja, que mais nenhum par de estados resulte num estado equivalente.

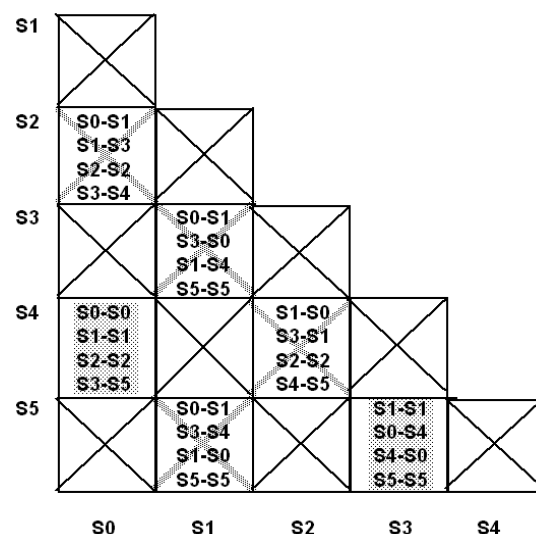


Figura 4 - Solução final da Carta de Implicações

Da Carta da Figura 4 apenas duas células não foram anuladas. Significa que os estados que formam as respectivas coordenadas das mesmas são equivalentes. Assim, S_4 e S_0 são equivalentes e S_5 e S_3 também, dando origem, cada um dos pares, a um único estado equivalente.

O resultado final pode ser consultado na tabela da Figura 5.

Present State	Next State				Output
	00	01	10	11	
S'_0	S'_0	S'_1	S'_2	S'_3	1
S'_1	S'_0	S'_3	S'_1	S'_3	0
S'_2	S'_1	S'_3	S'_2	S'_0	1
S'_3	S'_1	S'_0	S'_0	S'_3	0

Figura 5 - Tabela de estados simplificada

Se o resultado corresponde a uma tabela com um menor número de estados, então é possível implementar a mesma máquina de estados com menos recursos.

B. Usabilidade

Ser um processo iterativo de passos objectivos, e com um critério de paragem bem definido, foram factores que constituíram incentivo suficiente para avançar com a criação de um software que utiliza o Excel como interface e o VBA como linguagem de programação.

A mais valia desta ferramenta é que resolve uma tabela de transições de forma rápida, sem erros e com a vantagem de, graficamente, simular a resolução feita no papel. Este tipo de ferramenta constitui um excelente incentivo à aprendizagem.

O fluxo de informação resume-se essencialmente ao *input* de uma tabela de transições e ao *output* da mesma tabela, simplificada.

Na configuração da tabela de *input* são necessários os seguintes valores: o número de estados da máquina, o nº variáveis de entrada e o nº de variáveis de saída. Definidos estes parâmetros, é possível inserir a tabela de transições linha a linha e depois executar a ferramenta de simplificação.

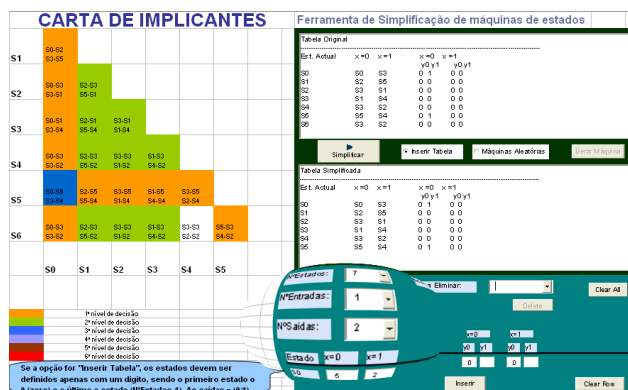


Figura 6 - Interface global da ferramenta

O resultado que se obtém na interface após inserir a tabela de transições e simplificar pode ser visto na Figura 7.

TABELA ORIGINAL												
Est. Actual	x=0	x=1	x=2	x=3	x=0	x=1	x=2	x=3	y0	y0	y0	y0
S0	S0	S1	S2	S3	1	1	1	1	0	0	0	0
S1	S0	S3	S1	S5	0	0	0	0	0	0	0	0
S2	S1	S3	S2	S4	1	1	1	1	0	0	0	0
S3	S1	S0	S4	S5	0	0	0	0	0	0	0	0
S4	S0	S1	S2	S5	1	1	1	1	0	0	0	0
S5	S1	S4	S0	S5	0	0	0	0	0	0	0	0

Tabela Simplificada												
Est. Actual	x=0	x=1	x=2	x=3	x=0	x=1	x=2	x=3	y0	y0	y0	y0
S0	S0	S1	S2	S3	1	1	1	1	0	0	0	0
S1	S0	S3	S1	S3	0	0	0	0	0	0	0	0
S2	S1	S3	S2	S0	1	1	1	1	0	0	0	0
S3	S1	S0	S0	S3	0	0	0	0	0	0	0	0

Figura 7 - Tabela de transições *input/output*

Do ponto de vista do objectivo final da ferramenta, podemos afirmar que é o que mais de importante ela produz: a máquina simplificada. Contudo a ferramenta tem um lado didáctico ao revelar de forma gráfica a execução da CI.

A ferramenta tem uma escala cromática de decisão. Ou seja, como já foi referido, o algoritmo assenta num processo iterativo e cada ciclo desse processo representa um plano de decisão.

O facto das células estarem coloridas, significa que os estados que lhe estão associados não são equivalentes. Um conjunto de células da mesma cor representa um grupo de conclusões tomadas no mesmo ciclo de decisão.

Os níveis considerados foram apenas 6, do laranja ao vermelho, sendo mais do que suficientes para os limites impostos à interface (nº de estados, nº de entradas e saídas)

	1º nível de decisão (laranja)
	2º nível de decisão (verde)
	3º nível de decisão (azul)
	4º nível de decisão (lilás)
	5º nível de decisão (bordeaux)
	6º nível de decisão (vermelho)

Figura 8 – Escala cromática de decisão

A resposta do algoritmo é feita de forma faseada, para que seja perceptível ao utilizador acompanhar a evolução da execução do algoritmo.

O resultado é aquele que se encontra na Figura 9 e pode ser comparado com o da Figura 4. Desde logo se vislumbra uma vantagem clara no mapa implementado em software: é fácil perceber os diferentes planos de decisão, neste caso o algoritmo atingiu o critério de paragem ao fim de 2 ciclos de iteração.

CARTA DE IMPLICANTES						
S1	S0-S0					
	S1-S3					
	S2-S1					
	S3-S5					
S2	S0-S1	S0-S1				
	S1-S3	S3-S3				
	S2-S2	S1-S2				
	S3-S4	S5-S4				
S3	S0-S1	S0-S1	S1-S1			
	S1-S0	S3-S0	S3-S0			
	S2-S4	S1-S4	S2-S4			
	S3-S5	S5-S5	S4-S5			
S4	S0-S0	S0-S0	S1-S0	S1-S0		
	S1-S1	S3-S1	S3-S1	S0-S1		
	S2-S2	S1-S2	S2-S2	S4-S2		
	S3-S5	S5-S5	S4-S5	S5-S5		
S5	S0-S1	S0-S1	S1-S1	S1-S1	S0-S1	
	S1-S4	S3-S4	S3-S4	S0-S4	S1-S4	
	S2-S0	S1-S0	S2-S0	S4-S0	S2-S0	
	S3-S5	S5-S5	S4-S5	S5-S5	S5-S5	
	S0	S1	S2	S3	S4	

Figura 9 – Carta de Implicações

As células a branco associam estados que são equivalentes. No exemplo dado S_4 é equivalente a S_0 e S_5 é equivalente a S_3 .

A ferramenta tem dois modos de funcionamento, um que permite simplificar uma máquina de estados a partir de uma tabela de transições e outro, num ambiente de aprendizagem, simula a construção aleatória de máquinas de estados finitos e resolve a CI. Este segundo modo não tem uma forma inteligente de construção de máquinas de estados, resultando em transições puramente aleatórias. Esta forma, simples, de apurar tabelas de transições, não permite criar com frequência exemplos com muitos estados simplificáveis. A pesar deste lado negativo, o segmento de código gerador da tabela de *input* é completamente independente do segmento de simplificação.

III. TECNOLOGIA UTILIZADA

Utiliza como linguagem de desenvolvimento, para processamento de informação, o **Visual Basic for Applications** (VBA) e o Excel como interface e suporte ao estudo.

Ao nível da arquitectura de software, foi desenvolvida segundo o paradigma procedimental simples e aquilo que é mais relevante apontar é a utilização de estruturas de dados para representar:

- Uma linha da tabela de transições de estados;
- Uma célula da CI;
- Uma linha da CI.

A partir de cada uma destas estruturas facilmente se obtém de um *array* de:

- a), a tabela de transições;
- b), uma linha da CI;
- c), a Carta propriamente dita.

IV. ALGORITMO

O algoritmo segue à risca o procedimento manual. O que pode variar numa máquina é o número de estados, o número de entradas e o número de saídas. O procedimento para a simplificar não se altera pelo facto dos parâmetros referidos se alterarem. Sendo assim, o tempo de processamento pode ser maior ou menor, mas a capacidade de resolução é sempre a mesma. Ou seja, o critério de paragem é sempre atingido e acontece no momento em que a máquina de estados atinge o estado mais simples.

A. Heurísticas do algoritmo

As heurísticas acabam por resumir o que já foi dito na secção anterior. Admitindo que os dados fornecidos correspondem à tabela de transições, devem ser dados os seguintes passos com o objectivo de a simplificar:

1. Desenhar a CI de acordo com os parâmetros da Máquina de Estados;
2. Colocar uma cruz nas células da carta cujos estados associados não tenham a mesma saída;
3. Preencher as restantes células com as transições de estados, de acordo com o que foi dito na secção II.
4. Percorrer a carta, de cima para baixo e da esquerda para a direita e verificar se existem associações de estados (S_x-S_y) que na carta já tenha sido descartada (estados envolvidos na associação considerados não equivalentes);
5. Depois de percorrer todas as células da CI voltar ao início e repetir o processo tendo em conta as conclusões tiradas até a última passagem;
6. Terminar na primeira iteração em que mais nenhuma conclusão se puder tirar.

B. Exemplo

Esta secção apresenta um pequeno exemplo resolvido pela ferramenta com o objectivo de consolidar o que foi dito.

1. Tabela de transições

A tabela da Figura 10 representa o comportamento da máquina de estados inicial. Esta informação será processada e constituirá o *input* da ferramenta.

Tabela Original				
Est. Actual	x=0	x=1	x=0 y0	x=1 y0
S0	S1	S3	0	0
S1	S3	S1	1	0
S2	S5	S4	1	0
S3	S2	S3	0	0
S4	S3	S4	0	0
S5	S2	S5	0	0

Figura 10 - Tabela de transições da máquina original

2. CI -1ª iteração

Na primeira iteração considera-se que os estados que tenham para as mesmas entradas, saídas diferentes, não possam ser considerados equivalentes. Isto resulta no descartar das primeiras, potenciais, relações de equivalência, assinaladas a laranja (sombreadas).

S1	S1-S3 S3-S1				
S2	S1-S5 S3-S4	S3-S5 S1-S4			
S3	S1-S2 S3-S3	S3-S2 S1-S3	S5-S2 S4-S3		
S4	S1-S3 S3-S4	S3-S3 S1-S4	S5-S3 S4-S4	S2-S3 S3-S4	
S5	S1-S2 S3-S5	S3-S2 S1-S5	S5-S2 S4-S5	S2-S2 S3-S5	S3-S2 S4-S5
	S0	S1	S2	S3	S4

Figura 11 - 1º passo da CI

3. CI -2ª iteração

Voltar ao início e repetir o processo, agora tendo em consideração que os estados que constituem as coordenadas das células a laranja (sombreadas), não são equivalentes.

O resultado desta segunda passagem está na Figura 12.

S1	S1-S3 S3-S1				
S2	S1-S5 S3-S4	S3-S5 S1-S4			
S3	S1-S2 S3-S3	S3-S2 S1-S3	S5-S2 S4-S3		
S4	S1-S3 S3-S4	S3-S3 S1-S4	S5-S3 S4-S4	S2-S3 S3-S4	
S5	S1-S2 S3-S5	S3-S2 S1-S5	S5-S2 S4-S5	S2-S2 S3-S5	S3-S2 S4-S5
	S0	S1	S2	S3	S4

Figura 12 - 2º passo da CI

4. CI -3ª iteração

Após a terceira passagem são descartadas mais duas células (sombreadas a escuro). A última a ser considerada fica em branco. Neste ponto verifica-se o critério de paragem e a CI diz-se resolvida – ver Figura 13.

S1	S1-S3 S3-S1				
S2	S1-S5 S3-S4	S3-S5 S1-S4			
S3	S1-S2 S3-S3	S3-S2 S1-S3	S5-S2 S4-S3		
S4	S1-S3 S3-S4	S3-S3 S1-S4	S5-S3 S4-S4	S2-S3 S3-S4	
S5	S1-S2 S3-S5	S3-S2 S1-S5	S5-S2 S4-S5	S2-S2 S3-S5	S3-S2 S4-S5
	S0	S1	S2	S3	S4

Figura 13 - 3º passo da CI

Da leitura final da CI pode-se concluir que o estado S₅ é equivalente a S₃.

O último passo de todo o processo é, utilizando a informação acima, representar a tabela de transições simplificada.

O resultado final está representado na Figura 14.

Tabela Simplificada				
Est. Actual	x=0	x=1	x=0 y0	x=1 y0
S0	S1	S3'	0	0
S1	S3'	S1	1	0
S2	S3'	S4	1	0
S3'	S2	S3'	0	0
S4	S3'	S4	0	0

Figura 14 - Tabela de transições simplificada

IV. CONCLUSÕES

O trabalho levado a cabo para desenvolver esta ferramenta não tem nada de extraordinário. Não revela avanços tecnológicos nem processos complexos de processamento de dados. Escusa-se de apontar novas metodologias para simplificar Máquinas de Estados Finitos. Pelo contrário, apresenta um algoritmo de papel e lápis bem conhecido, codificado numa linguagem de programação comum.

Em suma, deste ponto de vista não traz grande novidade, contudo tem o objectivo de revelar uma ferramenta que pode ser útil em ambientes de aprendizagem. Mais, pretende deixar uma chamada de atenção para o facto de esta resultar da combinação de outras ferramentas que exigem uma curva de aprendizagem pequena, mas com um resultado final interessante e útil.

O facto de utilizar o Excel como interface potencia a possibilidade do aluno copiar e colar numa outra folha de cálculo os exemplos mais relevantes. Permite utilizar a estrutura matricial da folha e tirar proveito do relevo cromático para destacar os diferentes momentos de decisão do algoritmo. Aglutina numa só ferramenta as vertentes de instrumento resolvente do problema e de demonstração de resolução.

De referir ainda a dupla funcionalidade da ferramenta. Permite definir a tabela de transições que vai ser simplificada ou, por outro lado, gerar aleatoriamente as máquinas e estudar as resoluções, opção que privilegia a aprendizagem.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Randy H. Katz , *Contemporary Logic Design* , Prentice-Hall, 1993