


ALGUMAS PROPRIEDADES DAS MÁQUINAS

SEM PERDA DE INFORMAÇÃO

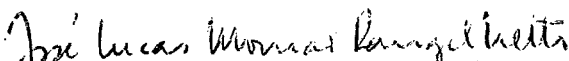
MOISÉS RENNÓ VILELA

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

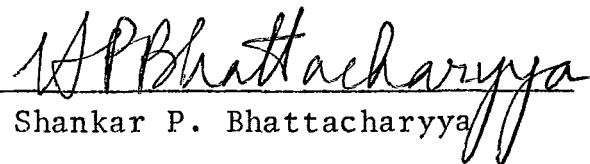
Aprovada por:



Paulo Augusto Silva Veloso
(Presidente)



José Lucas M. Rangel Neto



Shankar P. Bhattacharyya

RIO DE JANEIRO, RJ - BRASIL

MAIO DE 1977

VILELA, MOISÉS RENNÓ

Algumas Propriedades das Máquinas sem Perda de Informação.
Rio de Janeiro 1977.

VI, 96 p. 29,7 cm (COPPE - UFRJ, M.Sc.,
Engenharia de Sistemas, 1977.

Tese - Univ. Fed. Rio de Janeiro. Fac. de Engenharia.

1. Máquinas Sequenciais. I. COPPE/UFRJ II. Título (Série).

Este trabalho é dedicado a
Meire, Marytta, Myrella e
aos meus pais.

AGRADECIMENTOS

Ao Professor Paulo Augusto Silva Veloso pela indicação do assunto e orientação deste trabalho; ao Professor Nelson Maculan Filho tanto pelo apoio como pelo incentivo constante; aos Professores José Lucas M. Rangel Neto, Shankar P. Bhattacharyya; aos demais professores e colegas da COPPE que de alguma forma colaboraram com o nosso trabalho.

SINOPSE

Este trabalho trata de algumas propriedades das máquinas sem perda de informação, tais como, interconexões série, paralela, cascata e realimentação. Estuda também o comportamento destas máquinas sob homomorfismos, e faz o seu relacionamento com as máquinas de memória finita. Encontra-se também neste trabalho a síntese da máquina inversa de uma máquina sem perda de informação que é ao mesmo tempo máquina de memória finita.

A importância desta classe de máquinas é devido a suas aplicações em problemas de codificação de informação.

ABSTRACT

This work deals with some properties of information-lossless machines, such as series, parallel and cascade interconnections as well as those where feedback is present. It also studies the behavior of information lossless machines under homomorphisms and establishes their relationship with finite memory machines. The work also presents the synthesis of the inverse machine (that is a machine which is the inverse of an information-lossless machine acting at the same time as a finite-memory machine).

The importance of these classes of machine lies in their application to problems involving information coding.

S U M Á R I O

	Pág.
I. Introdução	1
II. Revisão da Literatura	5
III. Máquinas Sem Perda de Informação	7
IV. Classificação das Máquinas Sem Perda de Informa <u>ç</u> ção	19
V. Máquina Sem Perda de Informação Reduzida	24
VI. Interconexões de Máquinas Sequenciais Sem Perda de Informação	27
VII. Relação entre Máquinas Sem Perda de Informação e Máquinas de Memória Finita	48
VIII. Máquinas Definidas	63
IX. Conclusões	68
X. Referências Bibliográficas	71

I. INTRODUÇÃO

Um dos problemas principais na codificação e transmissão de informações é a determinação das condições sob as quais é possível reconstruir a sequência de entrada de uma máquina a partir de sua correspondente sequência de saída. Existe uma importante classe de máquinas que, quando excitadas por uma sequência de entrada, produzem uma sequência de saída de tal modo que, depois de um experimento de tamanho finito sobre as máquinas, sua sequência de entrada pode ser determinada a partir do conhecimento da correspondente sequência de saída, do estado inicial, do estado final e das especificações pelas quais as máquinas transformam sequência de entrada em sequência de saída. Estas máquinas são chamadas de máquinas sem perda de informação. Ainda dentro desta classe de máquinas encontramos as máquinas de ordem finita que são aquelas em que para o conhecimento de um símbolo da sequência de entrada basta conhecermos o estado inicial e uma sequência de saída de comprimento finito. Esta propriedade das máquinas sem perda de informação pode ser usada na solução de inúmeros problemas. Como um exemplo, consideremos o problema de verificar a precisão de operação do circuito mostrado na figura 1. Um dos métodos para resolver este problema consiste na redução da verificação da precisão de operação do circuito para sucessivas verificações da precisão de operação de cada máquina. Para aplicação deste método temos que levar em conta que somente temos acesso à saída da máquina M_7 . As saídas das máquinas M_5 e M_6 são entradas da máquina M_7 ; então, para verificarmos a precisão de operação de cada uma de-

las, é necessário conhecermos a entrada da máquina M_7 a partir da resposta do circuito. Esta situação é análoga para o caso das demais máquinas. Então, uma das condições necessárias simplificadoras para solução do problema de verificar a precisão de operação de circuitos como este é que todas as máquinas sejam sem perda de informação.

Aqui neste trabalho tratamos estas máquinas estritamente sob o ponto de vista teórico considerando-as sob o seu aspecto de transformação entrada/saída.

Analisando uma grande parte dos trabalhos existentes sobre máquinas sem perda de informação, sentimos a necessidade de se fazer um estudo de mais algumas de suas propriedades, tais como seu comportamento diante do homomorfismo e interconexões, seu relacionamento com classe de máquinas de memória finita e definidas.

No capítulo 3 introduzimos algumas terminologias teóricas sobre máquinas, definimos e identificamos máquinas sem perda de informação. Em KOHAVI¹ encontramos que se uma máquina é de ordem finita, é evidente que $k \leq n(n-1)/2+1$, onde k é o comprimento da sequência de saída que precisamos conhecer para determinarmos o primeiro símbolo da sequência de entrada e n é o número de estados da máquina. Neste capítulo nós mostramos esta evidência.

No capítulo 4 fazemos uma classificação de máquinas de ordem finita. Damos um algoritmo para construção de máquinas de ordem finita k que não são de ordem finita $(k-1)$.

No capítulo 5 estudamos o comportamento das máquinas sem perda de informação mediante o homomorfismo. Mostramos que quando uma máquina de ordem finita que possui estados equivalentes é submetida aos processos de redução sua ordem não altera.

No capítulo 6 interconectamos as máquinas sem perda de informação e analisamos quais seriam as características da máquina resultante.

Nos capítulos 7 e 8 fazemos o relacionamento das máquinas sem perda de informação com as máquinas de memória finita e definidas, verificando qual a vantagem de se ter uma máquina de memória finita que seja sem perda de informação. Damos um algoritmo para construção de tais máquinas.

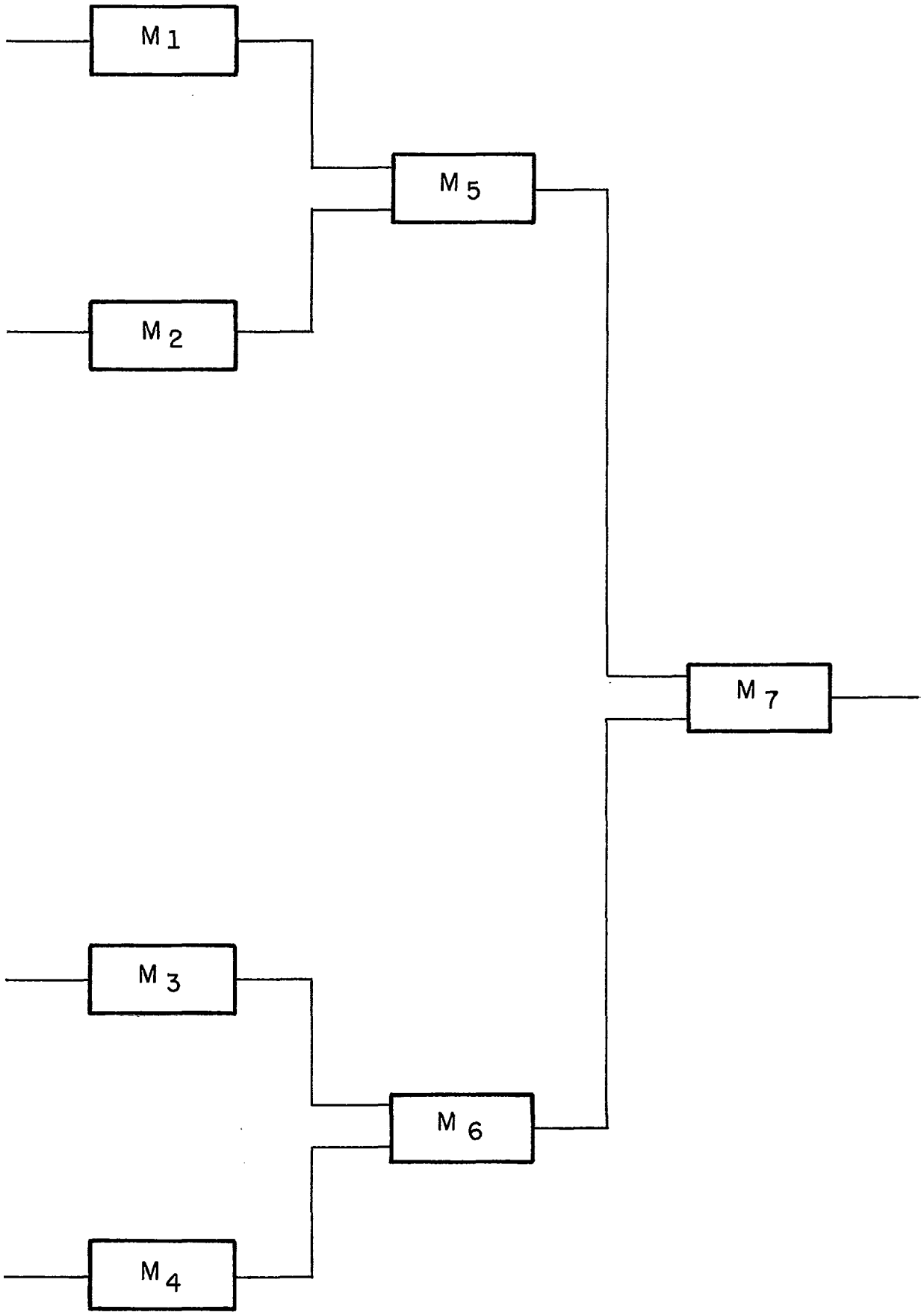


Figura 1

II. REVISÃO DA LITERATURA

Um dos primeiros pesquisadores a investigar máquinas sem perda de informação foi HUFFMAN³. Ele, em seu trabalho, definiu a classe de máquinas ditas sem perda de informação. Mostrou uma forma canônica de circuitos em que estas máquinas podem ser sintetizadas e investigou a existência dos inversos destas máquinas. Definiu também as máquinas de ordem finita k em que, para determinarmos o primeiro símbolo da sequência de entrada, basta conhecermos o estado inicial e uma sequência de saída de comprimento k . Apresentou um teste para verificar quando é que uma máquina é de ordem finita k . Analisou a construção destas máquinas diretamente em termos de Hardware. Um dos propósitos principais de seu trabalho foi desenvolver um diagrama que representasse uma máquina bem geral que fosse sem perda de informação.

EVEN⁴ descreve um teste mais eficiente que o teste desenvolvido por HUFFMAN³. Seu teste é baseado em termos dos estados da máquina. É mais facilmente analisado e pode ser minimizado antes de sua realização. Ele discorre sobre o que seria um grafo sem perda de informação e logo em seguida faz aplicação para máquinas sequenciais. Também aborda os inversos destas máquinas sem perda de informação. Ele, em seu trabalho, nos dá algoritmos simples e eficientes para detecção de máquinas sem perda de informação.

NEUMANN⁹, em seu trabalho, investiga os efeitos de erros temporários, tais como erros de entrada (transfor

mação de símbolos, inserção), erros de transição de estados, erros de saída, etc, em sistemas de comunicação que usam máquinas sem perda de informação. Ele mostra que existem codificadoras para as quais as decodificadoras trabalham incorretamente somente temporariamente quando qualquer um destes erros ocorrem no sistema de comunicação. Cada uma destas máquinas codificadoras sem perda de informação consideradas por NEUMANN⁹, num sistema de comunicação têm uma sequência de entrada que as resincroniza quando da ocorrência de um erro, bem como uma sequência de saída para resincronizar a decodificadora. Ele parte do ponto de que uma codificadora e uma decodificadora combinacional, podem se resincronizar para códigos infinitamente grande, quando da ocorrência de um erro, e faz uma generalização para o caso de máquinas sem perda de informação.

LAVALEE⁸ desenvolve um método para se obter a partir de uma máquina arbitrária M , uma máquina sem perda de informação que contenha M . Seu método é baseado em modificações da máquina original com adição de uma saída lógica. Para uma máquina com 2^k estados, o número de saídas lógicas adicionais necessário para se obter uma máquina M_1 que contenha M e seja sem perda de informação é igual a k .

SERANSKII¹⁰ desenvolve um método para se construir uma máquina que seja sem perda de informação sobre um conjunto regular de palavras não vazio especificado. Ele parte de uma máquina arbitrária M e vai introduzindo saídas adicionais até se obter uma nova máquina M_1 que tenha a propriedade mencionada.

III. MÁQUINAS SEM PERDA DE INFORMAÇÃO

III.1. Máquinas

Uma máquina sequencial (de Mealy) pode ser representada pela quintupla $M = \langle \Sigma, S, \Delta, f, g \rangle$, onde:

- a) Σ é um conjunto finito não vazio que representa o alfabeto de entrada.
- b) S é um conjunto finito não vazio que representa os estados da máquina.
- c) Δ é um conjunto finito não vazio que representa o alfabeto de saída.
- d) $f : S \times \Sigma \rightarrow S$ representa a função transição de estados.
- e) $g : S \times \Sigma \rightarrow \Delta$ representa a função saída. (Ver BOOTH²).

Em M o próximo estado $S(t+1)$ é determinado unicamente em função do estado atual $S(t)$ e da entrada presente $x(t)$, ou seja : para x pertencendo a Σ ($x \in \Sigma$), $z(t) \in \Delta, s(t), s(t+1) \in S$,

$$f(s(t), x(t)) = s(t+1)$$

$$g(s(t), x(t)) = z(t)$$

Podemos também usar uma seta (\rightarrow) para escrevermos as duas equações acima, isto é;

$$s(t), x(t) \xrightarrow{f} s(t+1)$$

que interpretamos como : $s(t), x(t)$ através da função f implica

em obtermos $S(t+1)$. Do mesmo modo $s(t), x(t) \xrightarrow{g} z(t)$ interpretamos como : $s(t), x(t)$ através da função g implica em obtermos $z(t)$. Sempre com esta notação queremos dizer que com a informação que temos à esquerda da seta determinamos a informação que está à direita da seta.

Usualmente nós estendemos as funções de transição $f : S \times \Sigma \rightarrow S$ e de saída $g : S \times \Sigma \rightarrow \Delta$ para $f : S \times \Sigma^* \rightarrow S$ e $g : S \times \Sigma^* \rightarrow \Delta^*$, tal que se $u, v \in \Sigma^*$ e $s(t) \in S$, $f(s(t), uv)$ é igual a $f(f(s(t), u), v)$ e $g(s(t), uv)$ é igual a $g(f(s(t), u), v)$ e $g(s(t), u)$. Onde Σ^* e Δ^* representam o conjunto de sequências de comprimento finito, positivo ou nulo sobre, respectivamente, Σ e Δ .

Dizemos que M é uma máquina de Moore se e somente se todos seus estados são estados de Moore, isto é, se e somente se para todo $\sigma, \gamma \in \Sigma$ e $s(t) \in S$, $g(s(t), \sigma) = g(s(t), \gamma)$. Neste caso podemos definir a função de Moore $h : S \rightarrow \Delta$, tal que $h(s)$ seja igual a $g(s, \sigma)$.

III.2. Definição (Ver KOHAVI¹)

Uma máquina $M = \langle \Sigma, S, \Delta, f, g \rangle$ é sem perda de informação se o conhecimento do estado inicial, da sequência de saída e do estado final é suficiente para determinar a sequência de entrada, isto é, se $s_i, s_f \in S$, $y \in \Delta^*$ e $x \in \Sigma^*$ então $s_i, s_f, y \rightarrow x$.

Uma máquina M será sem perda de informação quando nela não ocorrer, para algum estado, dois ou mais suces

sores com suas respectivas saídas idênticas, ou duas sequências de entrada distintas dando uma mesma sequência de saída para um mesmo estado inicial e um mesmo estado final.

III.3. Identificação de uma máquina sem perda de informação

Mostramos aqui neste ítem, com auxílio de dois exemplos, a definição, a identificação de uma máquina sem perda de informação e a condição para que ela seja sem perda de informação.

A máquina M da figura 2 é sem perda de informação pois:

1. nenhuma das linhas da tabela de transição da figura 2 possui dois próximos estados idênticos associados com símbolos de saída iguais.
2. A tabela de teste da figura 3 não possui nenhum par de estados $s_i = s_j$ que são (sob a saída) y_k sucessores de algum estado s_p , ou seja, não possui nenhum par compatível consistente de estados iguais. Na figura 3 o par não ordenado formado pelos estados A e B é o par compatível AB, pois os dois estados A e B são $y_k = 00$, sucessores de A. O par compatível BC é um par implicado pelo par compatível AB.

A máquina M da figura 4 é com perda de informação pois sua tabela de teste representada na figura 4 possui um par compatível (por exemplo BB) consistindo de dois estados idênticos.

	0	1
A	A/1	C/1
B	E/0	B/1
C	D/0	A/0
D	C/0	B/0
E	B/1	A/0

Figura 2

	0	1
A	-	AC
B	E	B
C	AD	-
D	BC	-
E	A	B
AC	-	-
AD	-	-
BC	AE, DE	-
AE	-	AB, BC
DE	AB, BC	-
AB	-	AB, BC

Figura 3

	0	1
A	A/1	B/1
B	A/0	C/1
C	B/1	A/0

	0	1
A	-	AB
B	A	C
C	A	B
AB	-	AC, BC
AC	-	AB, BB
BC	AA	BC

Figura 4

Na máquina da figura 4 temos $A, B, 111 \rightarrow 001$ e $A, B, 111 \rightarrow 110$, ou seja, um estado inicial A , um estado final B e uma única sequência de saída associada com duas sequências de entrada diferentes, contrariando a definição de máquinas sem perda de informação.

III.4. Máquina Inversa

Em KOHAVI¹ encontramos que uma máquina M_i é inversa de M , se quando M_i for excitada pela sequência de saída de M produzir a sequência de entrada de M com um atraso finito.

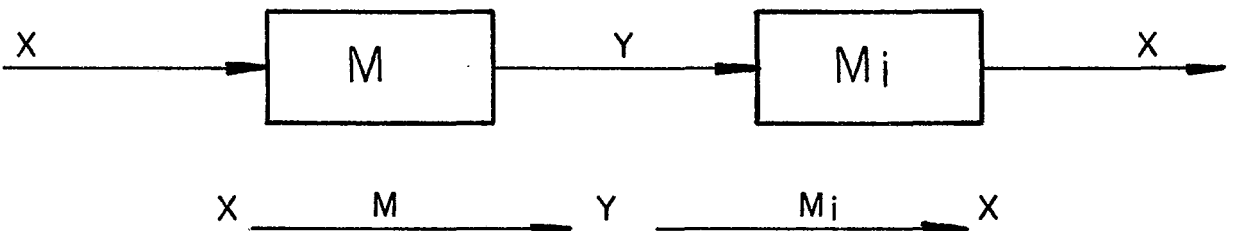


Figura 5

É evidente que a máquina M_i só pode ser construída se M for sem perda de informação, e ela poderá ser construída para produzir a sequência de entrada de M depois de um atraso finito se e somente se M for uma máquina sem perda de informação de ordem finita, isto é, se M for tal que o conhecimento do estado inicial e uma sequência de saída de comprimento finito forem suficientes para determinar um símbolo da sequência de entrada.

O sistema da figura 5 é usado para codificação e transmissão de informações. M codifica a informação e transmite para M_i que decodifica, reconstituindo assim a informação recebida por M .

Este sistema de codificação, transmissão e decodificação é usado na transmissão de informações através de canais em que o sigilo quanto às mesmas é muito importante, ou que os sinais estão sujeitos à intervenção humana ou a ruídos.

III.5. Máquinas sem perda de informação de ordem finita

Quando tivermos um sistema de máquinas sem perda de informação conforme a figura 5, sendo usado como um dispositivo de codificação e decodificação de informações, a reconstrução da sequência de entrada é feita levando-se em conta o estado inicial e o estado final. O maior problema neste dispositivo de decodificação é que a informação quanto ao estado final só é transmitida depois que toda sequência de saída for transmitida. Consequentemente toda informação deverá ser armazenada para depois iniciar a decodificação. Então, para o caso de

uma sequência de saída muito longa, este processo de codificação, transmissão e decodificação deixa de ser viável quanto ao seu uso. Tendo em vista esta limitação, desenvolveu-se um estudo para construção de uma máquina em que não fosse necessário o armazenamento de toda informação transmitida; uma máquina em que o processo de decodificação pudesse ser iniciado somente quando se conhecesse o estado inicial e uma sequência de saída de comprimento finito.

HUFFMAN³ definiu uma máquina que com o conhecimento do estado inicial e de uma sequência de saída de comprimento finito determinamos o primeiro símbolo da sequência de entrada. A esta máquina ele deu o nome de máquina de ordem finita.

Então se em $M = \langle \Sigma, S, \Delta, f, g \rangle$ para $s_1 \in \Delta$, $y \in \Delta^k$ (onde Δ^k representa o conjunto de todas as sequências de comprimento k de Δ), $x_1 \in \Sigma$, $s_1, y \rightarrow x_1$ dizemos que M é de ordem finita igual a k , se k for o mínimo com esta propriedade.

No sistema da figura 5 para uma máquina M de ordem finita k , M_i inicia a decodificação depois que receber k símbolos de saída transmitidos pela máquina M . Neste caso temos que: para $y_i \in \Delta$, $x_i \in \Sigma$

$$s_1, y_1 y_2 y_3 \cdots y_k \rightarrow x_1$$

$$s_1, x_1 \rightarrow s_2$$

$$s_2, y_2 y_3 \cdots y_{k+1} \rightarrow x_2$$

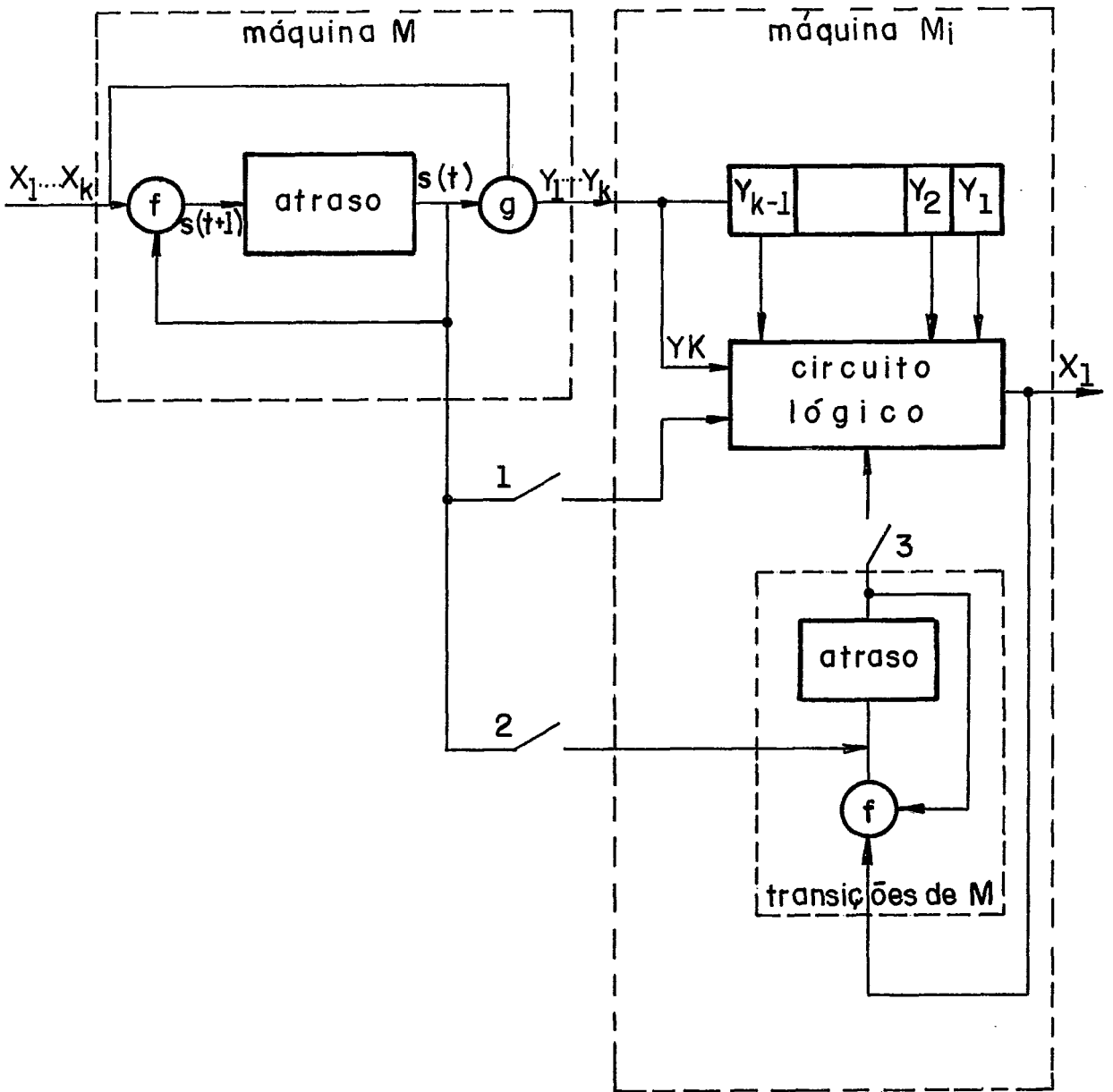


Figura 6

e assim por diante até que toda sequência de saída seja decodificada.

A figura 6 mostra um esquema de um sistema idêntico ao da figura 5, só que nesta temos uma máquina de ordem finita k . Na figura 6 representamos M conforme sua interpretação, isto é, estando no estado $s(t) \in S$, no instante t , M dá saída $y(t)$, tal que $s(t), u(t) \rightarrow y(t)$, e no próximo instante $(t+1)$ estará no estado $s(t+1)$, de modo que $s(t), u(t) \rightarrow s(t+1)$ é armazenado no atraso.

Transmitimos s_i com chaves 1 e 2 ligadas e vamos armazenando as saídas. Assim que acabar o armazenamento das saídas desligamos as chaves 1 e 2 e ligamos a chave 3. Com s_i e a saída o circuito lógico dá o primeiro símbolo da entrada que é introduzido numa seção onde temos a cópia das transições de M produzindo assim s_{i+1} , que é transmitido ao circuito lógico. Com s_{i+1} e os símbolos $y_{i+1}, y_{i+2} \dots y_{k+2}$ temos x_2 e assim por diante.

EVEN⁴ diz que uma máquina M é sem perda de informação de ordem finita $k = \ell + 2$ se e somente se seu grafo de teste for livre de ciclos, e se o comprimento do maior caminho do grafo é ℓ . Então consideremos por exemplo a máquina M sem perda de informação representada na figura 7. Sua tabela de teste está representada na figura 8. Desta tabela construímos seu grafo de teste que está representado na figura 8, e verificamos que o mesmo não tem ciclos e o seu maior caminho é igual a 1. Logo, a máquina da figura 7 é de ordem finita igual a 3, isto é, sua inversa tem que esperar receber 3 símbolos da sequên-

	0	1
1	2/0	1/0
2	4/0	3/0
3	2/1	1/1
4	4/1	3/1

Figura 7

	0	1
1	12	-
2	34	-
3	-	12
4	-	34
12	13,14 23,24	-
34	-	13,14 23,24

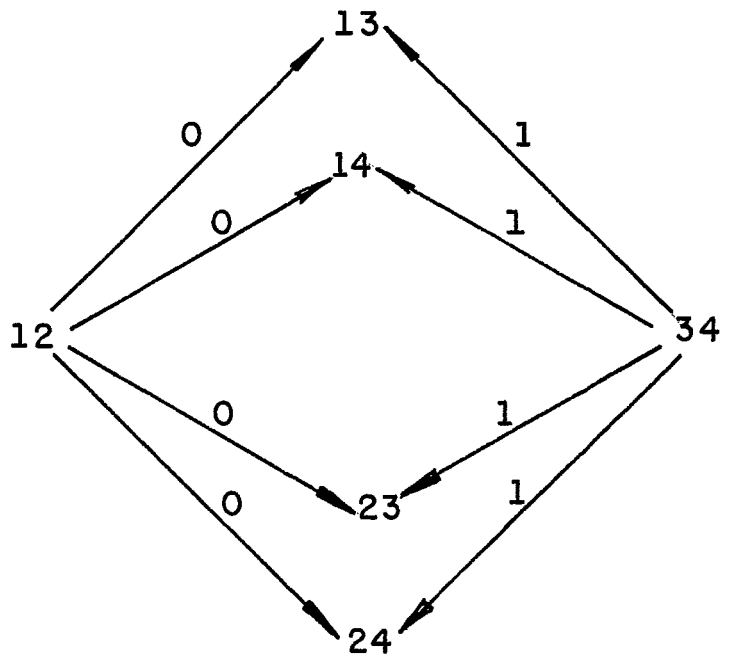


Figura 8

cia de saída para iniciar a decodificação.

Em KOHAVI¹ temos que se M é sem perda de informação de ordem finita k com n estados $k \leq n(n-1)/2+1$. Isto é evidente pois, se M tem ordem finita k podemos dizer que existe um $r \in S$ tal que $|a/z| = |b/z| = k$, onde $a_1 \neq b_1$, conforme figura 9 para $p_i, q_i \in S, a_i, b_i \in \Sigma, z_i \in \Delta, i = 1, \dots, k$. De acordo com a definição de máquinas sem perda de informação de ordem finita $p_i \neq q_i$ e $p_i q_i \neq p_j q_j$ ($i, j = 1, 2, \dots, (k-1)$). Se $p_i = q_i$ da figura 9, vemos que M será com perda de informação, pois teremos

$$r, p_i, z_1, z_2 \dots z_j \rightarrow a_1 a_2 \dots a_j \quad e$$

$$r, p_i, z_1, z_2 \dots z_j \rightarrow b_1 b_2 \dots b_j, \text{ isto é, aparecerá um par compatível consistindo de estados idênticos. Se } p_i = p_j \text{ e } q_i = q_j,$$

temos $p_i q_i = p_j q_j$, acontecendo o exposto na figura 10. Se $p_i = q_j$ e $q_i = p_j$, também temos $p_i q_i = p_j q_j$, acontecendo o exposto na figura 11. Destas duas figuras vemos que podemos construir duas sequências de entrada bastante longas que diferem no primeiro símbolo e que produzem a mesma sequência de saída. Para estes casos teremos no grafo de teste o par compatível $p_i q_i$ ligado a ele mesmo pela sequência de saída $z_{i+1} z_{i+2} \dots z_j$, isto é, vemos claramente a existência de um ciclo no grafo de teste, o que faz com que M deixe de ser sem perda de informação de ordem finita. Diante disto, vemos que todos os pares não ordenados são distintos. Com n estados podemos obter $\binom{n}{2} = n(n-1)/2$ pares de estados não ordenados e distintos. Da figura 9 vemos que temos $p_1 q_1, p_2 q_2, \dots, p_{k-1} q_{k-1}$, que são $(k-1)$ pares não ordenados e distintos. Então $k-1 \leq n(n-1)/2$.

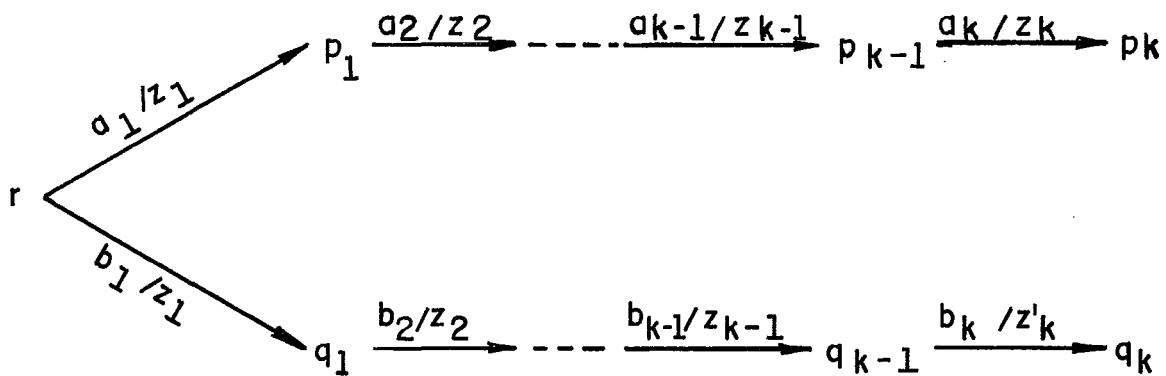


Figura 9

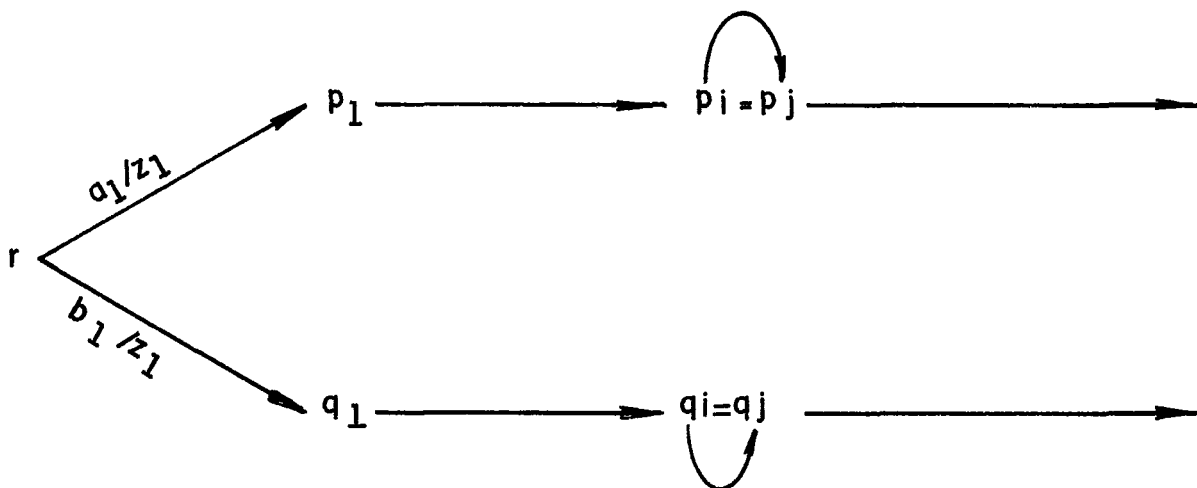


Figura 10

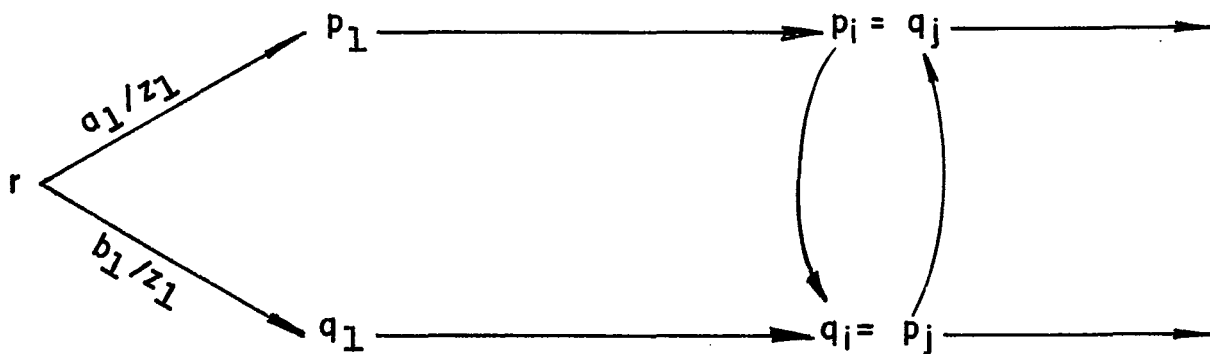


Figura 11

IV. CLASSIFICAÇÃO DAS MÁQUINAS SEM PERDA DE INFORMAÇÃO

- Proposição :

- a) Existem máquinas que são de ordem k mas que não são de ordem $k-1$.

- Demonstração :

Se seguirmos os passos do algoritmo abaixo, obtemos a máquina sem perda de informação de ordem finita K com 2^{k-1} estados, representada por sua tabela de transição na figura 13.

- Algoritmo :

1. Começamos a construir a árvore cuja forma mostramos também na figura 13, pela raiz que será o estado 1.
2. Da raiz passamos para o nível 1, que terá os estados 1 e 2. Do nível 1 passamos para o nível 2, que terá os estados 1, 2, 3 e 4, e assim por diante até o nível $k-1$, onde teremos os estados 1, 2, 3, 4 ... 2^{k-1} .
3. Colocamos as transições 0/0, 1/0 para os estados de 1 até 2^{k-2} .
4. Colocamos as transições 0/1, 1/1 para os estados de $(2^{k-2}+1)$

até (2^{k-1}) . Os estados 2^{k-1} , $2^{k-1}-1$, $2^{k-1}-2$, ... $2^{k-2} + 1$ vão respectivamente para os estados (1 e 2), (3 e 4), (5 e 6), $(2^{k-1}-1$ e $2^{k-1})$.

Como um exemplo de aplicação do algoritmo acima, vamos construir uma máquina que seja de ordem finita igual a 3 e que não seja de ordem finita 2. Com os passos 1 e 2 obtemos a figura 12.a. No nível $(k-1 = 2)$ temos os estados 1, 2, 3, 4. Com os passos 3 e 4 obtemos a figura 12.b.

Analisando a árvore representada na figura 12.b vemos que o passo 3 do algoritmo garante que para uma sequência de comprimento $(k-1)$ o primeiro símbolo da sequência de entrada não pode ser determinado. Com o passo 4 garantimos que com o conhecimento de qualquer sequência de saída de comprimento igual a 3 conseguimos determinar o primeiro símbolo da sequência de entrada.

O algoritmo acima constrói as máquinas para $k > 2$. Para $k \leq 2$ temos as máquinas representadas na figura 14.

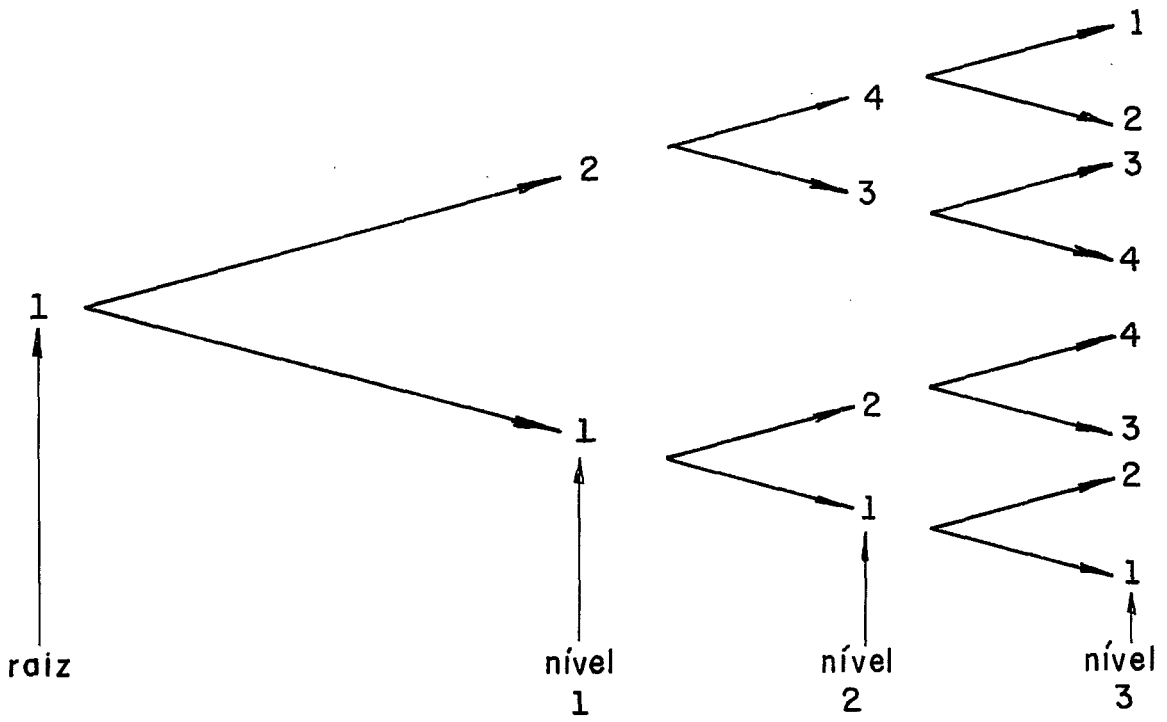


Figura 12a

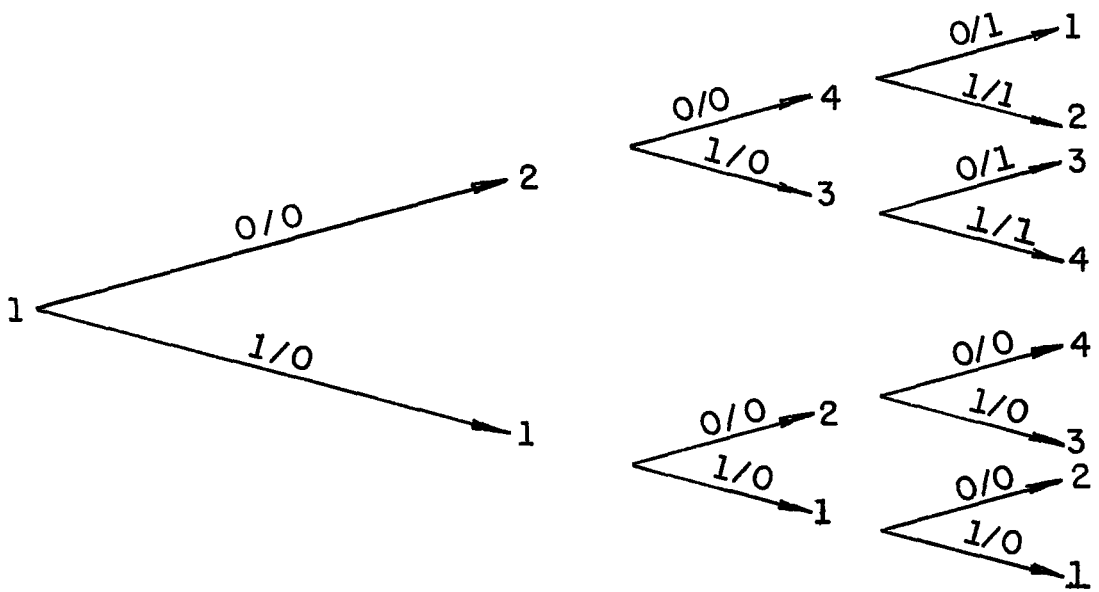
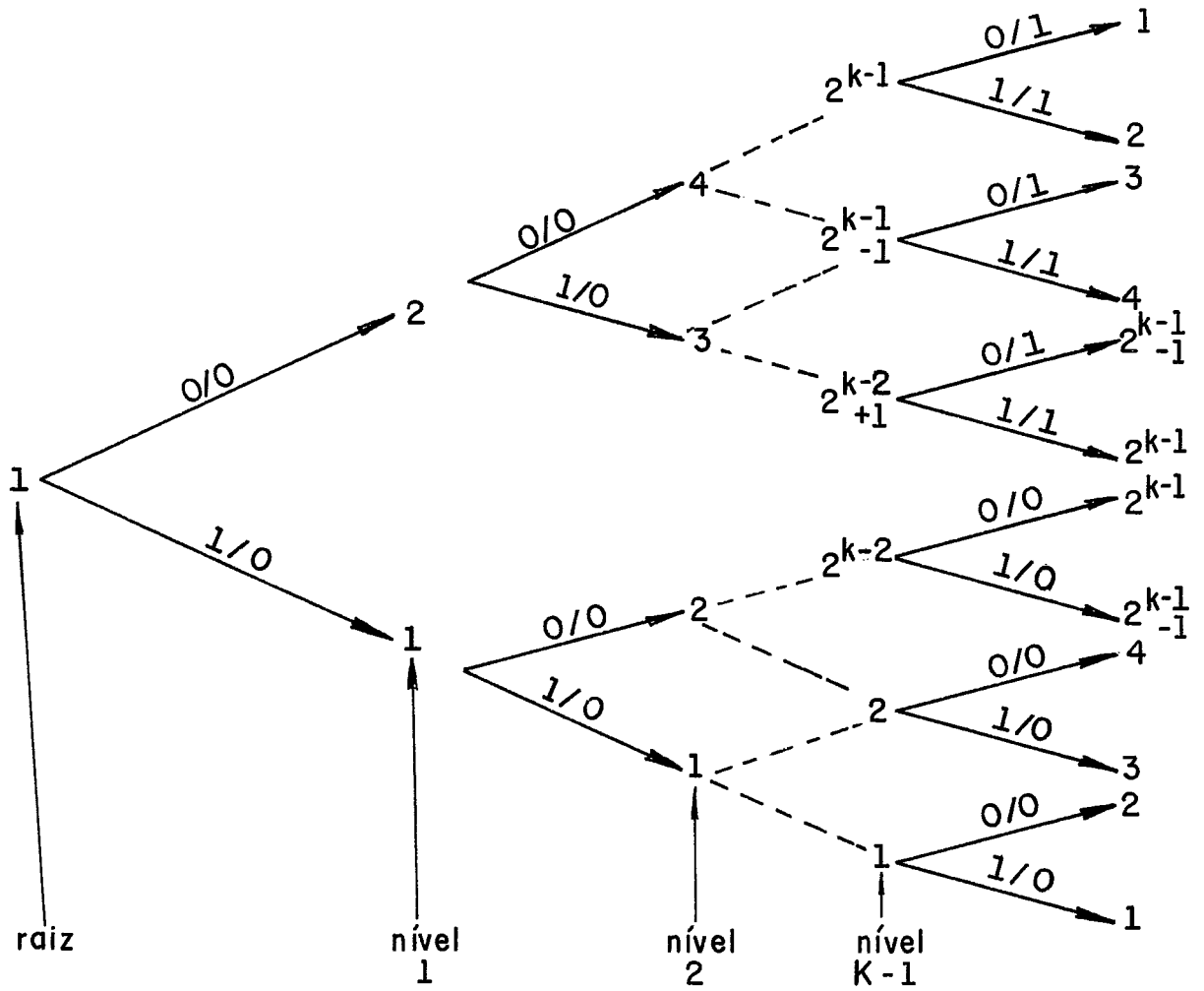
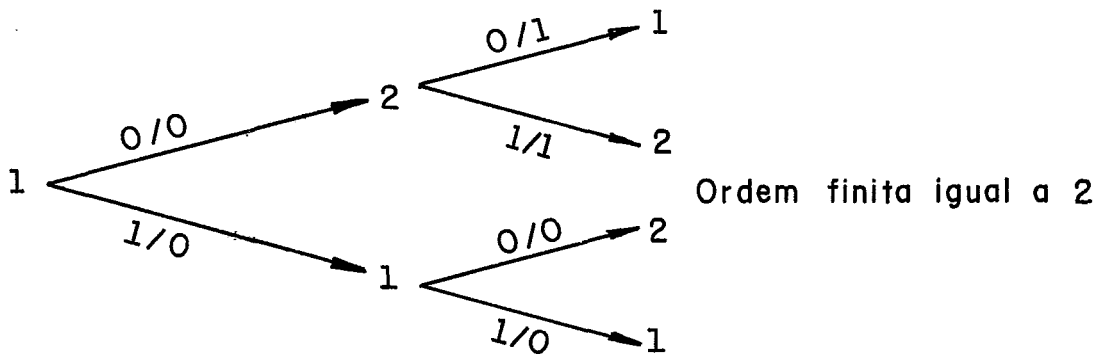


Figura 12b

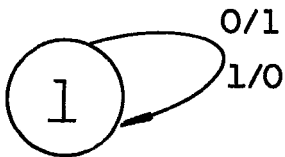


	0	1
1	2/0	1/0
2	4/0	3/0
3	6/0	5/0
2^{k-2}	$2^{k-1}/0$	$2^{k-1}/0$
2^{k-2}_{+1}	$2^{k-1}/1$ -1	$2^{k-1}/1$
2^{k-1}_{-1}	3/1	4/1
2^{k-1}	1/1	2/1

Figura 13



Ordem finita igual a 2



Ordem finita igual a 1

Figura 14

	0	1
s	p/o	q/o
p	p/o	p/l
q	q/o	q/l

máquina M_1

	0	1
s	pq	—
p	p	p
q	q	q
pq	pq	pq

tabela de teste da máquina M_1

Figura 15

	0	1
s	p/o	p/o
p	p o	p l

máquina M_2

	0	1
s	pp	—
p	p	p

tabela de teste da máquina M_2

Figura 16

V. MÁQUINA SEM PERDA DE INFORMAÇÃO REDUZIDA

Neste item mostramos o comportamento das máquinas sem perda de informação mediante a redução.

Dizemos que uma máquina $M = \langle \Sigma, S, \Delta, f, g \rangle$ é reduzida se e somente se não tem estados distintos que produzem uma mesma sequência de saída para uma mesma sequência de entrada, isto é, se e somente se não possuir estados distintos equivalentes. (Ver KOHAVI¹, página 285). Sabemos também que se $M_1 = \langle \Sigma, S_1, \Delta, f_1, g_1 \rangle$ possuir estados equivalentes, podemos, mediante processos de redução (ver KOHAVI¹, página 286), obter uma máquina reduzida $M_2 = \langle \Sigma, S_2, \Delta, f_2, g_2 \rangle$ que seja equivalente a M_1 . Uma das propriedades desta máquina reduzida é que ela é imagem homomorfa de M_1 , ou seja, existe uma função $\phi : S_1 \rightarrow S_2$ chamada homomorfismo, tal que para todo $s \in S_1, \sigma \in \Sigma$, $t_2(\phi(s), \sigma) = \phi(f_1(s, \sigma))$ e $g_2(\phi(s), \sigma) = g_1(s, \sigma)$.

Então sejam $M_i = \langle \Sigma, S_i, \Delta, f_i, g_i \rangle$ duas máquinas sequenciais com os mesmos alfabetos de entrada e saída, e seja $\phi : S_1 \rightarrow S_2$ um homomorfismo de M_1 sobre M_2 .

V.1. Proposição

- a) Se M_2 é sem perda de informação, então M_1 também é.
- b) Se uma das máquinas for de ordem finita, a outra também é, possuindo a mesma ordem.

- Demonstrações :

- a) A condição necessária para que M_1 seja sem perda de informação é que para todo $s_1 \in S_1$, $u, v \in \Sigma^*$ se $f_1(s_1, u)$ for igual a $f_1(s_1, v)$ e $g_1(s_1, u) = g_1(s_1, v)$, então u tem que ser igual a v .

Sabendo que ϕ é um homomorfismo de M_1 sobre M_2 , a condição acima fica

$$\phi(f_1(s_1, u)) = \phi(f_1(s_1, v))$$

ou seja,

$$f_2(\phi(s_1), u) = f_2(\phi(s_1), v)$$

$$g_2(\phi(s_1), u) = g_2(\phi(s_1), v)$$

Como M_2 é sem perda de informação $u = v$; logo M_1 é sem perda de informação.

- b) 1. Se M_1 é de ordem finita k_1 então M_2 é de ordem finita $k_2 \leq k_1$.

Dados $s_2 \in S_2$, $u, v \in \Sigma^{k_1}$, suponhamos que $g_2(s_2, u) = g_2(s_2, v)$.

- Como a função ϕ (do homomorfismo) é sobre, para algum $s_1 \in S_1$, $s_2 = \phi(s_1)$. Então $g_1(s_1, u) = g_1(s_1, v)$, e, como M_1 tem ordem k_1 , $u = v$.

2. Se M_2 é de ordem finita k_2 , então M_1 é de ordem finita $k_1 \leq k_2$.

Dados $s_1 \in S_1$, $u, v \in \Sigma^{k_2}$, suponhamos que $g_1(s_1, u) = g_1(s_1, v)$.

- Como ϕ é um homomorfismo temos que $g_2(\phi(s_1), u) = g_2(\phi(s_1), v)$, e como M_2 tem ordem k_2 , $u = v$.

Com a análise feita nos sub-itens b.1 e b.2, podemos concluir que se uma das máquinas for de ordem finita a outra também é, e que a ordem de uma é exatamente igual a da outra.

V.2. Observação

Se M_1 é sem perda de informação, M_2 não é necessariamente sem perda de informação. Vejamos a máquina M_1 representada pela sua tabela de transições na figura 15, onde $S_1 = \{s, p, q\}$ e $\Sigma = \Delta = \{0, 1\}$. Analisando M_1 , vemos que a mesma não é reduzida, pois p é equivalente a q . Aplicando os processos de redução, ver KOHAVI¹, página 286, obtemos a máquina reduzida M_2 equivalente a M_1 , representada na figura 16, que é com perda de informação.

VI. INTERCONEXÕES DE MÁQUINAS SEQUENCIAIS SEM PERDA DE INFORMAÇÃO

Sejam $M_i = \langle \Sigma_i, S_i, \Delta_i, f_i, g_i \rangle$, $i = 1, 2$ duas máquinas sequenciais arbitrárias das quais estudaremos o comportamento sob as conexões série, paralela, cascata e com realimentação.

VI.1. Conexão Série (Ver BOOTH², página 118)

VI.1.1. Definição

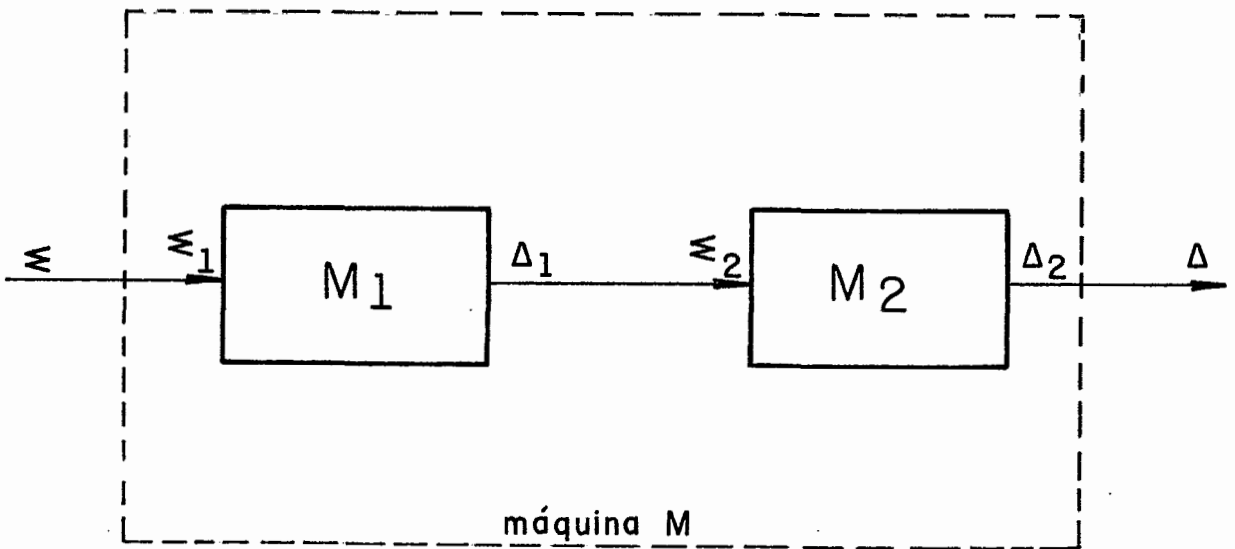


Figura 17

Na figura 17 ilustramos uma conexão série de M_1 com M_2 , com $\Delta_1 = \Sigma_2$, resultando uma máquina $M = \langle \Sigma, S, \Delta, f, g \rangle$ onde $S \cong S_1 \times S_2$, $\Sigma = \Sigma_1$ e $\Delta = \Delta_2$ e que para todo $(s_1, s_2), (t_1, t_2) \in S$, $u \in \Sigma$, $y \in \Delta$, $y_1 \in \Delta_1$, $s_1, t_1 \in S_1$, $s_2, t_2 \in S_2$, $f((s_1, s_2), u) = (t_1, t_2)$ e $g((s_1, s_2), u) = y$ sendo que $f_1(s_1, u) = t_1$, $g_1(s_1, u) = y_1$,

$$f_2(s_2, y_1) = t_2 \quad \text{e} \quad g_2(s_2, y_1) = y.$$

VI.1.2. Proposição

- a) Se M_1 e M_2 são máquinas sem perda de informação, então M também é.
- b) Se M_1 é sem perda de informação de ordem finita k_1 e M_2 sem perda de informação de ordem finita k_2 , M é sem perda de informação de ordem finita $k \leq k_1 + k_2 - 1$.
- c) Se M é sem perda de informação, então M_1 é sem perda de informação.
- d) Se M tem ordem finita k , M_1 tem ordem finita $k_1 \leq k$.

- Demonstração

- a) Se M_2 é sem perda de informação, então $s_2 \in S_2$, $t_2 \in S_2$, $y \in \Delta_2^* \rightarrow y_1 \in \Sigma_2^*$, para s_2 estado inicial e t_2 estado final.

Se M_1 é sem perda de informação, então $s_1 \in S_1$, $t_1 \in S_1$, $y_1 \in \Delta_1^* \rightarrow u_1 \in \Sigma_1^*$, para s_1 estado inicial e t_2 estado final.

Em M teremos (s_1, s_2) , (t_1, t_2) , $y \rightarrow u_1 \in \Sigma^*$ para (s_1, s_2) , $(t_1, t_2) \in S$.

Logo M é sem perda de informação.

b) Se M_2 tem ordem finita k_2 , então $s_1 \in S_2$, $y \in \Delta_2^{k_2} \rightarrow z_1 \in \Sigma_2$, onde z_1 é o primeiro símbolo de sequência de entrada de M_2 .

Se M_1 tem ordem finita k_1 , então $t_1 \in S_1$, $z \in \Delta_1^{k_1} \rightarrow u_1 \in \Sigma_1$, onde u_1 é o primeiro símbolo da sequência de entrada de M_1 .

Em M_1 conectado em série com M_2 , podemos obter u_1 pelo seguinte processo de decodificação:

$$s_1, y_1 y_2 y_3 \cdots y_{k_2} \rightarrow z_1$$

$$s_1, z_1 \rightarrow s_2$$

$$s_2, y_2 y_3 y_4 \cdots y_{k_2} \rightarrow z_2$$

$$s_2, z_2 \rightarrow s_3$$

$$s_3, y_3 y_4 y_5 \cdots y_{k_2+2} \rightarrow z_3$$

$$s_3, z_3 \rightarrow s_4$$

e assim por diante até que :

$$s_{k_1}, y_{k_1} y_{k_1+1} y_{k_1+2} \cdots y_{k_2+k_1-1} \rightarrow z_{k_1}$$

onde agora em M_1

$$t_1, z_1 z_2 z_3 \cdots z_{k_1} \rightarrow u_1$$

Então em M teremos $(t_1, s_1), y_1 y_2 y_3 \cdots y_{k_2+k_1-1} \rightarrow u_1$, ou seja, para determinarmos u_1 , devemos esperar no máximo $k = k_1 + k_2 - 1$ símbolos de saída; logo M é de ordem finita $k \leq k_2 + k_1 - 1$.

c) Imaginando que M_1 é com perda de informação temos que $t_1 \in S_1, u_1 \in \Sigma_1 \rightarrow t_2 \in S_1, z_1 \in \Delta_1$ e $t_1, u_2 \in \Sigma_1 \rightarrow t_2, z_1$ onde $u_1 \neq u_2$.

Neste caso, para qualquer estado s_1 e s_2 de M_2 , temos $s_1, z_1 \rightarrow s_2, y_1 \in \Delta_2$, fazendo com que em M tenhamos $(t_1, s_1), u_1 \rightarrow (t_2, s_2), y_1$ e $(t_1, s_1), u_2 \rightarrow (t_2, s_2), y_1$, o que nos leva a concluir que M será com perda de informação, contrariando a hipótese inicial.

Logo, se M for sem perda de informação, M_1 também é.

d) Fixando um estado $s_2 \in S_2$ e supondo que $s_1, u \rightarrow z$ em M_1 com $u = u_1 \dots u_k \in \Sigma_k$, vamos mostrar que $s_1, z \rightarrow u_1$.

Em M_2 $s_2, z \rightarrow y$. Como M tem ordem $k, (s_1, s_2), y \rightarrow u_1$. Portanto, em $M_1, s_1, z \rightarrow u_1$.

VI.1.3. Observações

a) A máquina inversa de M pode ser representada como mostra a figura 18.

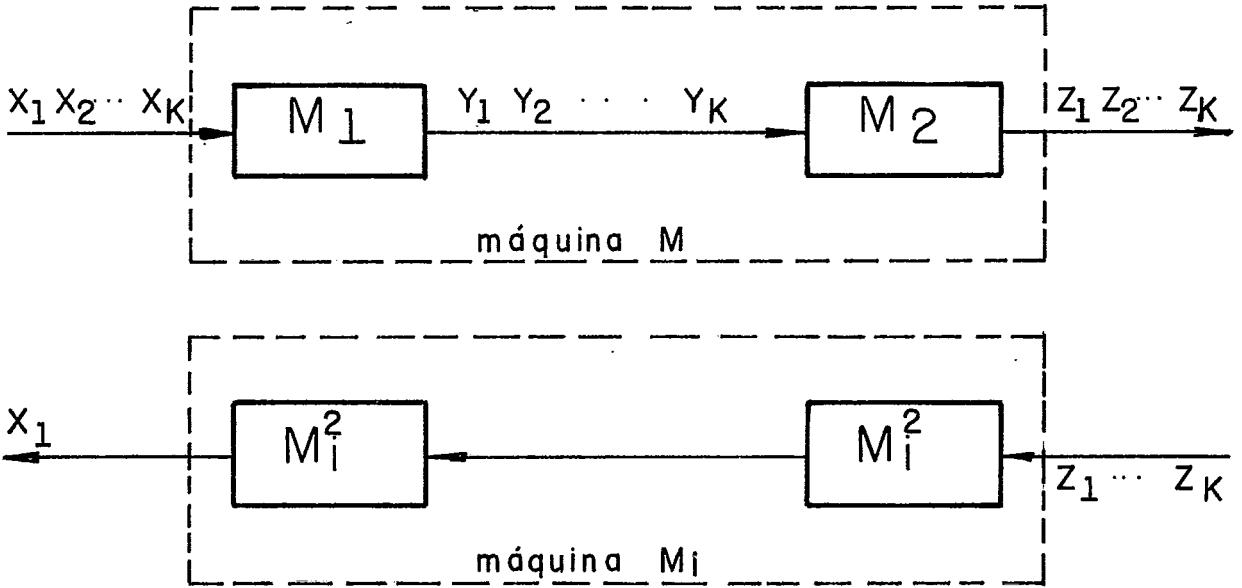


Figura 18

b) Sendo M uma máquina sem perda de informação de ordem finita ou não, M_2 pode ou não ser sem perda de informação. Senão vejamos: seja a máquina M sem perda de informação representada na figura 19.a, que é a conexão série de M_1 com M_2 , também representadas na figura 12.a. As máquinas M_1 e M , como vemos em suas tabelas de teste são sem perda de informação e a máquina M_2 é com perda de informação. Já mostramos também anteriormente que se M_2 é sem perda de informação, M é sem perda de informação se e somente se M_1 também for. Diante disto é claro que se M é sem perda de informação M_2 pode ou não ser.

c) No exemplo da figura 19.a temos que o alfabeto de saída da máquina M_1 é diferente do alfabeto de entrada da máquina M_2 , isto é, em Σ_2 ocorre o símbolo c que não ocorre em Σ_1 . Esta é a maneira mais fácil de ver a observação do item b. Esta mes

	a	b	c
A	A/d	A/b	A/a

máquina M_1

$$\Sigma_1 = \{a, b, c\} \quad \Delta_1 = \{a, b, d\}$$

	a	b	d
A	A	A	A

tabela de teste de M_1

	a	b	c	d
B	B/c	B/b	B/a	B/a

máquina M_2

$$\Sigma_2 = \{a, b, c, d\} \quad \Delta_2 = \{a, b, c\}$$

	a	b	c
B	BB	B	B

tabela de teste de M_2

	a	b	c
AB	AB/b	AB/c	AB/d

máquina M

$$\Sigma = \{a, b, c\} \quad \Delta = \{b, c, d\}$$

	b	c	d
AB	AB	AB	AB

tabela de teste de M

Figura 19a

X	0	1
A	B/a	A/a
B	A/b	B/b
C	C/c	D/c
D	D/d	C/d

máquina M_1

$$\Xi_1 = \{0, 1\} \quad S_1 = \{A, B, C, D\}$$

$$\Delta_1 = \{a, b, c, d\}$$

	a	b	c	d
E	F/a	G/a	F/b	G/c
F	G/b	F/c	G/a	F/d
G	F/b	G/c	G/d	G/a

máquina M_2

$$\Xi_2 = \{a, b, c, d\}$$

$$\Delta_2 = \{a, b, c, d\}$$

$$S_2 = \{E, F, G\}$$

	0	1
(AE)	(BF)/a	(AF)/a
(AF)	(BG)/b	(AG)/b
(AG)	(BF)/b	(AF)/b
(BE)	(AG)/a	(BG)/a
(BF)	(AF)/c	(BF)/c
(BG)	(AG)/c	(BG)/c
(CE)	(CF)/b	(DF)/b
(CF)	(CG)/a	(DG)/a
(CG)	(CG)/d	(DG)/d
(DE)	(DG)/c	(CG)/c
(DF)	(DF)/d	(CF)/d
(DG)	(DG)/a	(CG)/a

máquina M

$$\Xi = \{0, 1\} \quad \Delta = \{a, b, c, d\}$$

$$S = \left\{ \begin{array}{l} (AE), (AF), (AG), (BE), (BF), (BG) \\ (CE), (CF), (CG), (DE), (DF), (DG) \end{array} \right\}$$

Y	a	b	c	d
A	AB	-	-	-
B	-	AB	-	-
C	-	-	CD	-
D	-	-	-	CD
AB	-	-	-	-
CD	-	-	-	-

tabela de teste de M_1

Y	a	b	c	d
E	FG	F	G	-
F	G	G	F	F
G	G	F	G	G
FG	GG	FG	FG	FG

tabela de teste de M_2

Y	a	b	c	d
(AE)	(AF)(BF)	-	-	-
(AF)	-	(AG)(BG)	-	-
(AG)	-	(AF)(BF)	-	-
(BE)	(AG)(BG)	-	-	-
(BF)	-	-	(AF)(BF)	-
(BG)	-	-	(AG)(BG)	-
(CE)	-	(CF)(DF)	-	-
(CF)	(CG)(DG)	-	-	-
(CG)	-	-	-	(CG)(DG)
(DE)	-	-	(CG)(DG)	-
(DF)	-	-	-	(CF)(DF)
(DG)	(CG)(DG)	-	-	-
(BF)(BF)	-	-	-	-
(AG)(BG)	-	-	-	-
(CF)(DF)	-	-	-	-
(CG)(DG)	-	-	-	-

tabela de teste de M

Figura 19b

ma observação pode ser verificada com $\Delta_1 = \Sigma_2$, isto é, construímos uma máquina M_2 com perda de informação com sequências pertencentes a Σ_2^* que não pertencem a Δ_1^* , conforme o exemplo da figura 19.b. Na máquina M_2 temos as sequências de entrada ac e bd, que quando o estado inicial for E produzimos a sequência de saída aa e vamos para o estado final G. Se observarmos a máquina M_1 notaremos que a mesma nunca produzirá estas duas sequências de saída ac e bd.

Então quando M_1 estiver conectada em série com M_2 teremos a máquina M sem perda de informação conforme podemos verificar pela figura 19.b.

VI.2. Considerações a respeito de circuito combinacional

Um circuito combinacional é um circuito com somente um estado, isto é, que não tem nenhuma memória. Para um circuito combinacional o símbolo de entrada (ou uma combinação de símbolos) para um dado instante determina o símbolo de saída (ou uma combinação de símbolos). Pode ser representado conforme a figura 20.

Um circuito combinacional sem perda de informação é definido como sendo aquele que tem a propriedade adicional de permitir que a saída determine a entrada. E neste caso dizemos que não há combinações diferentes de símbolos de entrada produzindo uma mesma combinação de símbolos de saída. Então, se $x \rightarrow y$ e $y \rightarrow x$, dizemos que o circuito é sem perda de informação.

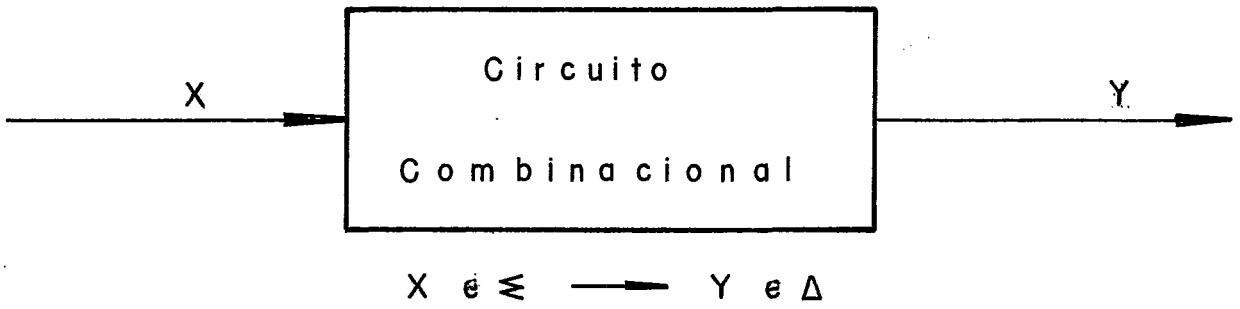


Figura 20

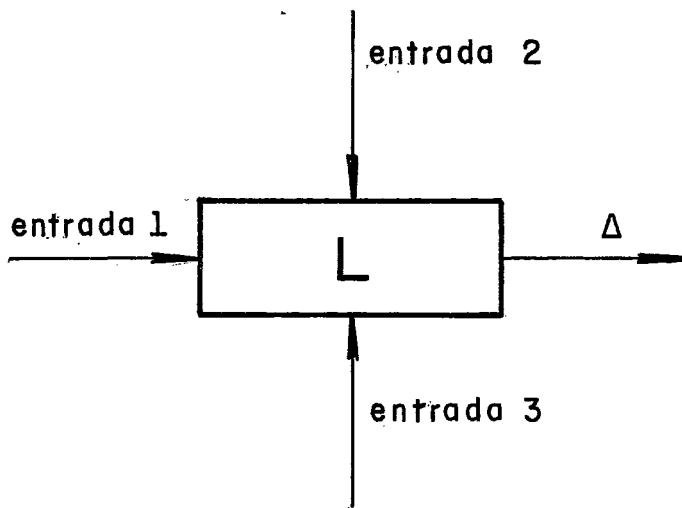


Figura 21

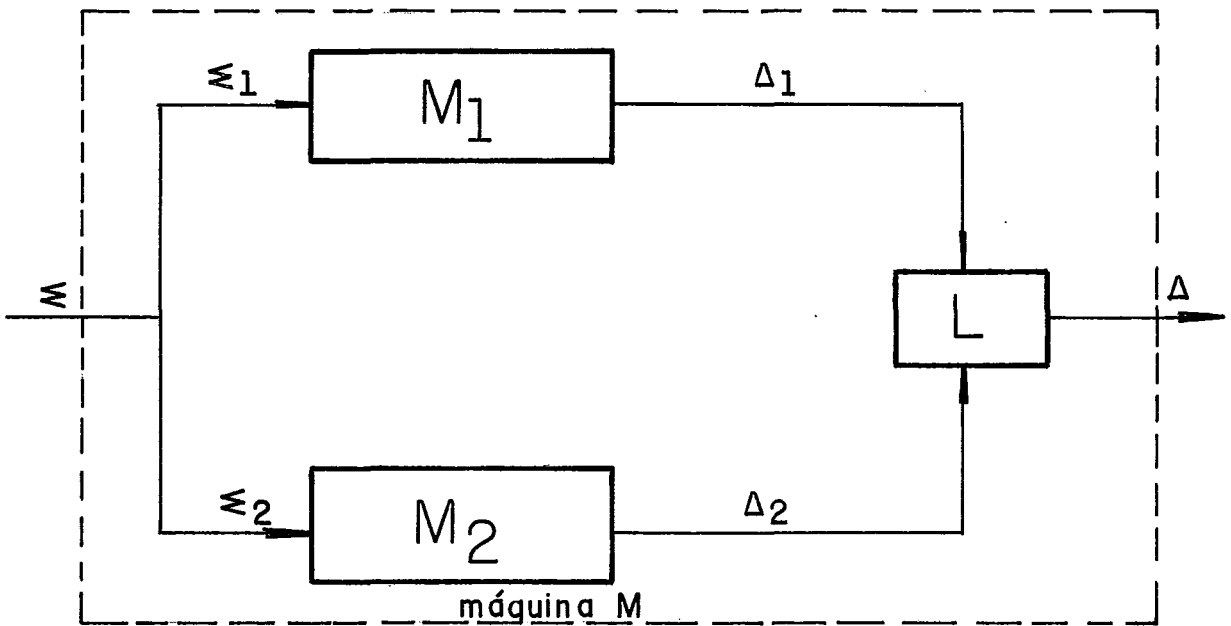


Figura 22

Na figura 21 temos a representação do tipo de circuito combinacional que usaremos neste trabalho. Nós dizemos que o circuito L da figura 21 é parcialmente sem perda de informação em relação a entrada 1 se a saída determinar a entrada 1. Dizemos também que L é um circuito controlado pela entrada 2 se para cada entrada 2 a saída determinar a entrada 1. As mesmas considerações podemos fazer levando em consideração as demais entradas.

VI.3. Conexão Paralela (ver BOOTH , página 118)

VI.3.1. Definição

A conexão paralela de M_1 com M_2 é ilustrada na figura 22. Neste caso supomos que as entradas de ambas as máquinas são as mesmas e que a saída da máquina M, resultante desta conexão, é formada pela combinação lógica das saídas de M_1 e M_2 . Então, $S = S_1 \times S_2$, $\Sigma = \Sigma_1 = \Sigma_2$, $L: \Delta_1 \times \Delta_2 \rightarrow \Delta$, e para todo $(t_1, s_1), (t_2, s_2) \in S$, $u \in \Sigma$, $z \in \Delta$, $t_1, t_2 \in S_1, s_1, s_2 \in S_2$, $y_1 \in \Delta_1, y_2 \in \Delta_2, (t_1, s_1), u \rightarrow (t_2, s_2), z$ onde $z = L(y_1, y_2)$, $t_1, u \rightarrow t_2, y_1 \in s_1, u \rightarrow s_2, y_2$.

VI.3.2. Proposição

a) Se M_J é sem perda de informação, L é um circuito parcialmente sem perda de informação em relação a saída da máquina M_J ; então M é sem perda de informação. ($J = 1, 2$).

b) Se M_J é de ordem finita k_J e L é parcialmente sem perda de informação em relação a saída da máquina M_J , então M é de ordem finita $k \leq k_J$.

- Demonstração

a) Considerando o caso em que $J = 1$, em L temos para $u \in \Delta_1$, $v \in \Delta_2$ e $y \in \Delta$ que $u, v \rightarrow y$ e $y \rightarrow u$. Em M_1 temos para $t_1, t_2 \in S_1$, $w \in \Sigma_1$ que $t_1, t_2, u \rightarrow w$, ficando então em M $(t_1, s_1), (t_2, s_2), y \rightarrow w$. Logo, M é sem perda de informação. O mesmo resultado pode ser obtido para $J = 2$.

b) Se M_1 for de ordem finita k_1 , temos para $t_1 \in S_1, x_1 \in \Sigma_1$, $y = y_1 y_2 y_3 \dots y_{k_1} \in \Delta_1^{k_1}$, que $t_1, y \rightarrow x_1$.

Se L é parcialmente sem perda de informação com relação a saída da máquina M_1 temos para $z_1 \in \Delta_2$, $z \in \Delta$ $y_1, z_1 \rightarrow z$ e $z \rightarrow y_1$.

Logo, temos M de ordem finita $k \leq k_1$ pois para $s_1 \in S_2$ teremos $(t_1, s_1) \in S$, $z_1 z_2 z_3 \dots z_{k_1} \rightarrow x_1$, uma vez que para $z \in \Delta^{k_1}$ em L $z_1 \rightarrow y_1, z_2 \rightarrow y_2, z_3 \rightarrow y_3, \dots, z_{k_1} \rightarrow y_{k_1}$ e em M_1 $t_1, y_1 y_2 \dots y_{k_1} \rightarrow x_1$.

De modo análogo, mostramos que chegaremos em M de ordem finita $k \leq k_2$ se L for parcialmente sem perda de informação com relação a Δ_2 e M_2 for sem perda de informação de ordem finita k_2 .

	0	1
A	A/0	B/0
B	A/0	B/1

máquina M_1

	0	1
C	C/1	D/1
D	C/0	D/1

máquina M_2

$Y_1 \in \Delta_1$	$Z_1 \in \Delta_2$	$L(Y_1, Z_1)$
0	1	0
0	0	1
1	1	1
1	0	0

circuito L

	0	1
AC	AC/0	BD/0
AD	AC/1	BD/1
BC	AC/0	BD/1
BD	AC/1	BD/1

máquina M

Figura 23

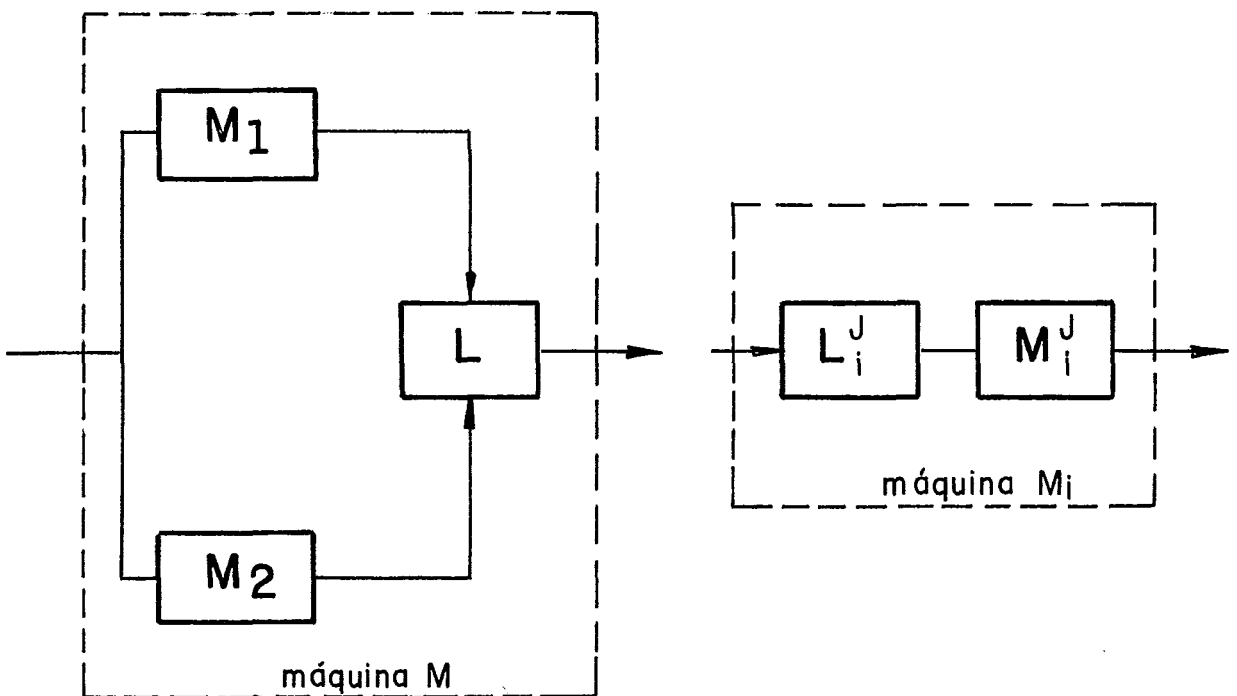


Figura 24

VI.3.3. Observações

Se M for sem perda de informação, de ordem finita ou não, M_1 , M_2 e L não precisam ser sem perda de informação ou de ordem finita. Por exemplo vejamos as máquinas M_1 , M_2 , M e o circuito L representados na figura 23. Vemos que M , conexão paralela de M_1 e M_2 com circuito lógico L , é sem perda de informação embora M_1 , M_2 e L sejam com perda de informação.

A máquina inversa de M , sendo M a conexão paralela de M_1 e M_2 com saída lógica L , é mostrada na figura 24. Na máquina inversa M_i temos o inverso do circuito lógico L que representamos por L_i^J , e a inversa de uma das máquinas que representamos por M_i^J , e neste caso podemos ter J igual a 1 ou 2. Para $J = 1$, $L_i^1(L(y_1, z_1)) = y_1$ e para $J = 2$ $L_i^2(L(y_1, z_1)) = z_1$. L_i^J representa o circuito inverso do circuito lógico L que é sem perda de informação com relação a saída da máquina M_J .

VI.4. Conexão Cascata (ver BOOTH², página 143)

VI.4.1. Definição

A figura 25 ilustra uma conexão cascata de M_1 com M_2 com interconexões lógicas $I: \Sigma \rightarrow \Sigma_1$, $J: \Sigma \times \Delta_1 \rightarrow \Sigma_2$, $L: \Delta_1 \times \Sigma \times \Delta_2 \rightarrow \Delta$. Dando como resultado uma máquina M onde $S = S_1 \times S_2$, e para todo $u \in \Sigma$, (t_1, s_1) , $(t_2, s_2) \in S$, $y \in \Delta$, $(t_1, s_1), u \rightarrow (t_2, s_2), y$. Sendo que para $t_1, t_2 \in S_1$, $s_1 s_2 \in S_2, w \in \Delta_1$ $v \in \Delta_2$, $t_1, I(u) \rightarrow t_2, w, s_1, J(u, w) \rightarrow t_2, v$, e $y = L(w, u, v)$.

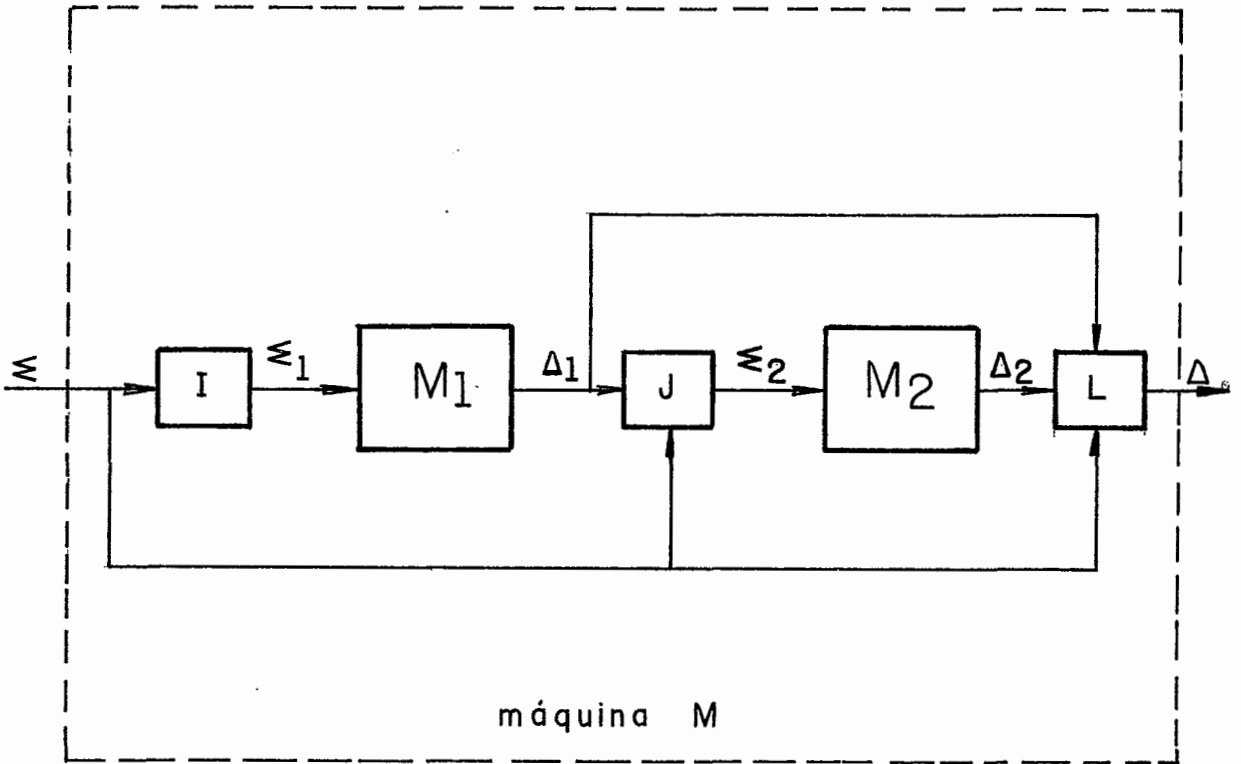


Figura 25

VI.4.2. Proposição

a) M será sem perda de informação se :

1. L for parcialmente sem perda de informação em relação a entrada do circuito I, ou
2. L for sem perda de informação em relação a saída da máquina M_1 , e M_1 e I forem sem perda de informação, ou
3. L for sem perda de informação em relação a saída da máquina M_2 , M_2 for sem perda de informação e J for sem perda de informação com relação a entrada do circuito I, ou
4. L for sem perda de informação com relação a saída da máquina M_2 , M_2 for sem perda de informação, J for sem perda de informação com relação a saída da máquina M_1 , M_1 e I forem sem perda de informação.

b) M será de ordem finita se :

1. L for parcialmente sem perda de informação com relação a entrada do circuito I, e neste caso M tem ordem finita igual a 1, ou
2. L for sem perda de informação com relação a saída da máquina M_1 , M_1 for de ordem finita k_1 e I for sem perda de informação, ou
3. L for sem perda de informação com relação a saída da máquina M_2 , M_2 for de ordem finita k_2 e J for sem perda de informação com relação a entrada do circuito I, ou
4. Se L for sem perda de informação com relação a saída da máquina M_2 , M_2 for de ordem finita k_2 , J sem perda de informação em relação a saída da máquina M_1 , M_1 for de ordem finita k_1 e I sem perda de informação.

- Demonstração

- a.1. Temos que em L para $v_1 \in \Delta_1$, $w_1 \in \Delta_2$, $u \in \Sigma$, $y \in \Delta$, $v_1, w_1, u \rightarrow y$ e $y \rightarrow u$. Em M_1 $t_1 \in S_1$, $I(u) \in \Sigma_1 \rightarrow t_2 \in S_1$, v_1 e em M_2 $s_1 \in S_2$, $J(v_1, u) \in \Sigma_2 \rightarrow s_2 \in S_2$, w_1 . E daí então M será sem perda de informação pois teremos $(t_1, s_1), (t_2, s_2), y \rightarrow u$ onde em L $y \rightarrow u$.
- a.2. Em I, $I(u) \rightarrow x_1$ e $x_1 \rightarrow u$. Em M_1 , $t_1, x_1 \in \Sigma \rightarrow t_2, v_1$ e $t_1, t_2, v_1 \rightarrow x_1$. Em L, $v_1, w_1, u \rightarrow y$ e $y \rightarrow v_1$. Em M_2 , $s_1, J(v_1, u) \rightarrow s_2, w_1$ e em M teremos $(t_1, s_1), (t_2, s_2), y \rightarrow u$, isto é, M sem perda de informação pois por L $y \rightarrow v_1$, por M_1 $t_1, t_2, v_1 \rightarrow x_1$ e por I $x_1 \rightarrow u$.

- a.3. Em J , $v_1, u \rightarrow z_1 \in \Sigma_2$ e $z_1 \rightarrow u$. Em L $v_1, w_1, u \rightarrow y$ e $y \rightarrow w_1$. Em M_2 $s_1, z_1 \rightarrow s_2, w_1$ e $s_1, s_2, w_1 \rightarrow z_1$. Em M_1 $t_1, I(u) \rightarrow t_2, v_1$ e em M teremos $(t_1, s_1), (t_2, s_2), y \rightarrow u$, isto é, M sem perda de informação pois por L , $y \rightarrow w_1$, por M_2 $s_1, s_2, w_1 \rightarrow z_1$ e por J $z_1 \rightarrow u$.
- a.4. Em I , $I(u) \rightarrow x_1$ e $x_1 \rightarrow u$. Em L $v_1, w_1, u \rightarrow y$ e $y \rightarrow w_1$. Em J $v_1, u \rightarrow z_1$ e $z_1 \rightarrow v_1$. Em M_1 $t_1, x_1 \rightarrow t_2, v_1$ e $v_1 \rightarrow x_1$. Em M_2 $s_1, z_1 \rightarrow s_2, w_1$ e $w_1 \rightarrow z_1$ e em M teremos $(t_1, s_1), (t_2, s_2)$, $y \rightarrow u$, isto é, M sem perda de informação pois por L $y \rightarrow w_1$, por M_2 $s_1, w_1 \rightarrow z_1$, por J $z_1 \rightarrow v_1$, por M_1 $t_1, t_2, v_1 \rightarrow x_1$ e por I $x_1 \rightarrow u$.
- b.1. Se L é sem perda de informação com relação a Σ , é evidente que em M cada símbolo da sequência de saída produz um símbolo da sequência de entrada; logo M tem ordem finita $k = 1$.
- b.2. Aqui temos para $y \in \Delta^{k_1} = y_1 y_2 \dots y_{k_1}$, $z \in \Delta_1^{k_1} = z_1 z_2 z_3 \dots z_{k_1}$ em L $y_1 \rightarrow z_1, y_2 \rightarrow z_2, y_3 \rightarrow z_3, \dots, y_{k_1} \rightarrow z_{k_1}$, em M_1 $t_1 \in S_1, z \rightarrow x_1 \in \Sigma_1$ e em I $x_1 \rightarrow u \in \Sigma$. Em M temos para $s_1 \in S_2$ $(t_1, s_1), y \rightarrow u$, isto é, para todo $y \in \Delta^{k_1}$ com um estado inicial $(t_1, s_1) \in S$ determinamos o primeiro símbolo da sequência de entrada.
- b.3. Para $y \in \Delta^{k_2} = y_1 y_2 \dots y_{k_2}$, $z \in \Delta_2^{k_2} = z_1 z_2 \dots z_{k_2}$ em L $y_1 \rightarrow z_1, y_2 \rightarrow z_2, y_3 \rightarrow z_3, \dots, y_{k_2} \rightarrow z_{k_2}$. Em M_2 $s_1, z \rightarrow x_2 \in \Sigma_2$. Em J $x_2 \rightarrow u \in \Sigma$. Em M ficamos com $(t_1 \in S_1, s_1 \in S_2), y \rightarrow u$, ou seja, uma máquina de ordem finita k_2 .
- b.4. Para $v_i \in \Sigma_2, w_i \in \Delta, z_i \in \Delta_2, y_i \in \Delta_1, s_i \in S_2$, e $i = 1, 2$,

1	0
0	0

circuito I

	0	1
A	A/1	A/1

máquina M_1

	0	1
B	B/0	B/0

máquina M_2

Δ_1	\approx	\approx_2
0	0	1
1	0	1
0	1	0
1	1	1

circuito J

Δ_1	Δ_2	\approx	Δ
0	0	0	1
0	0	1	1
0	1	0	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1
0	1	1	0

circuito K

	0	1
AB	AB/1	AB/0

máquina M

Figura 26

3 ..., em L $w_1 \rightarrow z_1, w_2 \rightarrow z_2, w_3 \rightarrow z_3, w_4 \rightarrow z_4, \dots, w_{k_2} \rightarrow z_{k_2}$.
 Em M_2 $s_1, z_1 z_2 \dots z_{k_2} \rightarrow v_1, s_1, v_1 \rightarrow s_2$. Em J $v_1 \rightarrow y_1$.
 Novamente em L $w_{k_2+1} \rightarrow z_{k_2+1}$. Em M_2 $s_2, z_2 z_3 \dots z_{k_2+1} \rightarrow v_2$ e $s_2, v_2 \rightarrow s_3$. Em J $v_2 \rightarrow y_2$. E assim por diante como no caso da conexão série, até que em L $w_{k_2+k_1-1} \rightarrow z_{k_2+k_1-1}$, em M_2 $s_{k_1}, z_{k_2} z_{k_2+1} \dots z_{k_2+k_1-1} \rightarrow v_{k_1}$, em J $v_{k_1} \rightarrow y_{k_1}$. E agora em M_1 $t_1, y_1 y_2 \dots y_{k_1} \rightarrow x_1 \in \Sigma_1$ e em I $x_1 \rightarrow u_1 \in \Sigma$. E aí então em M temos $(t_1, s_1), y_1 y_2 \dots y_{k_1+k_2-1} \rightarrow u_1$, ou seja, uma máquina de ordem finita $k_1 + k_2 - 1$.

VI.4.3. Observações

Se M é sem perda de informação, de ordem finita ou não, I, J, L, M_1 e M_2 não precisam ser. Por exemplo vejamos as máquinas representadas na figura 26. A máquina M é sem perda de informação embora sendo uma conexão cascata de I, J, L, M_1 e M_2 , em que todos são com perda de informação.

VI. 5. Conexão com Realimentação (ver BOOTH², página 118)

VI.5.1. Definição

Supondo $\Delta_1 = \Sigma_2$ e que M_2 é uma máquina de Moore com função de Moore $h: S_2 \rightarrow \Delta_2$, (isto é, $h(s) = g_2(s, y_1 \in \Sigma_2)$), a conexão com realimentação de M_1 com M_2 produz uma máquina $M = \langle \Sigma, S, \Delta, f, g \rangle$ representada na figura 27 onde $S = S_1 \times S_2$, $\Delta = \Delta_1$, $L: \Sigma \times \Delta_2 \rightarrow \Sigma_1$ e que para $u \in \Sigma, t_1, t_2 \in S_1, s_2 \in S_2, (t_1, s_2), u \rightarrow y$ em M e $t_1, L(u, h(s_2)) \rightarrow t_2, y$ em M_1 e $s_2, y \rightarrow s_3$ em M_2 .

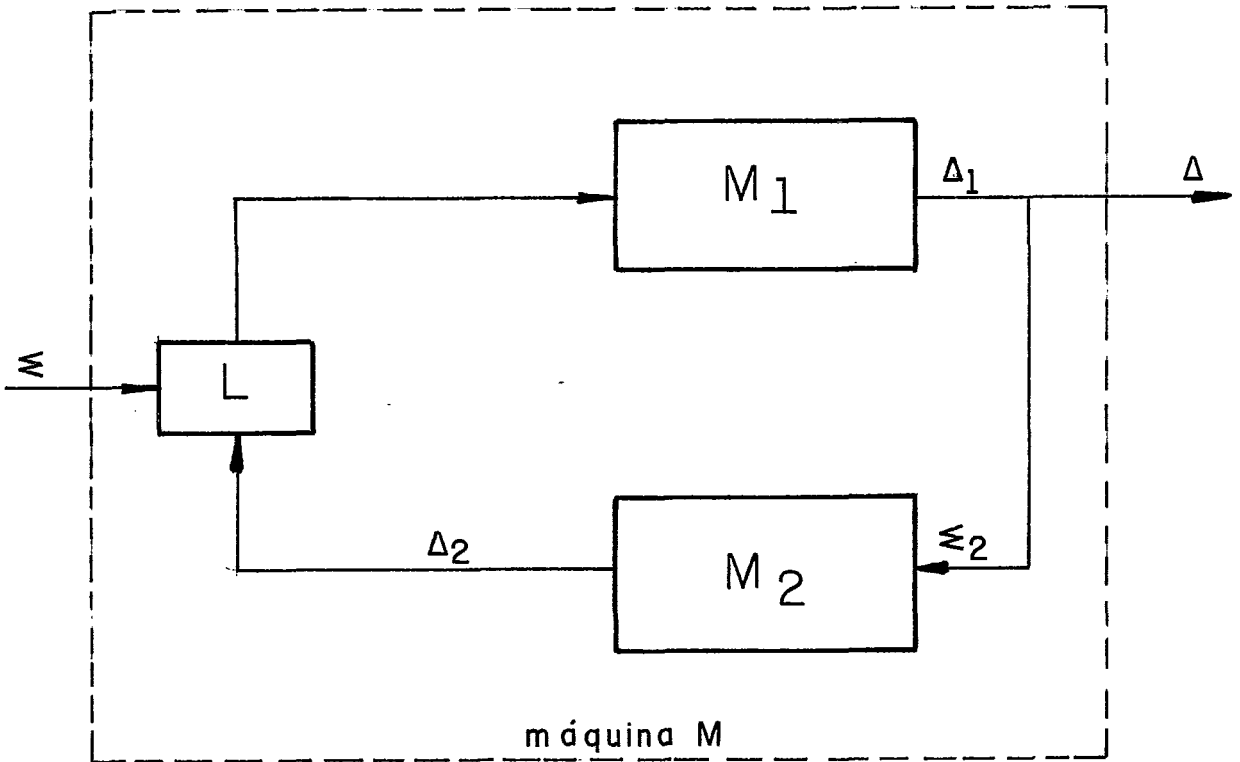


Figura 27

VI.5.2. Proposição

- a) Se L é um circuito combinacional sem perda de informação controlado pela saída de M_2 , M_2 sem perda de informação (de ordem finita), então M é sem perda de informação (de ordem finita $k = k_1$).
- b) Se M é sem perda de informação (de ordem finita), então L é um circuito controlado pela saída da máquina M_2 desde que a função h seja sobrejetora.
- c) Se M é sem perda de informação (de ordem finita), então M_1 é sem perda de informação (de ordem finita) desde que $L_z = L(.,z): \Sigma \rightarrow \Sigma_1$ seja uma função sobrejetora para cada $z \in \Delta_2$.

- Demonstração

a) Em M temos $(t_1, s_1), u \in \Sigma \rightarrow (t_2, s_2), y$.

Em M_1 $t_1, x \rightarrow t_2, y_1$ e $t_1, t_2, y_1 \rightarrow x_1$ pois M_1 é sem perda de informação.

Sendo L sem perda de informação controlado pela saída da máquina M_2 , temos que $u \in \Sigma, h(s_1) \rightarrow x$ e $h(s_1), x \rightarrow u$.

Em M_2 $s_1, y_1 \rightarrow s_2, h(s_1)$ e aí então $(t_1, s_1), (t_2, s_2), y \rightarrow u$, isto é, M é sem perda de informação.

Agora se M_1 é sem perda de informação de ordem finita k_1 temos que $t_1, x \rightarrow y$ e $t_1, y \rightarrow x_1$ para $y = y_1 y_2 \dots y_{k_1} \in \Delta_1^{k_1}$ e $x = x_1 x_2 \dots x_{k_1} \in \Sigma_1^{k_1}$.

Em M_2 $s_1, y_1 \rightarrow h(s_1)$, em $L, h(s_1), x_1 \rightarrow u$ e em M $(t_1, s_1), y \rightarrow u$ e aí então M é de ordem finita $k \leq k_1$.

Se M tem ordem finita k então $(t_1, s_1), y \rightarrow v_1 \in \Sigma$ para $y = y_1 y_2 \dots y_k \in \Delta_1^k$ e em L $v_1, h(s_1) \rightarrow v_1 \in \Sigma_1$ e com isto em M_1 $t_1, y \rightarrow u_1$ e $k_1 \leq k$. Logo $k = k_1$.

b) Se L não é um circuito sem perda de informação controlado por Δ_2 , temos que para $u, v \in \Sigma, s_1 \in S_2, x_1 \in \Sigma_1, u = v$ e h uma função sobrejetora, $u, h(s_1) \rightarrow x_1$ e $v, h(s_1) \rightarrow x_1$. Em M_1 para $t_1, t_2 \in S_1$ e $y_1 \in \Delta_1, t_1, x_1 \rightarrow t_2, y_1$. Em M_2 $s_1, y_1 \rightarrow s_2 \in S_2$. Daí em M teremos $(t_1, s_1), u \rightarrow (t_2, s_2), y_1$, $(t_1, s_1), v \rightarrow (t_2, s_2), y_1$ e $u = v$. Logo, para uma máquina M sem perda de informação, L tem que ser sem perda de informação por Δ_2 .

c) Sendo M_1 uma máquina que não é de ordem finita k , então para

$k > 0$, existe $z = z_1 z_2 \dots z_k \in \Delta_1^k$, $s_0 \in S_1$, $x = x_1 x_2 \dots x_k \in \Sigma_1^k$, $y = y_1 y_2 \dots y_k \in \Sigma_1$ tal que $s_0, x \rightarrow z$ e $s_0, y \rightarrow z$ e $x_1 = y_1$. Para todo $t_0 \in S_2$, como $L_{h(t_0)}$ é sobrejetora, existe $u_1, v_1 \in \Sigma$ tal que $u_1, h(t_0) \rightarrow x_1$, $v_1, h(t_0) \rightarrow y_1$ em L e $x_1 = y_1$ e $u_1 = v_1$.

Sendo $t_1 = f_2(t_0, z_1)$, existe $u_2, v_2 \in \Sigma$ tal que $u_2, h(t_1) \rightarrow x_2$ e $v_2, h(t_1) \rightarrow y_2$ e assim por diante, sendo $t_i = f_2(t_{i-1}, z_i)$, existe $u_{i+1}, v_{i+1} \in \Sigma$ tal que $u_{i+1}, h(t_i) \rightarrow x_{i+1}$ e $v_{i+1}, h(t_i) \rightarrow y_{i+1}$, para $i < k$.

Então para $u \in \Sigma^k$, $v \in \Sigma^k$ temos em $M(S_0, t_0)$, $u \rightarrow z$ e $(S_0, t_0), v \rightarrow z$, isto é, M não é de ordem finita.

Logo se M é de ordem finita M_1 é de ordem finita desde que L_z seja sobrejetora.

De modo análogo podemos mostrar o caso de M ser sem perda de informação.

VII. RELAÇÃO ENTRE MÁQUINAS SEM PERDA DE INFORMAÇÃO E MÁQUINAS DE MEMÓRIA FINITA

VII.1. Máquinas de Memória Finita

Uma máquina reduzida $M = \langle \Sigma, S, \Delta, f, g \rangle$ é de memória finita J se J é o menor inteiro, tal que o estado final S_f é determinado a partir do conhecimento dos últimos J símbolos da sequência de entrada e dos últimos J correspondentes símbolos da sequência de saída. (Ver KOHAVI¹, página 445).

Então se $y \in \Delta^J = y_1 y_2 \dots y_J$ e $x \in \Delta^J = x_1 x_2 \dots x_J$, $(x_1 x_2 \dots x_J), (y_1 y_2 \dots y_J) \rightarrow S_f$.

A condição para que M seja de memória finita J é que, se para um qualquer estado inicial de M aplicarmos uma sequência de entrada de comprimento J , transferirmos M para um estado conhecido, e alguma sequência de entrada de comprimento $(J-1)$ com sua correspondente sequência de saída não são informações suficientes para identificar o estado final, isto é, para $S_i, t_i \in S$ ($i = 1, \dots, J$), temos o exposto na figura 28.

Analisando a partir da figura 28, vemos ser evidente que $J \leq n(n-1)/2$, onde n é o número de estados da máquina, (ver GILL¹, página 2.24). Se $S_i = t_i$ teremos que $(x_1 x_2 \dots x_i), (y_1 y_2 \dots y_i) \rightarrow t_i$ e sendo $i < J$ estaríamos indo de encontro à escolha feita. Se $S_i = S_{i+h}$ e $t_i = t_{i+h}$, isto é, $S_i t_i = S_{i+h} t_{i+h}$, ficamos com o exposto na figura 28.a. Se $S_i = t_{i+h}$ e $t_i = S_{i+h}$, isto é, $S_i t_i = S_{i+h} t_{i+h}$, ficamos com o exposto na figura 28.b. Temos em KOHAVI¹, página 446, um algo

ritmo para construção de uma tabela de teste da qual se constrói um grafo de teste, segundo outro algoritmo exposto pelo mesmo autor na página 446, para identificação de uma máquina de memória finita. Neste teste levamos em consideração todos os pares não ordenados e distintos e seus respectivo sucessores. Será de memória finita se seu grafo de teste for livre de ciclos, e será de memória J , se $(J-1)$ for o comprimento do maior caminho do seu grafo (ver KOHAVI¹, página 447). Nas figuras 28.a e 28.b vemos claramente a existência de ciclos no grafo de teste. Em ambas, o par não ordenado $S_i t_i$ está ligado a ele mesmo pela relação entrada/saída. $x_i x_{i+1} \dots x_{i+h-1} y_i y_{i+1} \dots y_{i+h-1}$. Com isto vemos que na figura 28 temos que ter J pares não ordenados e distintos. Sabemos também que com n estados podemos obter $n(n-1)/2$ pares não ordenados e distintos. Logo $J \leq n(n-1)/2$.

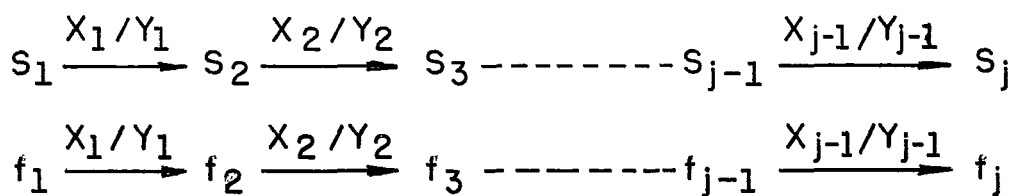
Se FM_i representa o conjunto de máquinas de memória finita de ordem i , $FM_1 \subset FM_2 \subset FM_3 \subset \dots \subset FM_i \subset \dots \subset FM_{J-1} \subset FM_J$. Esta afirmativa é fácil de se comprovar, bastando considerar o seguinte:

1) É fácil de se construir uma máquina de memória finita J , com $(J+1)$ estados, que não é de memória finita $(J-1)$, se seguirmos o algoritmo abaixo:

a) para todo $i < J + 1$

$$S_i, 0 \rightarrow S_{i+1}, 0$$

$$S_i, 1 \rightarrow S_{i+1}, 1$$



onde $s_i \neq f_i$ e $s_i f_i \neq s_{i+h} f_{i+h}$

Figura 28

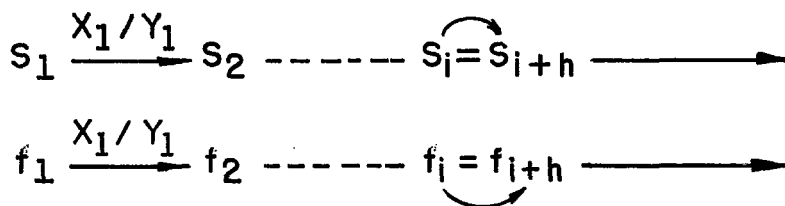


Figura 28a

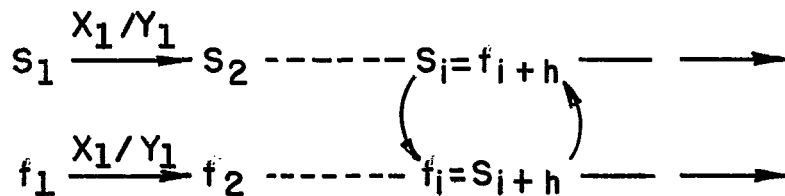


Figura 28 b

	0	1
1	2 / 0	2 / 1
2	3 / 0	3 / 1
3	4 / 0	4 / 1
5	5 / 0	5 / 1
⋮		
⋮		
⋮		
J-1	J / 0	J / 1
J	(J+1) / 0	(J+1) / 1
J+1	1 / 1	1 / 0

Figura 29

b) para $\ell = J + 1$

$$S_i, 0 \rightarrow S_1, 1$$

$$S_i, 1 \rightarrow S_1, 0$$

Se aplicarmos os algoritmos de identificação de uma máquina de memória finita, que encontramos em KOHAVI¹, nas páginas 445 e 446, na máquina da figura 29, veremos que ela tem memória J e não tem memória $(J-1)$. NEWBORN⁷ também em seu trabalho fornece máquinas que são de memória finita J e que não são de memória finita $(J-1)$.

2) Todas as máquinas que são de memória finita J são também de memória finita $(J+1)$.

VII.2. Máquina de Memória Finita e de Ordem Finita

Para M de memória J e ordem finita k com n estados $J + k \leq n(n-1) + 1$, já que tanto J como $(k-1)$ são no máximo $n(n-1)/2$.

VII.3. Proposição

Para todo n e todo $k \leq 1 + (n-1)n/2$ existe uma máquina com n estados de memória finita $J = n(n-1)/2$ e ordem finita k .

- Demonstração

É fácil de se construir uma máquina com n estados, que seja de ordem finita k e memória finita J com J e k máximas em que $J + k - 1 = n(n-1)$.

Na figura 30 temos um grafo de teste de uma máquina com n estados, de ordem finita k e memória finita J , sendo J e k máximas. No grafo da figura 30, considerando apenas as linhas cheias, estamos levando em conta apenas o teste para ordem finita da máquina e considerando as linhas cheias mais as linhas pontilhadas, estamos levando em conta o teste para memória da máquina. A quantidade representada entre colchetes ao lado de cada vértice do grafo representa o número de ordem do par não ordenado representado por aquele vértice. Para qualquer $n < 2$ este grafo pode ser construído seguindo o seguinte algoritmo:

1) Para todo $i < j < n$ colocamos o par $i + 1, j + 1$ como 1 sucessor (sob a saída) do par ij .

2) Para $j = n$

$i = n-1$ 13 é 2 sucessor (sob a saída) de ij

$i = n-2$ 14 é 3 sucessor (sob a saída) de ij

$i = n-3$ 15 é 4 sucessor (sob a saída) de ij

$i = 2$ $1n$ é $(n-1)$ sucessor(sob a saída) de ij

3) Fazemos os estados i e j serem n sucessores (sob a saída) do estado 1, formando assim um par compatível ij que será igual

$\lceil n(n-1)/2 \rceil$ $\lceil n(n-1)/2-1 \rceil$ $\lceil n(n-1)/2-3 \rceil$

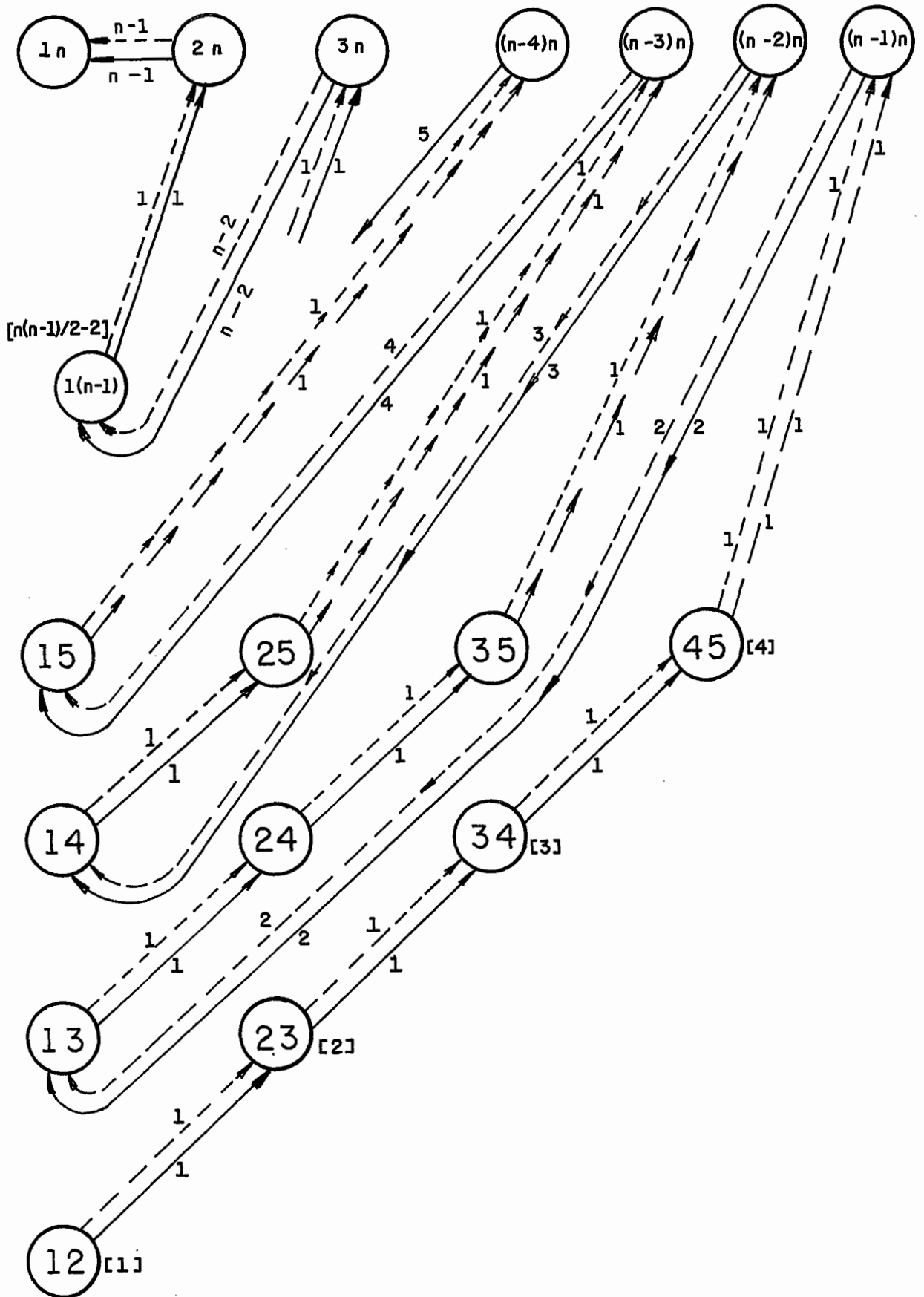


Figura 30

ao $((n(n-1)/2) - (k-2))$ ésimo par se M tiver ordem k.

4) Repetimos os passos 1 e 2 só que agora os pares serão sucessores sob o símbolo de entrada (formando assim as linhas pontilhadas).

Com este algoritmo construímos uma máquina não completamente especificada que é de memória $J = n(n-1)/2$ e ordem finita $k = (n(n-1)/2) + 1$. Completamos a especificação da máquina de maneira conveniente, conforme ficará claro em um exemplo, de modo a não formar nenhum ciclo no grafo de teste.

Para maior esclarecimento do algoritmo anterior façamos um exemplo para $n = 5$.

Com os passos 1 e 2 obtemos o grafo de teste da figura 31 e a tabela de teste da figura 32. Com aplicação do passo 3 fazemos o par de estados cujo número de ordem é $(10 - (11-2)) = 1$ serem 5 sucessores, sob o símbolo de saída do estado 1, para obtermos assim uma memória de ordem finita $k = 11$, formando então a quinta coluna da tabela de teste da figura 32. Com o passo 4 o grafo de teste passa a ser representado como está na figura 33. Com isto obtemos a figura 34 que representa a tabela de teste para memória da máquina, não completamente especificada. Para completarmos a especificação da máquina, devemos acrescentar um estado na última linha, um na primeira linha e dois em cada uma das demais linhas. Para isto devemos acrescentar $(n-1=4)$ símbolos de saída, obtendo assim a máquina completamente especificada da figura 35, por sua tabela de teste de memória.

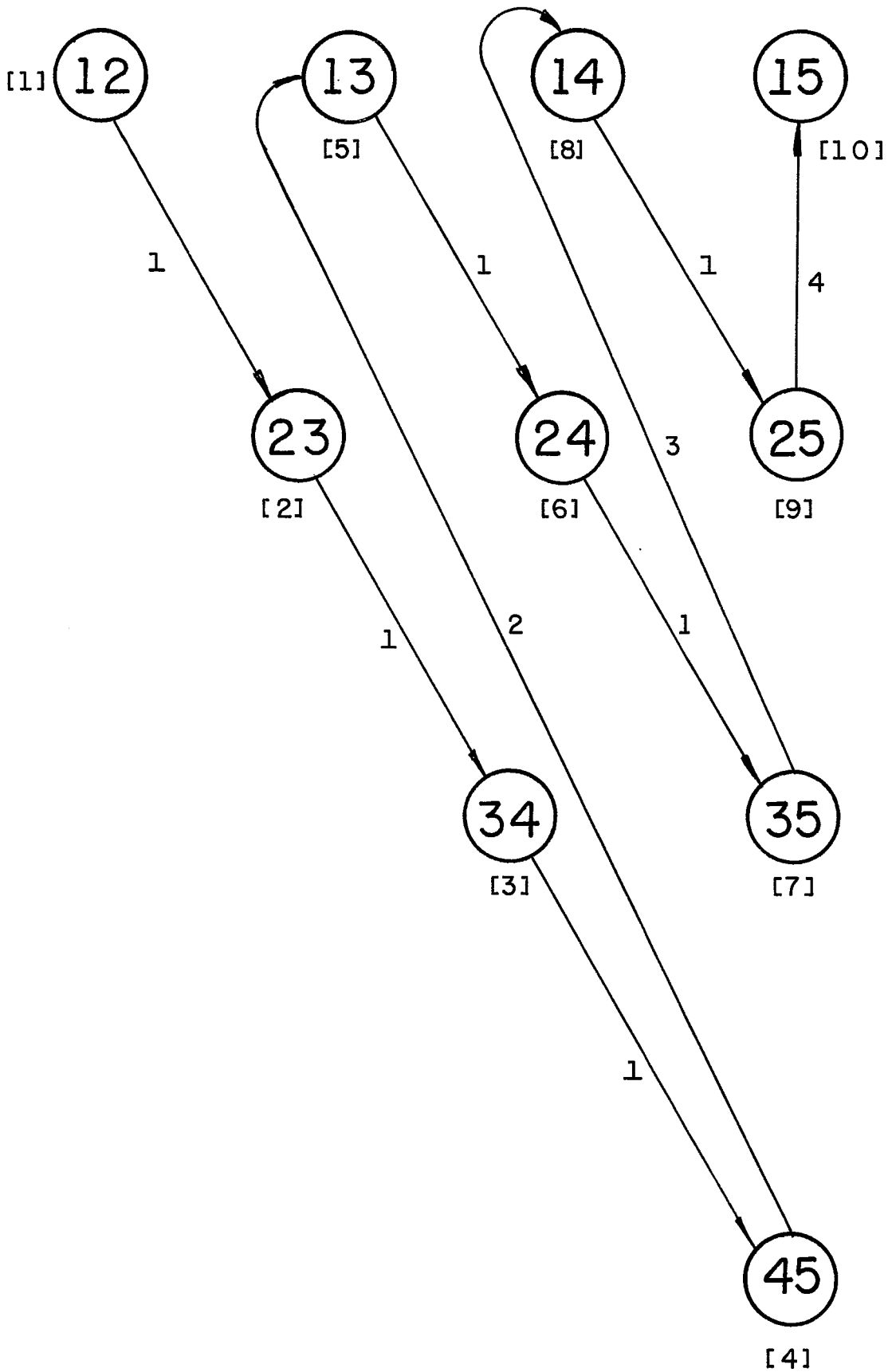


Figura 31

	1	2	3	4	5
1	2	-	-	-	12
2	3	-	-	5	-
3	4	-	4	-	-
4	5	3	-	-	-
5	-	1	1	1	-
12	23	-	-	-	-
23	34	-	-	-	-
34	45	-	-	-	-
45	-	13	-	-	-
13	24	-	-	-	-
24	35	-	-	-	-
35	-	-	14	-	-
14	25	-	-	-	-
25	-	-	-	15	-
15	-	-	-	-	-

Figura 32

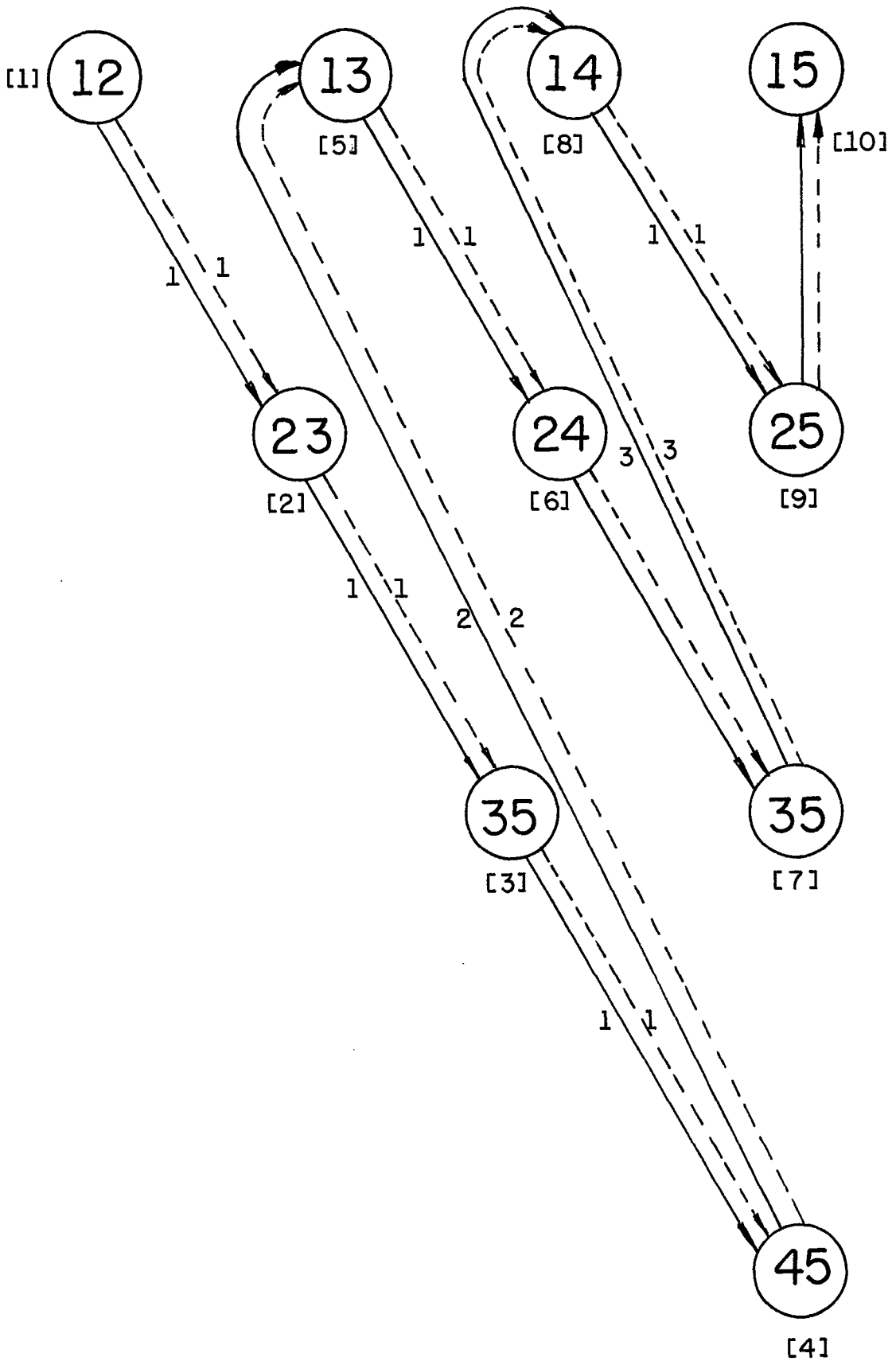


Figura 33

	1/11/2 1/31/4 1/5...	2/1 2/2 2/3 2/4 2/5...	3/1 3/2 3/3 3/4 3/5....	4/1 4/2 4/3 4/4 4/5....
1	2 - - - -	- - - - -	- - - - 1	- - - - 2
2	3 - - - -	- - - - -	- - - - -	- - - 5 -
3	4 - - - -	- - - - -	- - 4 - -	- - - - -
4	5 - - - -	- 3 - - -	- - - - -	- - - - -
5	- - - - -	- 1 - - -	- - 1 - -	- - - 1 -
12	23 - - - -	- - - - -	- - - - -	- - - - -
13	24 - - - -	- - - - -	- - - - -	- - - - -
14	25 - - - -	- - - - -	- - - - -	- - - - -
15	- - - - -	- - - - -	- - - - -	- - - - -
23	34 - - - -	- - - - -	- - - - -	- - - - -
24	35 - - - -	- - - - -	- - - - -	- - - - -
25	- - - - -	- - - - -	- - - - -	- - - 15 -
34	45 - - - -	- - - - -	- - - - -	- - - - -
35	- - - - -	- - - - -	- - 14 - -	- - - - -
45	- - - - -	- 1 3 - - -	- - - - -	- - - - -

Figura 34

	1/1	1/2	1/3	1/4	1/5	1/6	1/7	1/8	1/9	2/1	2/2	2/3	2/4	2/5	2/6	2/7	2/8	2/9
1	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-
2	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	-
3	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-
4	5	-	-	-	-	-	-	-	-	-	3	-	-	-	-	-	-	-
5	-	-	-	-	-	1	-	-	-	-	1	-	-	-	-	-	-	-
12	23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	34	-
24	35	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
35	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-	13	-	-	-	-	-	-	-

	3/1	3/2	3/3	3/4	3/5	3/6	3/7	3/8	3/9	4/1	4/2	4/3	4/4	4/5	4/6	4/7	4/8	4/9
1	-	-	-	-	1	-	-	-	-	-	-	-	-	2	-	-	-	-
2	-	-	-	-	-	-	-	-	3	-	-	-	5	-	-	-	-	-
3	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4
4	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-	-	-	5
5	-	-	1	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	15	-	-	-	-	-	-
34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45
35	-	-	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figura 35

VII .4. Observações

Note que com o algoritmo de construção da máquina usada na demonstração do item VII.3, a menos que haja pequenas modificações, podemos, dado um n qualquer, controlar a ordem e a memória da máquina a ser construída, ou seja :

1. Com o passo 3 controlamos a ordem finita da máquina.
2. O controle da memória pode ser feito aplicando o passo 4 para os pares não ordenados cujos números de ordem varia de 1 até $(J-1)$ e depois completamos a especificação da máquina construindo a tabela de teste da memória finita da máquina de maneira adequada para não formar ciclos no grafo de teste.

Para $n=2$ é fácil de verificar a relação $J + k - 1 \leq 2$, ou seja, para $n = 2$ temos as máquinas da figura 36.

A figura 37 mostra um esquema de um sistema análogo ao da figura 6, só que nesta temos uma máquina de memória finita J e ordem finita k . Neste sistema de comunicação no início, além de transmitirmos os símbolos codificados, transmitimos também o J primeiros símbolos da sequência de entrada. O sistema da figura 37 permanece com a chave ligada até que tenhamos armazenado os J primeiros símbolos da sequência de entrada. Quando isto acontecer, desligamos a chave 1, esperamos até que o símbolo de saída de ordem $(J+k)$ tenha sido armazenado, ligamos a chave 2 e iniciamos o processo de decodificação, Os J símbolos de entrada com os J símbolos de saída introduzidos no cir -

cuito lógico 1 da figura 37 nos dão o estado S_J , considerado aqui como o estado inicial. Este estado inicial, com os símbolos de saída de ordem $J+1, J+2, J+3, \dots, J+k$, introduzidos no circuito lógico 2, nos fornece o símbolo de entrada de ordem x_{j+1} que agora, através da chave 2, é armazenado para ser introduzido no circuito lógico 1, e assim por diante até que a sequência de entrada tenha sido decodificada. Note que neste esquema, começamos a decodificar a informação apenas do símbolo de entrada de ordem $J+1$ para frente, porém aqui a máquina decodificadora é mais simples do que a decodificadora da figura 6 que tem uma seção onde devemos ter as transições da máquina codificadora.

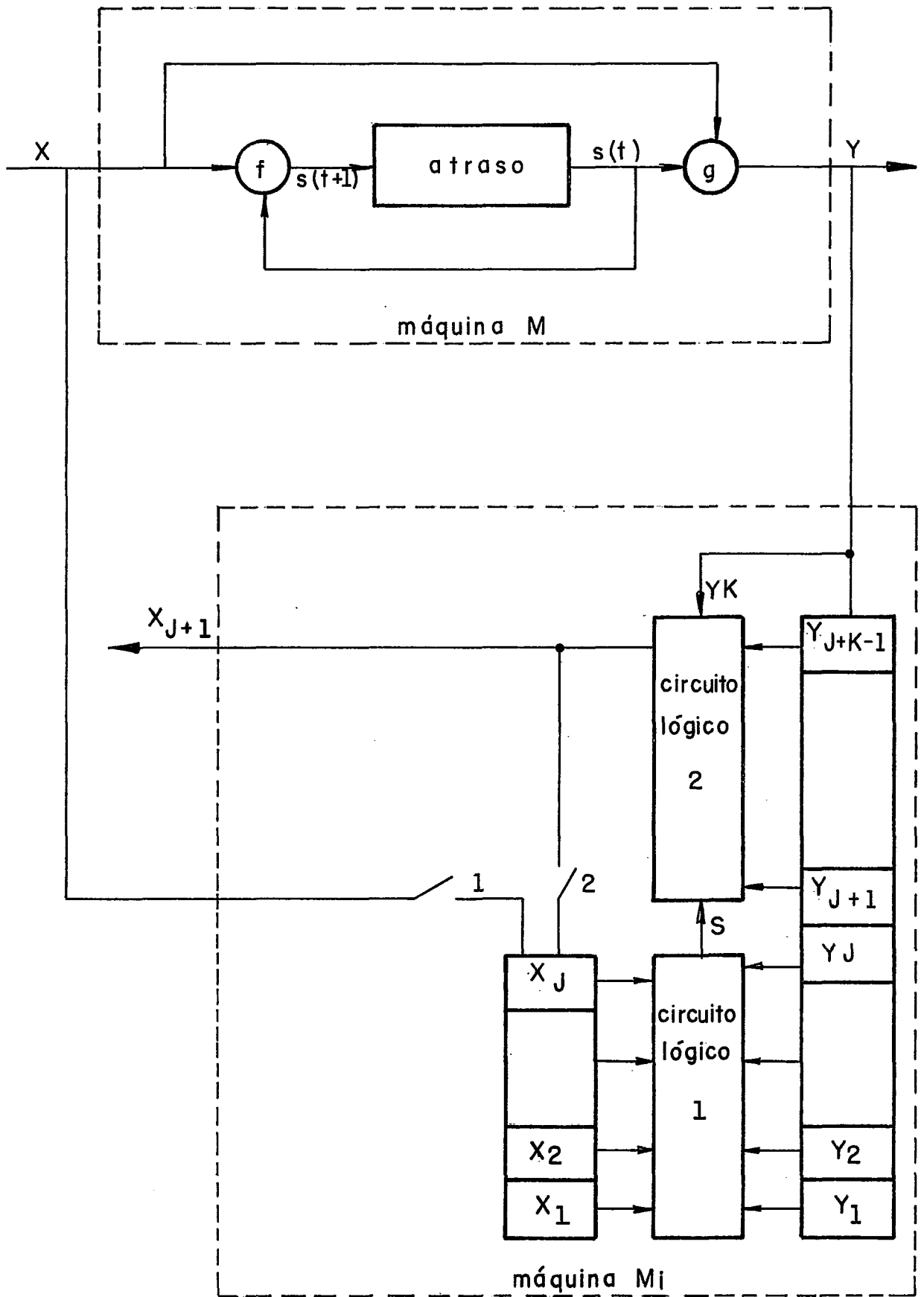


Figura 37

VIII. MÁQUINAS DEFINIDAS

Uma máquina reduzida é uma máquina definida de grau I se I é o menor inteiro para o qual o estado final fica determinado por uma sequência de entrada de comprimento I , isto é, se $x \in \Sigma^I = x_1 x_2 \dots x_I$ então $x \rightarrow S_f$. Em KOHAVI¹, páginas 450, 451 e 452, temos esta definição e dois algoritmos de identificação de máquinas reduzidas.

Para este tipo de máquina, considerando que:

- 1) se ela tem grau I , então $I \leq n-1$, onde n é o número de estados (ver KOHAVI¹, página 454);
- 2) é fácil de se ver que se ela é definida de grau I é também de memória finita $J \leq I$;

podemos facilmente concluir que:

- a) Se M tem ordem finita k , memória finita J , e é definida de grau I , então $I+J+k-1 \leq (n-1)(n+4)/2$ ou $I+k-1 \leq J+k-1 \leq (n-1)(n+2)/2$ e
- b) para um dado n qualquer podemos facilmente construir uma máquina com n estados que seja definida de grau $I = n-1$, de memória finita $J = n-1$ e de ordem finita $k = (n(n-1)/2)+1$.

Vejamos, para maior esclarecimentos, a construção de uma máquina com 4 estados. Facilmente obtemos uma máquina definida de grau $I = 3$, memória finita $J = 3$ e ordem finita $k = 7$, usando o mesmo algoritmo dado no item VI.3. Aqui a memória J fica limitada a $(n-1)$ por ser a máquina também definida. Com os passos 1 e 2 obtemos a figura 38. Completamos a espe

cificação da máquina usando as tabelas de teste para identificação de ordem finita e do grau de definição da máquina mostrada na figura 39. Note que aqui também podemos facilmente controlar o I, do mesmo modo do item VI.3, apenas tomando certos cuidado no preenchimento das tabelas.

Na figura 40 temos um sistema de codificação, transmissão e decodificação usando uma máquina definida de grau I e também de ordem finita k. Neste sistema transmitimos inicialmente os I primeiros símbolos de entrada, desprezando os I primeiros símbolos de saída, começando a armazenar do símbolo de saída de ordem I+1 para frente. Com os I primeiros símbolos de entrada, a máquina decodificadora toma conhecimento de qual é o estado S_I . Com este estado e os símbolos de saída de ordem I+1, I+2, ... I+k ela determina o símbolo de entrada de ordem I+1. Agora, com os símbolos de entrada 2,3,4...I+1, ela determina o estado S_{I+1} e assim por diante até que toda sequência tenha sido decodificada. Note que neste esquema a máquina decodificadora precisa armazenar apenas I+k símbolos, sendo I símbolos de entrada, e k símbolos de saída.

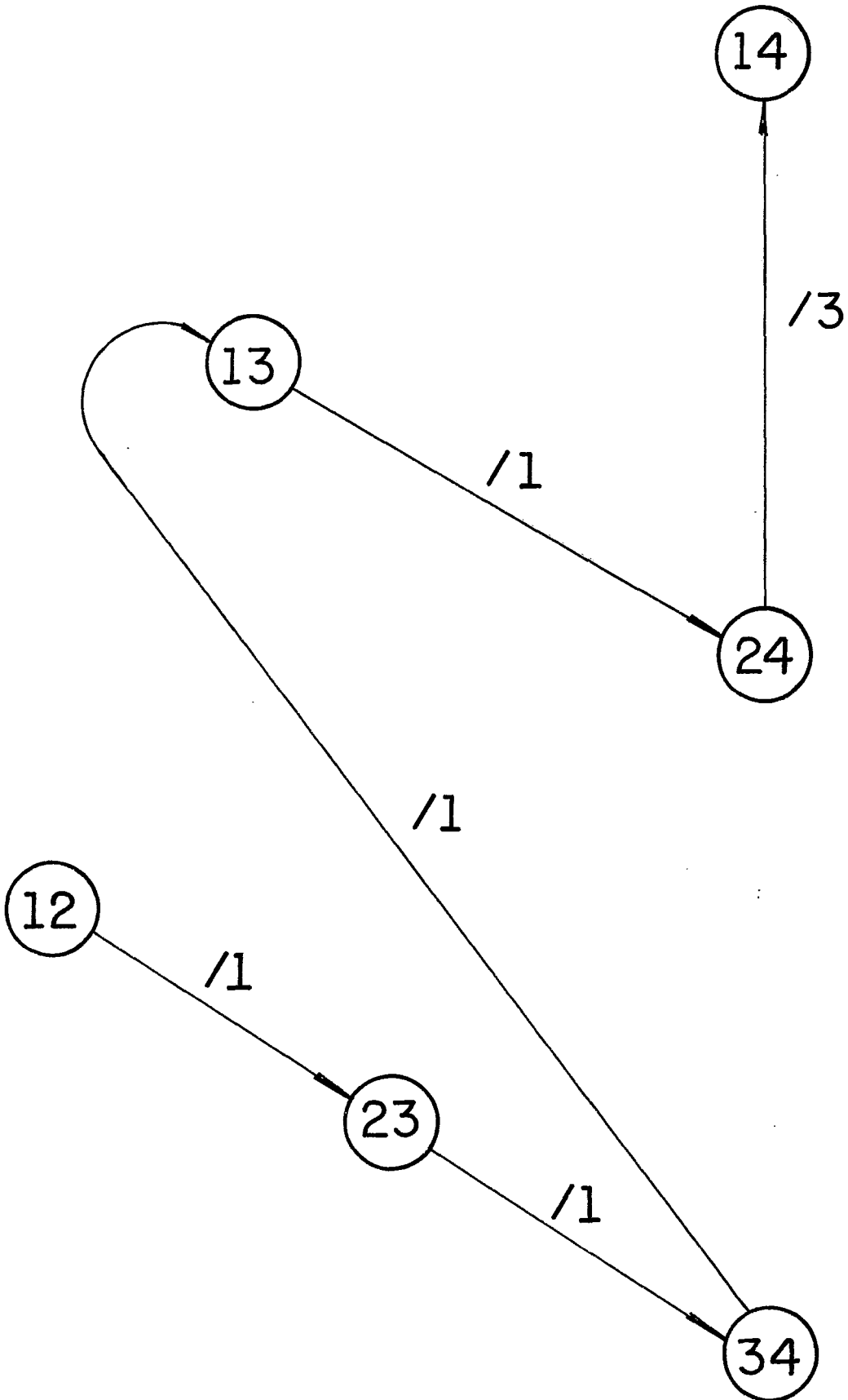


Figura 38

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	2	-	-	12	-	3	-	-	1	-	-	4	-	-	1	-	-	-	-	-	-	-	-	-
2	3	-	4	-	-	-	3	-	-	1	-	-	-	-	-	1	-	1	-	-	2	-	-	-
3	4	3	-	-	-	-	-	-	-	-	1	-	4	-	-	-	1	1	1	-	-	2	-	-
4	-	1	1	-	4	-	-	3	-	-	-	-	-	4	-	-	-	-	-	1	-	-	2	-
12	23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	-	13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

	1	2	3	4	5	6	7
1	2	3	1	4	1	1	2
2	3	3	1	4	1	1	2
3	4	3	1	4	1	1	2
4	4	3	1	4	1	1	2
12	23	33	11	44	11	11	22
13	24	33	11	44	11	11	22
14	24	33	11	44	11	11	22
23	34	33	11	44	11	11	22
24	34	33	11	44	11	11	22
34	44	33	11	44	11	11	22

Figura 39

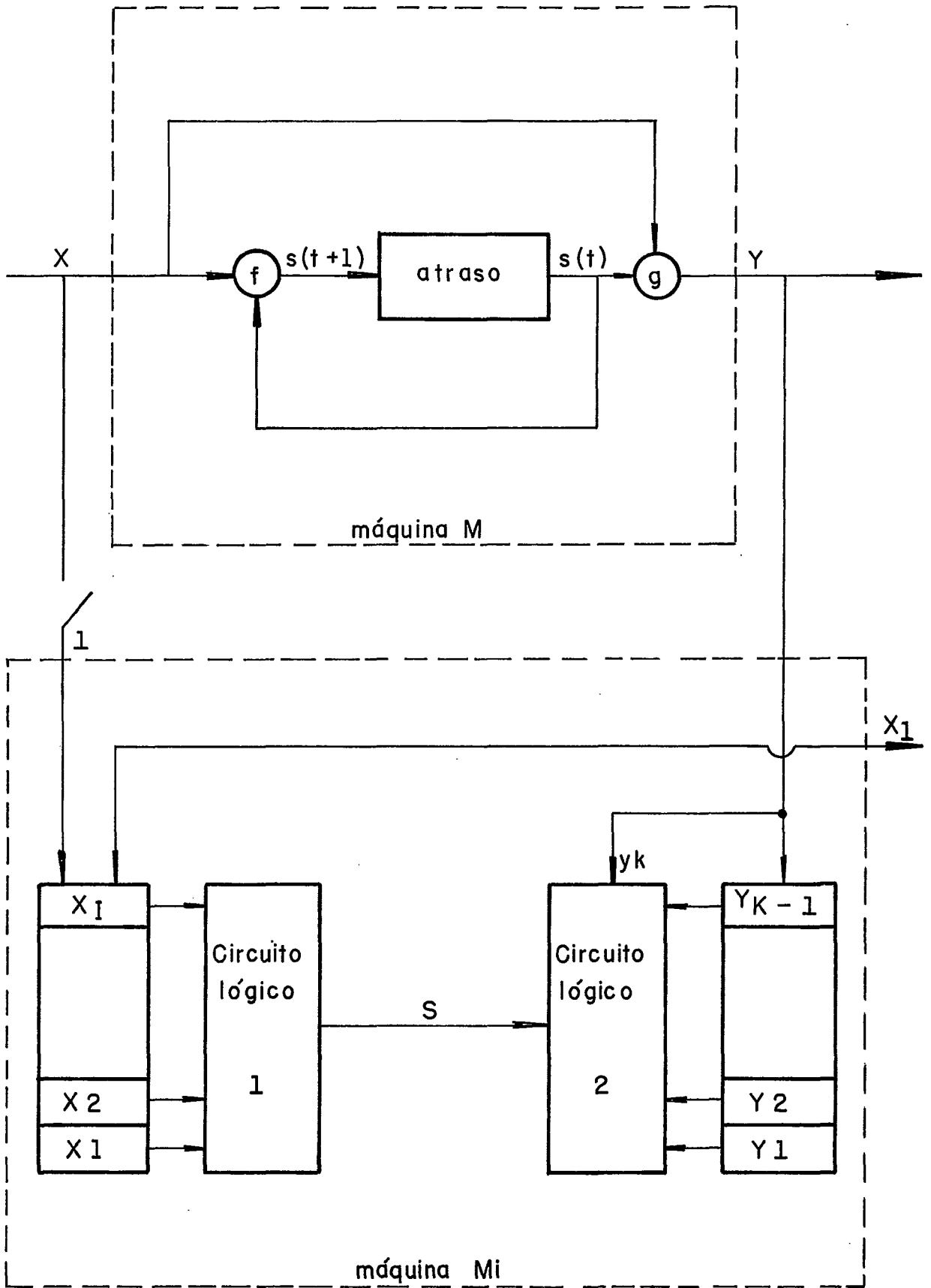


Figura 40

IX. CONCLUSÕES

Mostramos que se uma máquina tem imagem homomorfa que é sem perda de informação, então ela também é sem perda de informação. No item V.2. vimos que se uma máquina não é reduzida e é sem perda de informação, a máquina reduzida obtida a partir da primeira pelos processos de redução não precisa ser sem perda de informação. Então vemos que podemos a partir de uma máquina reduzida qualquer, pelo processo "state splitting" obter uma máquina não reduzida equivalente à primeira que seja sem perda de informação. Este processo não foi desenvolvido neste trabalho, por já existir um processo mais eficiente e fácil de se obter a partir de uma máquina qualquer, uma que seja sem perda de informação e que ainda continua sendo reduzida (ver item II).

No caso da conexão série de uma máquina M_1 com uma máquina M_2 sem perda de informação, temos como resultado uma máquina M que é sem perda de informação se e somente se M_1 também for. Quando M_2 for de ordem finita k_2 , M será de ordem finita K se e somente se M_1 for de ordem finita k_1 e então $k_1 \leq k \leq k_1 + k_2 - 1$.

Na conexão paralela da máquina M_1 com a máquina M_2 , com saída lógica L , uma condição suficiente para que M seja sem perda de informação é que L seja sem perda de informação com relação à saída de uma das máquinas que também deverá ser sem perda de informação. E se uma delas for de ordem finita k , M será de ordem finita menor ou igual a k se L for sem

perda de informação com relação à saída da máquina de ordem finita k . Se tivermos uma máquina M sem perda de informação, que possa ser decomposta na conexão paralela de M_1 com M_2 com saída lógica L ; podemos afirmar que M_1 , M_2 e L não precisam ser sem perda de informação, isto é, a condição de que uma delas seja sem perda de informação para que M seja também é suficiente mas não necessária.

Quanto à conexão cascata, também estabelecemos as condições suficientes mas não necessárias para que a máquina resultante seja sem perda de informação. Estas condições foram mais facilmente encontradas, uma vez impostas as condições para os casos especiais de conexão cascata, que são as conexões série e paralela. (Ver item VI.4.2)

Para conexão com realimentação é suficiente que L seja sem perda de informação, controlado por Δ_2 e M_1 sem perda de informação para que a máquina resultante seja sem perda de informação. Se M é sem perda de informação, então é necessário que L seja controlado por Δ_2 se a função h de máquina M_2 for sobrejetora, e M_1 sem perda de informação se $L_2: \Sigma \rightarrow \Sigma_1$ uma função sobrejetora paracada $z \in \Delta_2$.

No relacionamento das máquinas sem perda de informação com as máquinas de memória finita vimos que se FM representar o conjunto de máquinas de memória finita, $LM \cap FM$ não é vazia. Vimos também um algoritmo fácil para construção de qualquer máquina situada nesta interseção. Concluímos que em sistemas de comunicação torna-se mais interessante trabalhar com máquinas situadas nesta interseção uma vez que na construção da

máquina decodificadora, podemos substituir a parte que contém as transições da máquina codificadora por um simples circuito lógico e linhas de atraso. Nestes sistemas, a transmissão do estado inicial se resume na transmissão de alguns símbolos da sequência de entrada ou de saída, que poderia ser por exemplo uma chamada de posição para máquina decodificadora. Outra vantagem ainda é que estas máquinas se comportam bem diante de erros temporários. (Ver NEUMANN⁹).

Como uma continuação do estudo das propriedades das máquinas sem perda de informação parece interessante estudar qual seria o seu relacionamento com as máquinas lineares e qual seria o tipo de linguagem aceita por máquinas sem perda de informação.

X. REFERÊNCIAS BIBLIOGRÁFICAS

1. KOHAVI, Zvi - Switching and finite automata theory, TATA McGraw-Hill, Bombay, New Delhi, 1970.
2. BOOTH, Taylor L. - Sequential machines and automata theory, John Wiley & Sons, Inc., New York, 1967.
3. HUFFMAN, David A. - Canonical forms for information lossless machines, IRE Transactions on Circuit Theory, CT-6, Special Supplement, 41-59, maio de 1959.
4. EVEN, Shimon - On information-lossless automata of finite order, IEEE Transactions on Electronic Computers EC-14, 561-569, agosto de 1965.
5. VELOSO, Paulo Augusto Silva - On interconnections of definite and finite recall machines, Publicação Técnica PTC-5176, COPPE-UFRJ, março de 1976.
6. GILL, A. - Finite-state systems in L.A. Zadek e E.Polak(eds), System Theory - McGraw-Hill, New York, 43 - 94 1969.
7. NEWBORN, Monroe M. - Maximal memory binary output finite-memory sequential machines, IEEE Transactions on Computers, Janeiro de 1968.

8. LAVALEE, Pierre e Zvi Kohavi - Design of sequential machines with fault detection capabilities, IEEE Transactions on Computers, vol. EC-16, número 4, pp. 473, agosto de 1967.
9. NEUMANN, Peter G. - Error limiting coding using information lossless sequential machines, IEEE Transactions Information Theory, vol. IT-10, pp 108, número 2, abril de 1967.
10. SERANSKII, D.V. - The transformation of an automaton into an information-lossless automaton, Automation and Remote Control Automat i Telemekhan, pp. 1452, fevereiro de 1975.