

MASTER

Strategies for the discrete AND-channel

Ligtenberg, L.G.T.M.

Award date:
1983

[Link to publication](#)

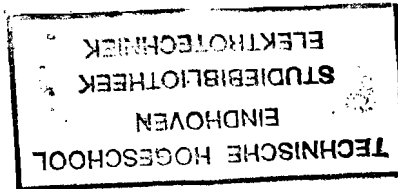
Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain



EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering

Group Information- and Communication Theory

STRATEGIES FOR THE DISCRETE
AND-CHANNEL

by

L.G.T.M. Ligtenberg

Coach Dr.K.A.Post

Prof.dr.ir.J.P.M.Schalkwijk

mei 1983

THE DISCRETE AND-CHANNEL

CONTENTS:

Chapter 1	Introduction	1
Chapter 2	A channel model	7
Chapter 3	History of the discrete case	14
Chapter 4	Better strategies for larger symmetric arrays	20
Chapter 5	Further results on our strategy	26
Chapter 6	Conclusions and recommendations	29
	Acknowledgement	31
	References	32
Appendix A:	The divisions of the Post-Ligtenberg strategy	1
Appendix B: (Dutch)	Het interactieve programma	1
	Een algemene bespreking van het programma	1
	Speciale opties	2
	De gedetailleerde beschrijving van het programma	4
	De beschrijving van de belangrijkste subroutines	5
	Aanbevelingen voor verbeteringen van het programma	7

1 INTRODUCTION

Shannon [1] derived inner and outer bounds to the capacity region of the TWO WAY CHANNEL (TWC). A TWC is a discrete memoryless channel with finite input and output alphabet, and defined by a matrix $\{P(Y_1, Y_2 | X_1, X_2)\}$ of transition probabilities.



figure 1.1 The Two Way Channel

Shannon's inner bound region equals the convex hull of the region of the rate pairs $(I(X_1; Y_2 | X_2), I(X_2; Y_1 | X_1))$ where the input probability distribution $P(X_1, X_2)$ is allowed to vary over all product distributions $P(X_1, X_2) = P(X_1)P(X_2)$. Likewise the Shannon outer bound region is the convex hull of the region of rate pairs $(I(X_1; Y_2 | X_2), I(X_2; Y_1 | X_1))$ where the input distribution $P(X_1, X_2)$ is no longer restricted to be of the product type.

Summarizing, let

$$G_1 = \{(I(X_1; Y_2 | X_2), I(X_2; Y_1 | X_1)) : P(X_1, X_2) \text{ product PD}\} \quad (1.1)$$

and

$$G_0 = \{(I(X_1; Y_2 | X_2), I(X_2; Y_1 | X_1)) : P(X_1, X_2) \text{ arbitrary PD}\} \quad (1.2)$$

then

$$\text{co } G_1 \subset \text{capacity region TWC} \subset \text{co } G_0 \quad (1.3)$$

Blackwell's Binary Multiplying Channel (BMC), defined by the expression $Y_1 = Y_2 = X_1 X_2$, see fig 1.2, is an example of a simple TWC for

which the inner and outer bound regions differ.

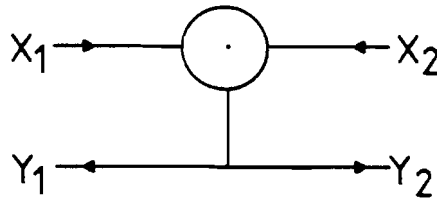


figure 1.2 The Binary Multiplying Channel

In fact, figure 13 of reference [2], reproduced here as figure 1.3, gives the inner and outer bound region of the BMC.

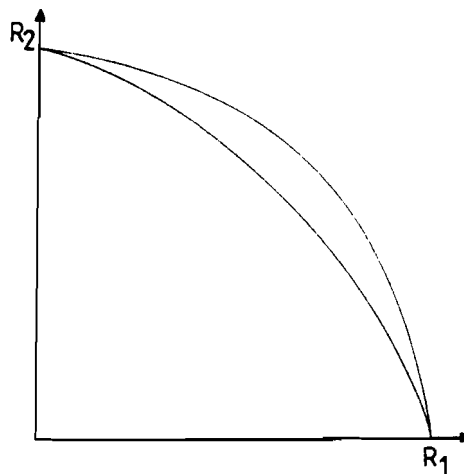


figure 1.3

Now, as a heuristic introduction, let us consider a specific situation that occurs during the transmission procedure for the BMC. Let user k has the set of M_k messages at his disposal, from which he decides to send message m_k to the other user ($k=1,2$). Transmission is finished as soon as both users know the message pair (m_1, m_2) in the Cartesian product $M_1 \times M_2$. Of course user k knows his own message m_k but a priori only knows that the message of user p , m_p , belongs to M_p ($p \neq k$). After every transmission, a subset $S(t)$ of $M_1 \times M_2$ should be determined in which the unknown message pair (m_1, m_2) has to be found. Our goal is reached after t_0 transmissions when $S(t_0)$ has a small size (in a sense

that shall be explained later). Besides, the transmission coding strategy should be applicable to all message pairs (m_1, m_2) and determined in such a way that we obtain an average mutual information which is as large as possible.

To be more specific, let $S(t)$ for some t be given. What can we say about $S(t+1)$? We are only concerned about the subset $S(t)$, because we know from the last t transmissions that (m_1, m_2) lies within this subset. The intermediate $(t+1)$ st transmission corresponds to a single application of the BMC. If and only if both users send the symbol one, they will receive $Y_1=Y_2=1$ (figure 1.2). So when user1 decides to send the symbol one for those messages α which are element of the subset V_1 (i.e. $\alpha \in V_1$ and $V_1 \subset M_1$) and user2 decides to send the symbol one for those messages β which are element of the subset V_2 (i.e. $\beta \in V_2$ and $V_2 \subset M_2$) we know after the $(t+1)$ st transmission that $(m_1, m_2) \in (V_1 \times V_2) \cap S(t)$ if one receives $Y_1=Y_2=1$ for the $(t+1)$ st transmission or that $(m_1, m_2) \in (S(t) \setminus (V_1 \times V_2))$ if one receives $Y_1=Y_2=0$ for the $(t+1)$ st transmission. Here $V_1 \times V_2$ is the cartesian product of V_1 and V_2 . See figure 1.4.

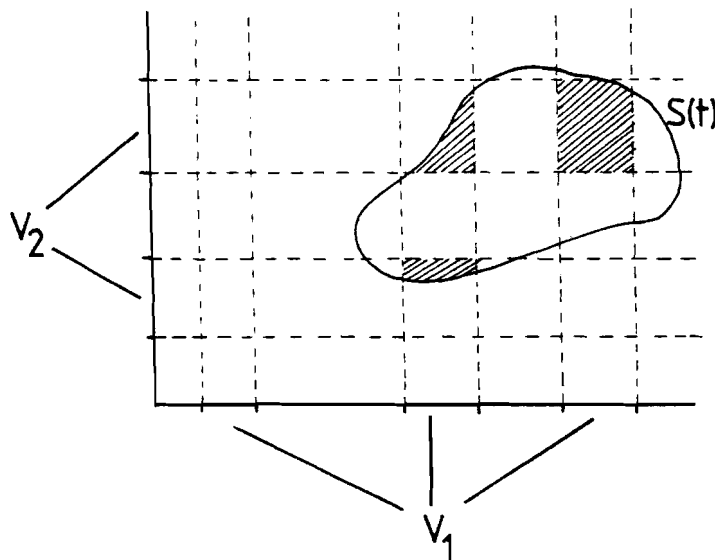


figure 1.4 The results of a transmission

Thus apparently $S(t+1)$ is one of the following subsets, depending on $Y_1=Y_2$.

$$S(t+1) = \begin{cases} S_1(t+1) = (V_1 \times V_2) \cap S(t) & \text{if } Y_1=Y_2=1 \\ S_2(t+1) = S(t) \setminus (V_1 \times V_2) & \text{if } Y_1=Y_2=0 \end{cases} \quad (1.4)$$

So we find a subset $S(t+1)$ of $S(t)$, and with this subset we continue our search to find (m_1, m_2) in the same way as before by determining $S(t+2)$ etc.

In [2], Schalkwijk described his strategy for a continuous uniform distribution of message pairs (m_1, m_2) in a unit square $[0,1] \times [0,1]$ using a Markov chain with three states depending on the parameters α and γ . These states are related to the geometric shape of the set $S(t)$, to which the message pair (m_1, m_2) belongs after t transmissions as follows:

Two parameters α and γ are chosen beforehand. ($\alpha > 0$, $\gamma > 0$ and $\alpha + \gamma > 1$).

1. $S(t)$ can be a rectangle. Then we choose $V_1 \times V_2$ to be a right lower similar subrectangle with similarity ratio α .

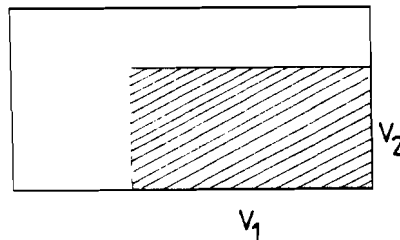


figure 1.5 The rectangle situation

So in this case $S_1(t+1)$ is again a rectangle with probability α^2 and an L-shaped set, $S_0(t+1)$ with probability $1-\alpha^2$.

2. $S(t)$ can be an L-shaped set as described above in sub 1. Then we choose $V_1 \times V_2$ to be a left upper rectangle, similar to the outer rectangle with similarity ratio γ .

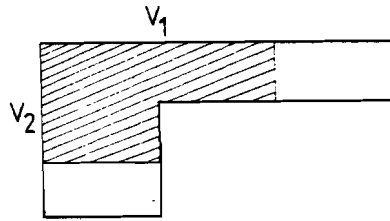


figure 1.6 The L-shaped situation

So in this case $S_0(t+1)$ is a union of two projectionally separated (and hence disconnected) rectangles with probability $2(1-\gamma)/(1+\alpha)$ and $S_1(t+1)$ is a clipped L-shaped set with probability $1 - \{2(1-\gamma)/(1+\alpha)\}$. It is important to notice that now three independent subsets of $S(t+1)$ remain, because the two rectangles of S_0 are disconnected i.e. user1 and user2 can independently detect in which rectangle the message pair (m_1, m_2) lies.

3. $S(t)$ can be a clipped L-shaped set as described in sub 2. Then we choose $V_1 \times V_2$ to be the smallest inscribed left upper subrectangle, that splits our set into 3 disjoint rectangles.

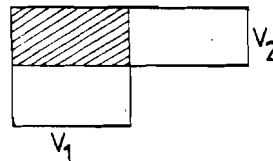
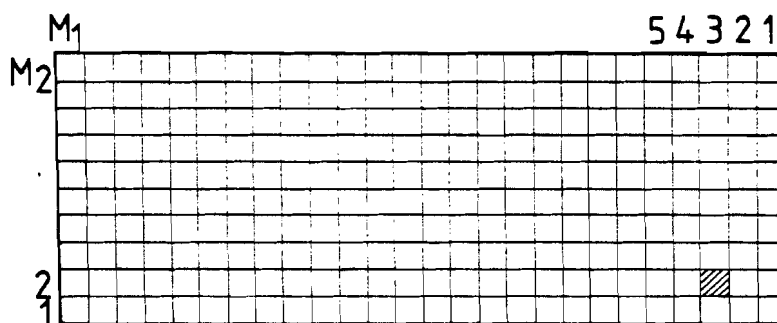


figure 1.7 The clipped L-shaped situation

In this case $S_1(t+1)$ is a rectangle and $S_0(t+1)$ is the union of two projectionally separated rectangles. So anyhow the state $S_k(t+1)$ is a rectangle state with probability 1 ($k=0,1$).

Now $S(0)$ is a rectangle state $M_1 \times M_2$. Hence a Markov chain is born! Optimizing this strategy for α and γ , Schalkwijk found $\alpha(=0.67571)$ and $\gamma(=0.52545)$, and hence he found a rate pair $(R_1, R_2)=(0.61914, 0.61914)$ for this strategy, which is beyond Shannon's inner bound region. (figure 1.3)

In this paper we will discuss strategies for the BMC with a finite number of equiprobable messages (M_1) for user1 and also (M_2) for user2 (The discrete case). It is easily show that an isomorphic mapping of Schalkwijk's strategy to our situation is impossible, because we have to deal with more constraints than Schalkwijk. This is due to the fact, that the lines that split the rectangle into subsets, now cannot be choosen as to isolate a fraction α or $(1-\gamma)$ of the messages in either direction, but these lines now are restricted to isolated rational fractions j/M_k ($j \in \{0,1,2,\dots,M_k\}$) of messages.



message pair (3,2)

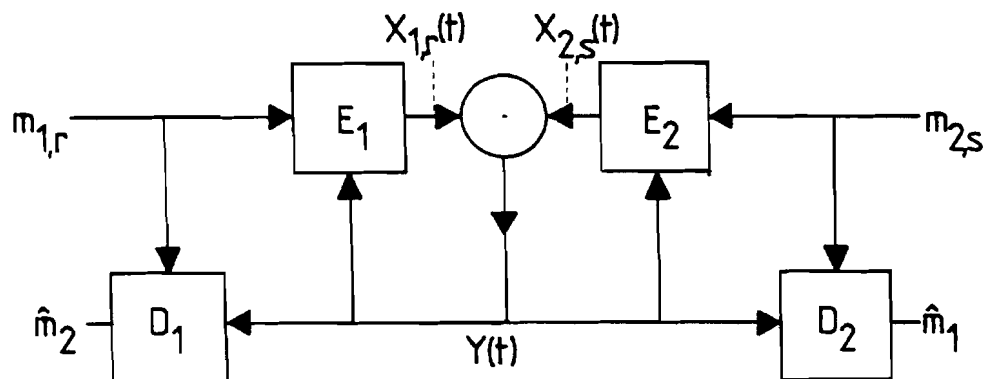
figure 1.8 The discrete case

In this paper we will discuss several attempts which are made to solve the discrete case i.e. to single out every message pair (m_1, m_2) of the $M_1 \times M_2$ array.

2 A CHANNEL MODEL

Until now we have talked about M_1 and M_2 meaning a set of messages of user1 resp. user2. From this point, it is possible that we mean the set M_1 resp. M_2 , or the number of messages of user1 resp. user2. This will cause no confusion.

Now let us consider the discrete case, and try to form a model for this situation, formulate some definitions and derive some expressions which we will need further on in this paper. Figure 2.1 gives a model of the discrete binary multiplying channel for the discrete case, under the assumption that user1 has M_1 messages and user2 has M_2 messages at his disposal.



E_k : encoder of side k

D_k : decoder of side k

figure 2.1 A channel model

$m_{1,r}$ and $m_{2,s}$ are assumed to be messages of user1 resp. user2. Thus $r \in \{1,2,\dots,M_1\}$ and $s \in \{1,2,\dots,M_2\}$. All of these messages have to be encoded by binary symbols 0,1 before they can be sent over the channel at discrete points of time t ($t=1,2,\dots$). Assume that we can encode all of these messages $m_{1,r}$ and $m_{2,s}$ in binary sequences of length n , then we can represent every message by a binary vector $\underline{X}_{1,r}$ resp. $\underline{X}_{2,s}$ i.e.

$$\underline{X}_{1,r} = (X_{1,r}(1), X_{1,r}(2), \dots, X_{1,r}(n)) \quad (2.1)$$

and

$$\underline{X}_{2,s} = (X_{2,s}(1), X_{2,s}(2), \dots, X_{2,s}(n)) \quad (2.2)$$

With these time dependent components of the vector $\underline{X}_{1,r}$ and $\underline{X}_{2,s}$ we can describe the channel output at time t ($Y(t)$), thus

$$Y(t) = X_{1,r}(t) \cdot X_{2,s}(t) \quad (2.3)$$

and also can define an output vector, starting at $t=1$, and of length, say t_0 , which gives for a certain couple of input vectors $\underline{X}_{1,r}$ and $\underline{X}_{2,s}$ the output sequence of the BMC i.e.

$$\underline{Y}_{t_0} = (Y(1), Y(2), \dots, Y(t_0)) \quad (2.4)$$

were $Y(t) \in \{0,1\}$. So for every couple $\underline{X}_{1,r}$ and $\underline{X}_{2,s}$, there exist an output vector \underline{Y}_n of length n . Thus,

$$\underline{Y}_n = \underline{X}_{1,r} \cdot \underline{X}_{2,s} \quad (\text{componentwise product}) \quad (2.5)$$

Now let us try to describe a certain situation $S(t)$. Such a situation depends on the last t transmissions done by user1 and user2, for the different messages $m_{1,r}$ and $m_{2,s}$ resp. for all possible r and s , and the choice, which subset one chooses after every transmission to investigate further on. The latter is not so easy to describe as one would expect. For instance looking backwards to the Schalkwijk strategy, there can exist situations, that the subset $S_0(t)$ is a union of several projectionally separated rectangles.

All of these rectangles can be investigated independently from each other i.e. it is possible that there exist several message pairs (m_1, m_2) which can be described with the same vector \underline{Y}_n . So the problem is: How can we define the different subsets of $S(t)$ after the $(t+1)$ st transmission? Therefore let us consider a specific situation!

Let $S(t)$ be a subset of $S(0)$ after t transmissions, which one has to investigate in the $(t+1)$ st transmission. As described in the introduction, user1 can decide to send the symbol one for those messages $m_{1,\alpha}$ which are element of V_1 (i.e. $m_{1,\alpha} \in V_1$ and $V_1 \subset M_1$) and user2 decides to send the symbol one for those messages $m_{1,\beta}$ which are element of the subset V_2 (i.e. $m_{2,\beta} \in V_2$ and $V_2 \subset M_2$). Thus we can define two subsets after the $(t+1)$ st transmission

$$S_1(t+1) = (V_1 \times V_2) \cap S(t) \qquad Y(t+1)=1 \qquad (2.6)$$

and

$$S_0(t+1) = S(t) \setminus (V_1 \times V_2) \qquad Y(t+1)=0 \qquad (2.7)$$

If we want to find $S(t+1)$ (i.e. the subset in which the message pair lies, which we are looking for) one has not only the choice between $S_0(t+1)$ and $S_1(t+1)$ (dependent on $\underline{X}_{1,r}(t+1)$, $\underline{X}_{2,s}(t+1)$ and \underline{Y}_{t+1}), but one has also to investigate if $S_k(t+1)$ ($k=0,1$) consists of several projectionally separated parts. If so, $S(t+1)$ is that part of $S_k(t+1)$ in which our message pair lies. So let us examine $S_k(t+1)$.

Between the different message pairs $(m_{1,r}, m_{2,s})$ from $S_k(t+1)$, we can define an equivalence relation as follows:

Definition: Two message pairs (α_1, α_2) and (β_1, β_2) , both element of a set S , are in the relation \sim , if there exist a number q and a sequence $\{(p_{10}, p_{20}), \dots, (p_{1q}, p_{2q})\} \subset S$ for which

- (i) $\alpha_1 = p_{10} \qquad \wedge \alpha_2 = p_{20}$
- (ii) $\beta_1 = p_{1q} \qquad \wedge \beta_2 = p_{2q}$
- (iii) $p_{1j} = p_{1(j+1)} \qquad \forall p_{2j} = p_{2(j+1)} \qquad \text{for all } j \in \{0, 1, \dots, q-1\}$

Lemma: This relation is an equivalence relation.

Proof:

1 First of all, this relation is reflexive. Take $q=0$, then

$$(\alpha_1, \alpha_2) = (p_{1q}, p_{1q}).$$

2 Secondly this relation is symmetric. Assume that $(\alpha_1, \alpha_2) \sim (\beta_1, \beta_2)$. Then there exist a sequence

$$\{(p_{10}, p_{20}), (p_{11}, p_{21}), \dots, (p_{1q}, p_{2q})\} \subset S$$

satisfying (i), (ii) and (iii). Now obviously the reverse sequence

$$(p_{1k}^*, p_{2k}^*) := (p_{1, q-k}, p_{2, q-k}) \quad \text{for } (0 \leq k \leq q)$$

yields $(\beta_1, \beta_2) \sim (\alpha_1, \alpha_2)$.

3 Finally, this relation is transitive. Let us assume that $(\alpha_1, \alpha_2) \sim (\beta_1, \beta_2)$ due to the sequence

$$\{(p_{10}, p_{20}), (p_{11}, p_{21}), \dots, (p_{1q}, p_{2q})\} \subset S$$

and that $(\beta_1, \beta_2) \sim (\gamma_1, \gamma_2)$ due to the sequence

$$\{(p_{1q}, p_{2q}), \dots, (p_{1m}, p_{2m})\} \subset S$$

then $(\alpha_1, \alpha_2) \sim (\gamma_1, \gamma_2)$ due to the sequence

$$\{(p_{10}, p_{20}), (p_{11}, p_{21}), \dots, (p_{1m}, p_{2m})\} \subset S$$

Hence \sim is an equivalence relation.

The induced equivalence classes are exactly the minimal projectionally disjoint subsets of S . We call these classes "ambiguity classes" because, they have all the same vector $\underline{Y}(t+1)$.

So now we can say that after $(t+1)$ transmissions we choose for $S(t+1)$ that ambiguity class of $S_k(t+1)$ ($k=0,1$), depending on $\underline{X}_{1,r}(t+1)$, $\underline{X}_{2,s}(t+1)$ and $\underline{Y}(t+1)$, to which $(m_{1,r}, m_{2,s})$ belongs.

The last question we have to answer is: after how many transmissions do we have a reliable estimate of m_2 for user1 and m_1 for user2? I.e. what is the smallest n , for which

$$P((m_{1,r}, m_{2,s}) = (\hat{m}_1, \hat{m}_2) \mid \underline{Y}_n) = 1 \quad (2.7)$$

The solution is easy! When $S(t_0)$ after t_0 transmissions is a class, which consists of only one message pair (singleton), then $\hat{m}_1 = m_{1,r}$ and $\hat{m}_2 = m_{2,s}$. So we can characterize every message pair by a certain vector \underline{Y}_{t_0} ($t_0 < n$). It is important to notice that, given a certain strategy δ , the vectors \underline{Y}_{t_0} may have different lengths, and there may exist several vectors, for different message pairs, which are identical. In other words, we can define $\underline{Y}_{t_0, r, s, \delta}$

$$\begin{aligned} \underline{Y}_{t_0, r, s, \delta} &:= \text{the output vector } \underline{Y}_t \text{ with minimal number of} \\ &\quad \text{digits } (t_0) \text{ to single out a message pair} \\ &\quad (m_{1,r}, m_{2,s}) \text{ in the matrix } M_1 \times M_2 \text{ dependent} \\ &\quad \text{on the strategy } \delta. \end{aligned} \quad (2.9)$$

and also $\left| \underline{Y}_{t_0, r, s, \delta} \right|$

$$\begin{aligned} \left| \underline{Y}_{t_0, r, s, \delta} \right| &:= \text{the total length of the vector } \underline{Y}_{t_0, r, s, \delta} \\ &\quad \text{i.e. the number of transmissions which} \\ &\quad \text{have to be done to single out the message} \\ &\quad \text{pair } (m_{1,r}, m_{2,s}) \text{ for a certain strategy } \delta. \end{aligned} \quad (2.10)$$

So

$$\left| \frac{Y}{t_0}, r, s, \delta \right| = t_0 \quad (2.11)$$

Now let us derive an expression for the transmission rates R_{12} (from user1 to user2) and R_{21} (from user2 to user1). Therefore we count the total number of transmissions, necessary to transmit every possible message pair $(m_{1,r}, m_{2,s})$ given the set $M_1 \times M_2$. We define this as the number $\#(M_1, M_2, \delta)$.

$$\#(M_1, M_2, \delta) := \sum_{1 < r < M_1} \sum_{1 < s < M_2} \left| \frac{Y}{t_0}, r, s, \delta \right| \quad (2.12)$$

The average number of transmissions per message pair is

$$M(m_1, m_2, \delta) = \sum_{1 < r < M_1} \sum_{1 < s < M_2} P(r, s) \cdot \left| \frac{Y}{t_0}, r, s, \delta \right| \quad (2.13)$$

where $P(r, s)$ is the probability that a message pair $(m_{1,r}, m_{2,s})$ is sent over the channel. Given the situation that all messages in $M_1 \times M_2$ are equiprobable and thus all $P(r, s)$ are equal, then

$$M(m_1, m_2, \delta) = \frac{\#(M_1, M_2, \delta)}{M_1 M_2} \quad (2.14)$$

Now, we can define $R_{12, \delta}$ (R_{12} under the strategy δ) and $R_{21, \delta}$ in a way analogous to the practice in one-way-channel coding.

$$R_{12, \delta} = \frac{\log_2(M_1)}{M(m_1, m_2, \delta)} = \frac{\log_2(M_1)}{\#(M_1, M_2, \delta)} M_1 M_2 \quad (2.15)$$

and

$$R_{21, \delta} = \frac{\log_2(M_2)}{M(m_1, m_2, \delta)} = \frac{\log_2(M_2)}{\#(M_1, M_2, \delta)} M_1 M_2 \quad (2.16)$$

When we consider the symmetric case of the discrete problem (i.e. $M:=M_1:=M_2$) we can simplify these expressions as follows:

$$\#(M, M, \delta) := \#(M_1, M_2, \delta) \Big|_{M=M_1=M_2} \quad (2.17)$$

$$R_{\delta} := R_{12, \delta} := R_{21, \delta} := \frac{\log_2(M)}{\#(M, M, \delta)} M^2 \quad (2.18)$$

In this paper we mostly will deal with the symmetric case, and our aim is to maximize R_{δ} , hence to minimize $\#(M, M, \delta)$ over all possible strategies δ . Therefore we define the minimal number of transmissions for a symmetric $M \times M$ array as:

$$\text{MIN}(M) := \min_{\text{all } \delta} \#(M, M, \delta) \quad (2.19)$$

and the maximal rate for a symmetric $M \times M$ array as:

$$\text{MAX}(R) := \max_{\text{all } \delta} R_{\delta} = \frac{\log_2(M)}{\text{MIN}(M)} M^2 \quad (2.20)$$

3 HISTORY OF THE DISCRETE CASE

The first efforts to compute the rate of the BMC in the discrete case were made by Ligtenberg [3]. This was done by imitating the Schalkwijk strategy in the discrete case. In the first chapter of this paper, we have already mentioned, that such a translation has to deal with more constraints in the discrete case. Schalkwijk [2] described his strategy for a continuous uniform distribution of message pairs (m_1, m_2) in a unit square $[0,1] \times [0,1]$ using a Markov chain with three states depending on the optimal parameters α and γ . In our case however, the separated lines that split the rectangle into subsets must be chosen as rational numbers which most of the time do not equal (αM) or (γM) , where M is the number of messages.

Therefore the translation was made as follows:

From figure 7 of reference [2] we see that one can split the unit rectangle of Schalkwijk in 6 subrectangles. The cost of transmissions of every message point in such a subrectangle is 1,2 or 3 transmissions (see figure 3.1)

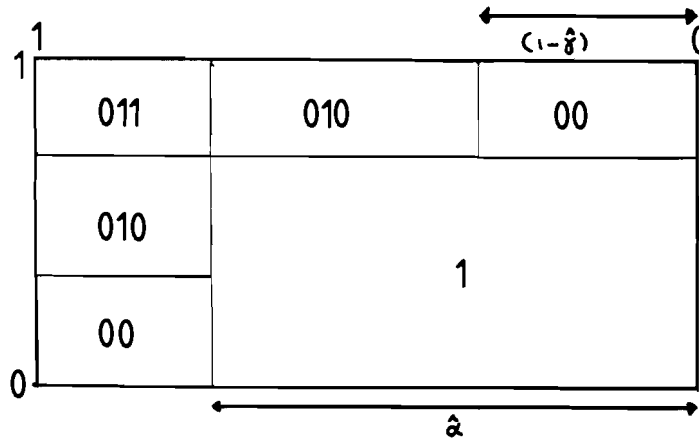


figure 3.1 The Schalkwijk strategy

So the aim of the translation was to split the array of $M_1 \times M_2$ message pairs into 6 subrectangles as described above. Therefore he defined the following parameters:

$$\begin{aligned} T_{11} &= \hat{\alpha}M_1 & T_{21} &= \hat{\alpha}M_2 \\ T_{12} &= (1-\hat{\gamma})M_1 & T_{22} &= (1-\hat{\gamma})M_2 \end{aligned} \quad (3.1)$$

Because T_{11}, T_{12}, T_{21} and T_{22} are almost never rational numbers, we searched for the dissections of the $M_1 \times M_2$ array that are closest to the dissection made by T_{11}, T_{12}, T_{21} and T_{22} . This was done by searching those message pairs (I, J) and (K, L) which had minimal distance to (T_{11}, T_{12}) respectively (T_{21}, T_{22}) . With these messages pairs (I, J) and (K, L) , one can make a division as shown in figure 3.2.

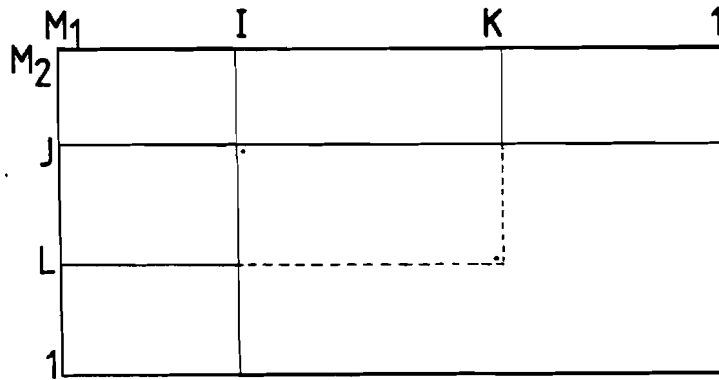


figure 3.2 An discrete division

And it is easily show, that we can compute $\#(M_1, M_2)$ with the next formula.

$$\begin{aligned} \#(M_1, M_2) &= 3M_1M_2 - 2IJ - (M_2 - J)(K - 1) - (M_1 - I)(L - 1) + \\ &\quad \#(I, J) + \#(K - 1, M_2 - J) + \#(L - 1, M_1 - I) + \\ &\quad \#(M_1 - I, M_2 - J) + \#(I - K + 1, M_2 - J) + \#(J - L + 1, M_1 - 1) \end{aligned} \quad (3.2)$$

Because these results may not be optimal (the choise of (I, J) and (K, L) may not be the best) Ligtenberg also computed $\#(M_1, M_2)$ for other possible choices of (I, J) and (K, L) . Namely for all those message pairs in the direct environment of (I, J) respectively (K, L) . (i.e.

$(I-1, J-1), (I-1, J), (I-1, J+1), (I, J-1), (I, J+1), (I+1, J-1), (I+1, J), (I+1, J+1)$ etc) So, for one computation of $\#(M_1, M_2)$ he investigated 9×9 possibilities altogether. This strategy is called the Para-Schalkwijk strategy. The results of these strategies however, were rather disappointing. The largest rate he found for $M_1 = M_2 = 4000$ was $R = .6099$, and for $M_1 = M_2 = 8000$ he found $R = .6103$, which is far below Shannon's inner bound ($R = .61695$). The results are shown in the figure 3.3.

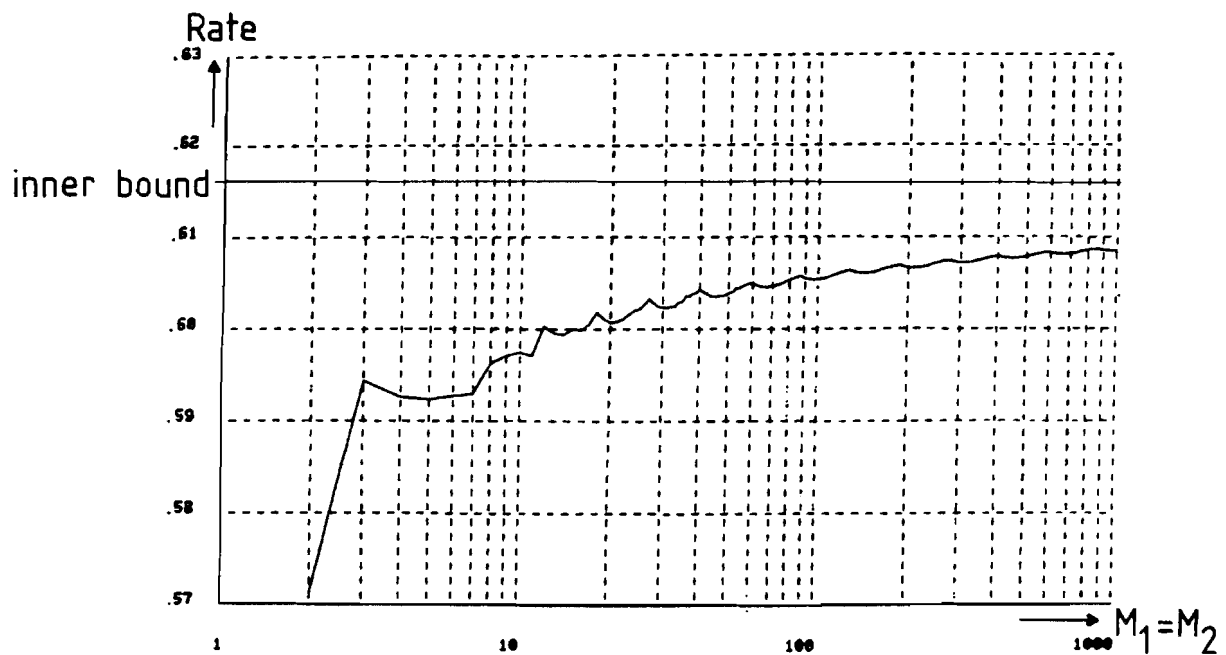


figure 3.3 Results of the Para-Schalkwijk strategy

The next efforts were made by Hollmann [4]. He derived a lower bound for $\#(M_1, M_2)$ for the $M_1 \times M_2$ array, where M_2 is small and M_1 arbitrary. For the proof he used the following lemma.

Lemma : Let Γ and Δ have disjoint projections Ψ and Ω respectively on the y - axis, and let Σ be an interval on the x - axis. Suppose

$$\Gamma \supset \Sigma \times \Psi \quad \text{and} \quad \Delta \supset \Sigma \times \Omega \quad (\text{cartesian product})$$

then

$$\text{MIN}(\Gamma \cup \Delta) > \text{MIN}(\Gamma) + \text{MIN}(\Delta) + |\Sigma| \tag{3.3}$$

The idea of the lemma is shown in figure 3.4

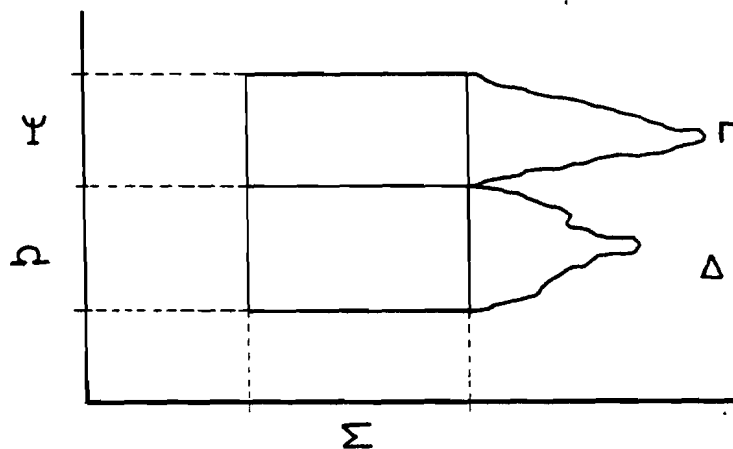


figure 3.4 The Hollmann lemma

Here $\text{MIN}(\Gamma)$ is defined as the minimal number of transmissions, needed to single out every message pair of Γ , etc.

With this lemma he derived that the minimal number of transmissions for an $M_1 \times M_2$ array can be estimated from below by

$$\#(M_1, M_2) > \min_{\substack{k, p \\ 1 < k < M_1 \\ 1 < p < M_2}} \{M_1 M_2 + \#(M_1, p) + \#(k, M_2 - p) + \#(M_1 - k, M_2 - p) + k\} \tag{3.4}$$

Here k and p are integers, which describe the L-shaped, which remains after the first transmission in the $M_1 \times M_2$ array (figure 3.5).

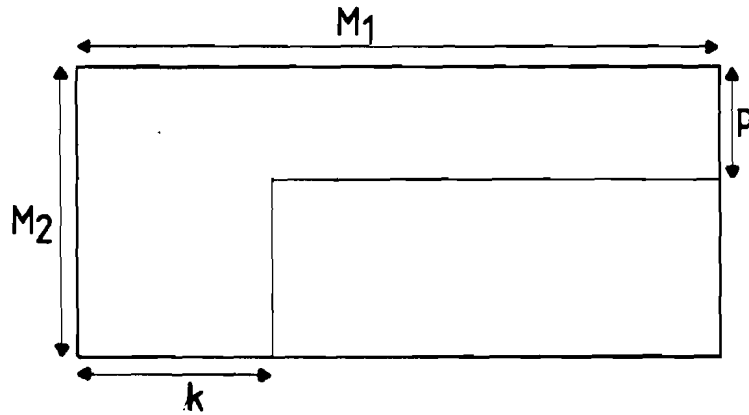


figure 3.5

Hollmann found for his lower bound the strategies for $M_2=1,2,3$ except for the 3×4 array. He computed that $\#(3,4) > 36$, but showed by an exhaustive search that there is no strategy which realizes 36. The optimal strategy he found, gives a number of transmissions which equals 37. For $M_1 \times 4$ arrays, Hollmann's lemma is too weak to prove optimality for "good strategies", that were found. The strategies Hollmann found for $M_2=3$ were non-Schalkwijk.

The next table on page 19 shows the differences between the Para-Schalkwijk strategy and the Hollmann strategy.

M_2	$M_1=3$	$\#(M_1, M_2)$	
		Para-Schalkwijk	Hollmann
1		5	5
2		14	14
3		24	24
4		37	37
5		51	50
6		66	65
7		82	80
8		98	96
9		115	113
10		132	130

tabel 3.1

4 BETTER STRATEGIES FOR LARGER SYMMETRIC ARRAYS

In this chapter, we will discuss strategies, which we used to solve the arrays 5×5 , 6×6 , 7×7 , 8×8 , 9×9 and 10×10 . The philosophy behind this strategy is, that one uses the information of the solution of smaller arrays, to compute $\#(M_1, M_2)$, for the array $M_1 \times M_2$, which we were investigating. We shall show that for small arrays, this strategy will produce the same $\#(M_1, M_2)$ as Ligtenberg [3] found with his computer-search for the Para-Schalkwijk strategy, but that for greater arrays ($M_1 = M_2 = 7, 8$ etc) $\#(M_1, M_2)$ increases slower, when we compare these results with the Para-Schalkwijk results

Let us first discuss the solution of the 3×3 array. Ligtenberg [3] and Hollmann [4], both computed $\#(3, 3) = 24$ as the number of transmissions to solve the 3×3 array. So it is possible to give a Para-Schalkwijk division of the 3×3 array. Figure 4.1 on page 21 shows this division.

Looking at this figure, one recognizes the three states (rectangle, L-shape and clipped L-shape) which we described in chapter 1 of this paper. So indeed, this is the Para-Schalkwijk solution which we were searching.

However we can explain this figure in another way. This explanation is in conformity with our philosophy. So, let us now discuss our philosophy using this explanation and figure 4.1.

In the first transmission, we split $S(0)$ in $S_0(1)$ and $S_1(1)$. We assume that we know the solution of $S_1(1)$, because this is the 2×2 array. However, we have already mentioned, that for our strategy, we use the information of the smaller arrays, which we have already solved. Therefore the division of the 2×2 array is also drawn in figure 4.1 by the sub-figures $S_1(1)$ and $S_{10}(2)$. Now we can see how we use the knowledge of the 2×2 array to

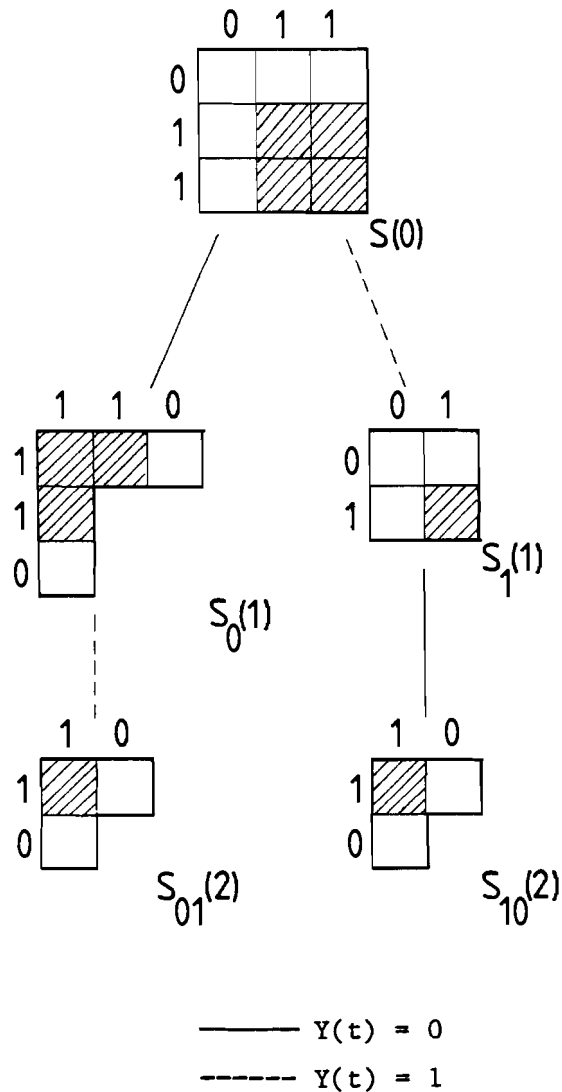


figure 4.1 The Para-Schalkwijk division

solve $S_0(1)$, because the sizes of the part we want to exclude from $S_0(1)$ are exactly the same as $S_{10}(2)$. I.e. $S_{01}(2)$ equals $S_{10}(2)$. This is the main idea, which we used, when we searched for better strategies.

Let us discuss an other example, for instance the 5x5 array. Ligtenberg [3] computed $\#(5,5)=58$ using a Para-Schalkwijk strategy. The solution of the Para-Schalkwijk strategy is given in figure 4.2 on page 22.

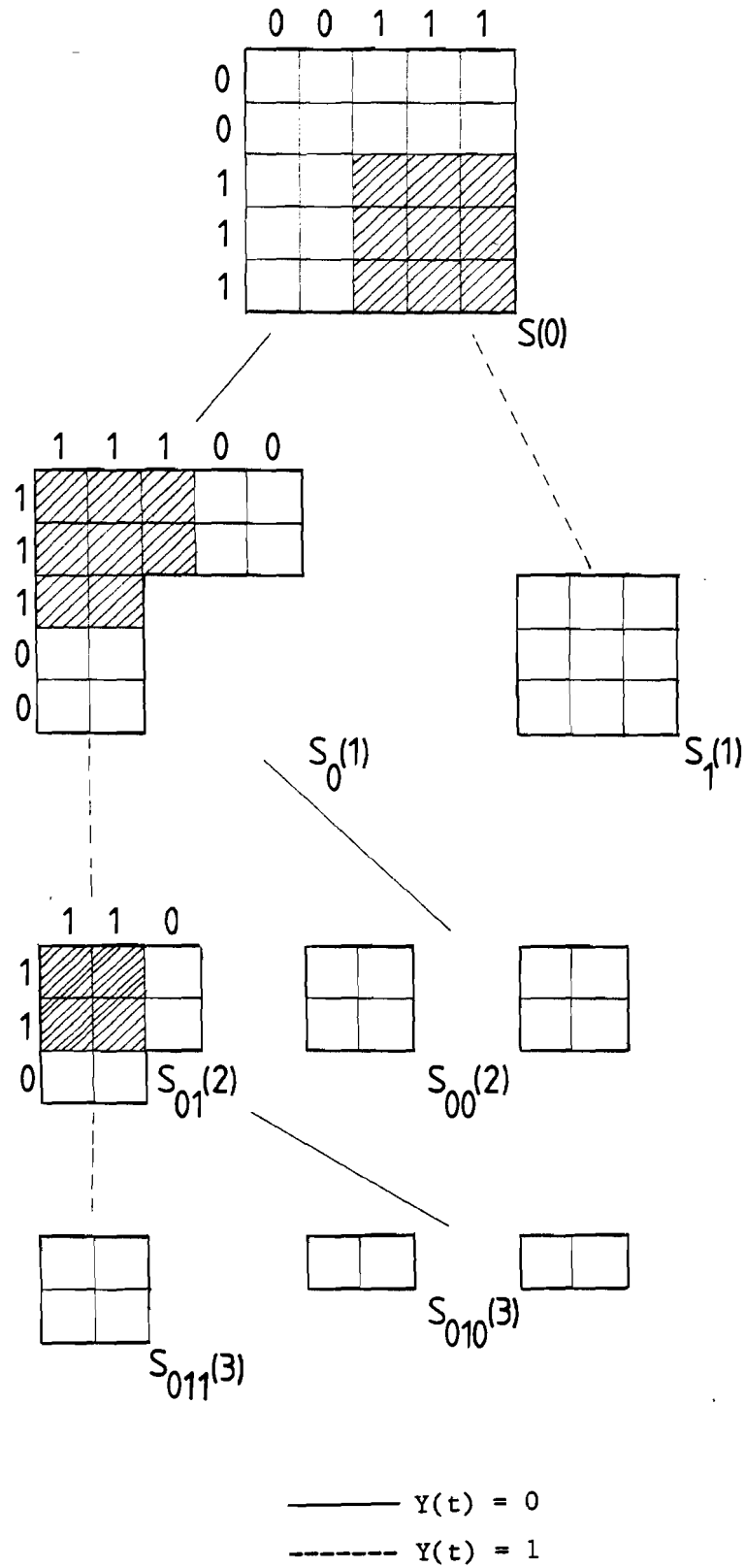


figure 4.2 The Para-schalkwijk strategy for the 5x5 array

It is easily shown, that $\#(5,5)=98$, when we use the formula (3,2) of

chapter 3. ($(K,L)=(3,3)$ and $(I,J)=(3,3)$). However we found a strategy for the 5×5 array, which has the same $\#(5,5)$ as Ligtenberg computed with his computer-search. This division, using our philosophy, is given in figure 4.3.

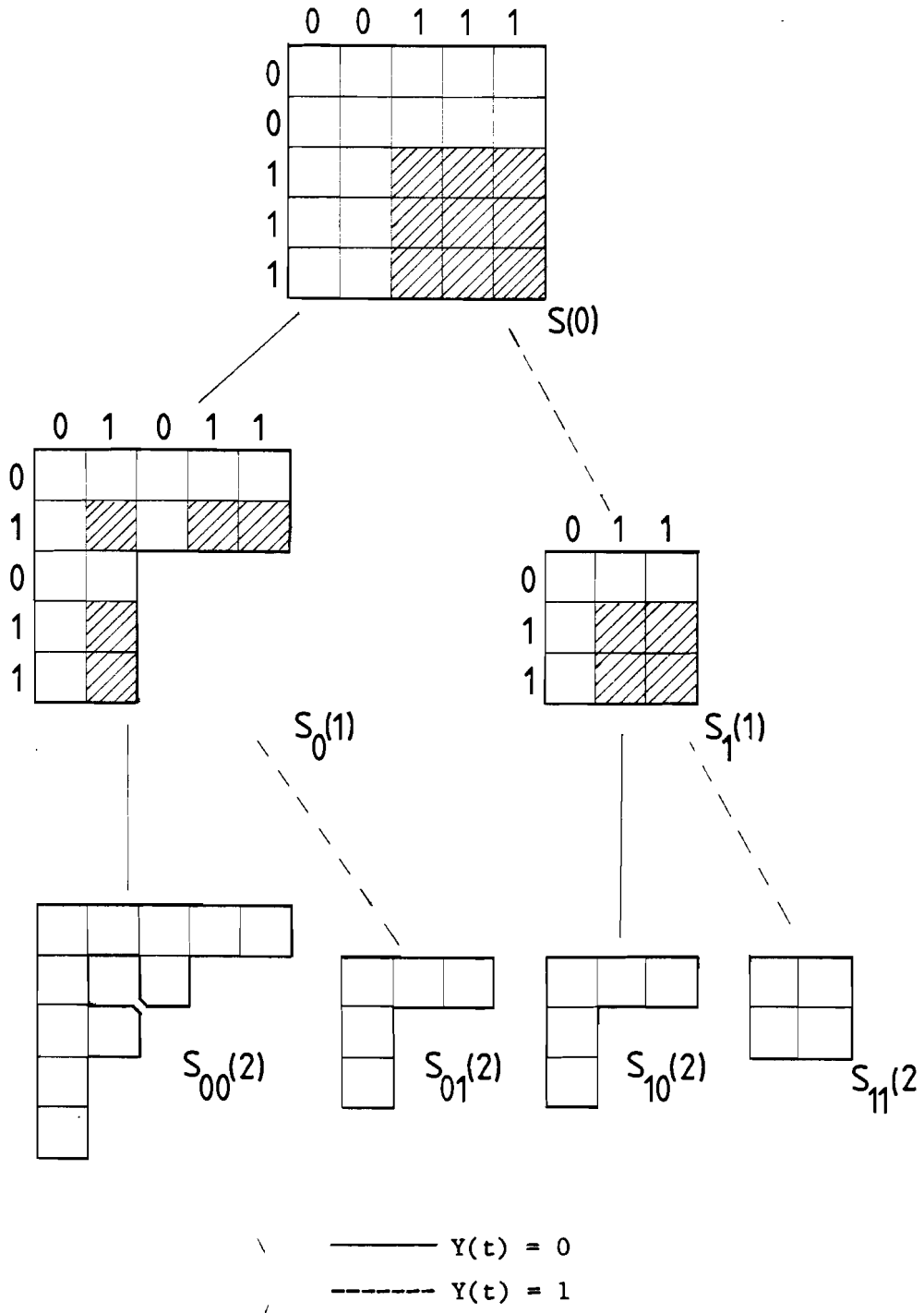


figure 4.3 The Post-Ligtenberg strategy for the 5×5 array

Again we see that $S_{01}(2)=S_{10}(2)$ for this strategy. And when we count the number of transmissions, necessary to single out every message pair of the 5x5 array, we can compute $\#(5,5)$ for this division, and find that $\#(5,5)=58$. But there is one difficulty, we have to deal with. When we look at the $S_{00}(2)$ -figure, it is impossible for us, to give a good strategy, for an "optimal" division of this figure. I.e. this solution was found by trial and error. Unfortunately, until now we have not been able to solve this problem.

In spite of this, we had proved that there exist an other strategy for the 5x5 array, which realizes the same $\#(5,5)$ as the Para-Schalkwijk strategy. Therefore, the last problem did not keep us from investigating greater arrays, using our basic idea, that $S_{01}(2)=S_{10}(2)$. So we investigated the 6x6, 7x7, 8x8, 9x9 and 10x10 array and found the following results.

	Para-Schalkwijk		New strategy	
	$\#(*,*)$	rate	$\#(*,*)$	rate
6 x 6	157	.593	157	.593
7 x 7	232	.593	230	.598
8 x 8	322	.596	319	.602
9 x 9	430	.587	427	.601
10 x 10	556	.597	552	.602

tabel 4.1 First results on the new strategy

The divisions of these arrays, are given in appendix A. Looking at these figures, one will notice that we have not been able to stick to the fact, that $S_{01}(2)=S_{10}(2)$. For example when one looks at our division of the 10x10 array, $S_{01}(2)$ does not equal $S_{10}(2)$. Nevertheless, this the best solution for the 10x10 array, which we found during our investigation, although we cannot prove, that this is the best solution for the 10x10 array. So from now on, we have to deal with two other problems:

- 1 First of all, although the results which we found are as good as, or better than the Para-Schalkwijk strategy, we are not able to prove the optimality of these strategies, i.e. we have never proved that $\#(M,M) = \text{MIN}(M)$ for the arrays of which we are talking now;
- 2 Secondly, although the idea seemed to be good, we showed, that it is not a universal idea, i.e. we proved, that there exist better strategies for certain arrays, when we do not stick to our basic idea, that $S_{01}(2)$ equals $S_{10}(2)$.

5 FURTHER RESULTS ON OUR STRATEGY

Despite the problems, we met during our investigation, we believed that our idea was good enough to improve the results we found until now. So we investigated the ways, how we could use our philosophy for greater arrays. The main thought was, that we should avoid the problem, that the remaining figure (for instance $S_{00}(2)$ in the 5×5 division) is too big, to be investigated by hand, i.e. at a certain moment, it must be possible to find the best division of the remaining figure by trial and error. This means, that we must give a description of a strategy, which tells us, how to handle after 2,3 or even 4 transmissions. We succeeded in our investigation and found the sequence 5×5 , 8×8 , 12×12 , 18×18 , 27×27 , 40×40 and 58×58 , as a sequence of arrays for which our strategy produced rates, which are much greater than the rates Ligtenberg [3] found with his computer-search. In the last part of this paper, we will discuss these strategies.

The basic idea is again, that we have to use the "good divisions" of the arrays which we have already solved. Not once, as we did in the 5×5 array, but as many times as possible. For example, let us discuss the 18×18 array. (figures of this division are all drawn in appendix A) For the clearness of the discussion, we will introduce a new notation. When we talk about a (sub-)figure of the 12×12 array in this discussion, we shall indicate these as: $S_{\dots}^*()$. The (sub-)figures of the 18×18 array are given without the "*". So we shall indicate these as: $S_{\dots}()$.

In the first transmission, one excludes the 12×12 array. Looking at the division of the 12×12 array, we see that the sizes of the $S_0^*(1)$ -figure are exactly the same as the sizes of the $S_{01}(2)$ -figure. So this means for the 18×18 division that $S_{01}(2)$ equals $S_{10}(2)$. However the remaining part of the 18×18 division ($S_{00}(2)$) is too big to investigate by hand. So one singles out the $S_{00}(2)$ -figure of the 12×12 division. This means for the 18×18 division that $S_{100}(3)$ equals $S_{001}(3)$ and what remains is the $S_{000}(3)$ -figure, which is small enough to investigate by trial and

error. The solution of all the other (sub-)figures are known by the user, because they are used by the investigation of the smaller arrays. So we can compute $\#(18,18)$, and find that $\#(18,18)=2216$. With formula (2.18) of chapter 2, we now can compute the rate of this strategy, and find $R_\delta = .60968$ which is much better then the computer-search of Ligtenberg [3], because he found $\#(18,18)=2245$, what means that $R_\delta = .6018$.

So we investigated the next array, namely the 27×27 array, and searched a new division, using the 18×18 array and the same starting-points (i.e. $S_{01}(1)=S_{10}(1)$, $S_{100}(2)=S_{001}(2)$ and S_{000} by trial and error), and found $\#(27,27)=5683$, which means that the rate $R_\delta = .60994$.

In an analogous way, we searched for "good divisions" for the 40×40 and 58×58 array. In these divisions, we also used that $S_{1000}(3)=S_{0001}(3)$. The next table shows the results.

	Para-Schalkwijk		New strategy	
	$\#(*,*)$	rate	$\#(*,*)$	rate
5 x 5	98	.5923	98	.5923
8 x 8	322	.5962	319	.6019
12 x 12	860	.6002	848	.6088
18 x 18	2245	.6018	2216	.6097
27 x 27	5746	.6032	5683	.6099
40 x 40	14091	.6043	13943	.6107
58 x 58	32580	.6049	32250	.6111

table 5.1 Further results on the new strategy

The next figure shows the results, compared with the Para-Schalkwijk results. In fact, it is figure 3.3 plus the isolated points of the new strategy of Post and Ligtenberg.

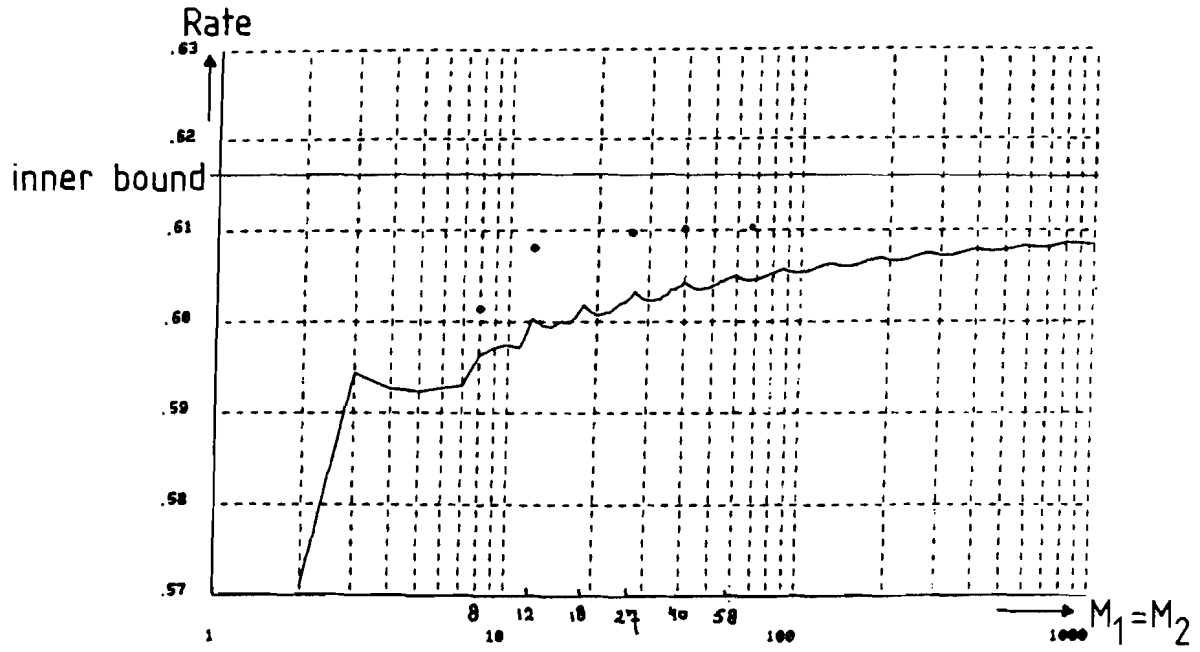


figure 5.1 The results of the Post-Ligtenberg strategy

6 CONCLUSIONS AND RECOMMENDATIONS

Finding the results, described in the last two chapters, convinced us that there must be a strategy for an array smaller than 100×100 , which division delivers a rate greater than Shannon's inner bound. However, we think that one must use other ideas to reach this.

Let us discuss the results. Looking at figure 5.1, one notices that the increase of the rates of the arrays, which we investigated, is indeed great. However, when we draw a figure through these few points, the increase is going slower and slower for greater M . On the other hand, we do believe that we have found the best, or almost the best divisions for the small arrays ($M < 10$) and that we have not found them for the greater arrays ($M=18,27,40$ and 58) which we investigated. So we are convinced that there exist other strategies which are better than the one we use now.

The idea that there must be better strategies, came up during our investigation, and was found by accident. This happened as follows: For a faster and more reliable computation of new strategies, we developed a computer-program which is described in appendix B. This program is interactive, and gives the user the possibility to introduce his strategy step by step. One day, when we were trying out a strategy, we excluded by accident an a-symmetric part from a symmetric sub-figure. The result however, of this strategy was identical to a symmetric division we found six months before. This was remarkable, because until then, we only used symmetric strategies for symmetric (sub-)figures, because we thought that this was the best way to solve them. So we investigated that particular sub-figure by trial and error and concluded that the a-symmetric solution was better than every symmetric one. This information might be a good starting point, for further investigation on this subject.

An other starting-point is perhaps, to continue the investigations of Hollmann [4], i.e. to derive lower bounds for the discrete case of the AND-CHANNEL. These bounds are necessary, to help the computer, finding better divisions of the (sub-)figures. Because, without these restrictions it is impossible to do an automatic computer-search over all possible strategies. The number of strategies is for $M=10$ already too large to run on the computer. In the future, we shall also need these bounds, in order to prove that our divisions are optimal

ACKNOWLEDGEMENT

The author wishes to thank Prof. Dr. Ir. J.P.M. Schalkwijk and Dr. K.A. Post for the informative discussions during my graduation on the Eindhoven University of Technology. Their criticism was also very useful, by the preparation of this manuscript. I also want to thank all members of the group of Prof. Schalkwijk for the pleasant time, which I spent with them.

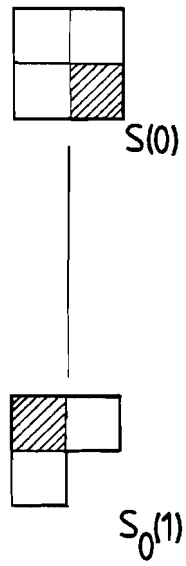
REFERENCES:

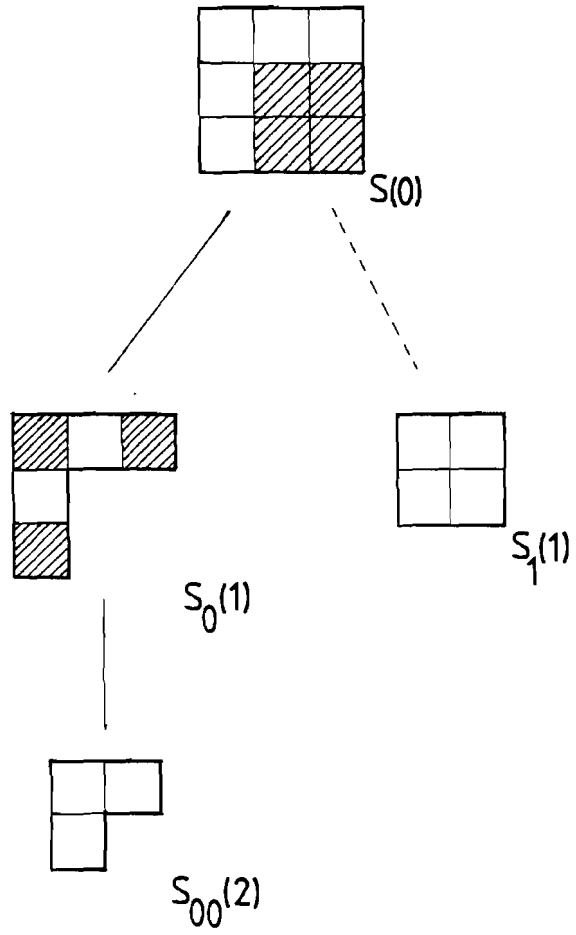
- [1] C.E. Shannon, "Two-way communication channels", Proc. 4th Berkeley Symp. Math. Statist. and Prob., vol. 1, pp. 611-644, 1961.
Reprinted in Key papers in the development of Information Theory, (D. Slepian), Ed.) New York: IEEE Press, 1974, pp. 339-372.
- [2] J.P.M. Schalkwijk, "The binary multiplying channel - A coding scheme that operates beyond Shannon's inner bound region", IEEE Trans. Inform. Theory, vol. IT-28, pp. 107-110, Jan 1982.
- [3] L.G.T.M. Ligtenberg, "De overschreiding van de inner bound in het discrete geval, voor het binary multiplying channel", trainee report, Eindhoven University of technology, Department of Electrical Engineering, Group Information- and Communication Theory, Sept. 1981. (Dutch)
- [4] H. Hollmann, "Het binary multiplying channel", trainee report, Eindhoven University of Technology, Department of Electrical Engineering, Group Information- and Communication Theory, Jan. 1982. (Dutch)

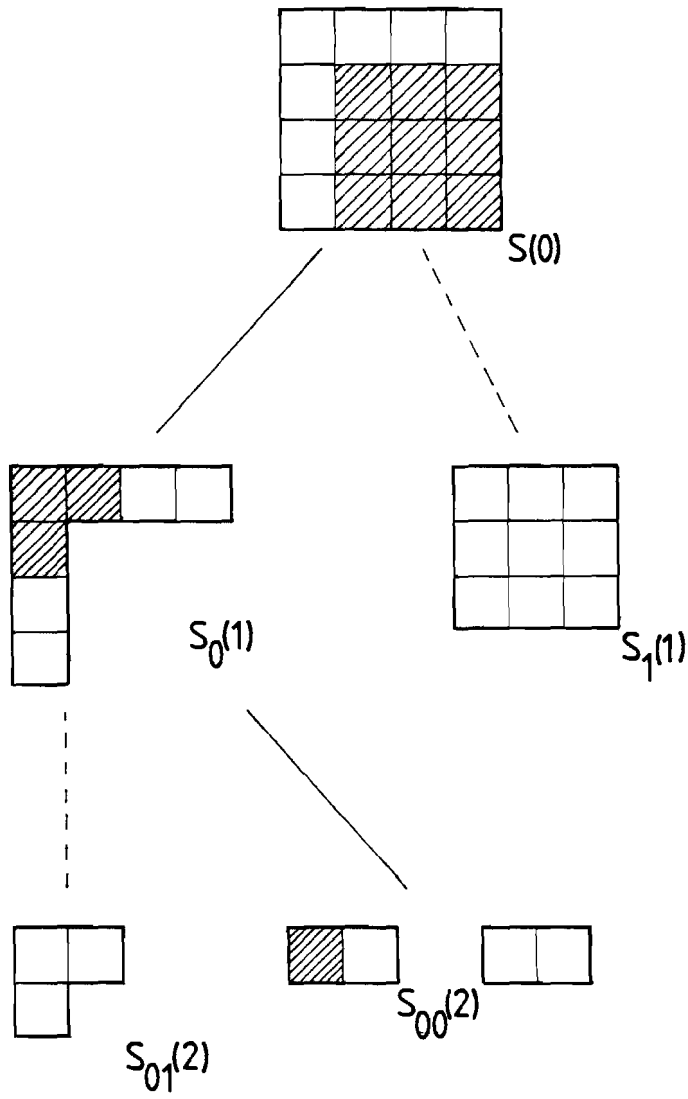
The next table gives the results of the divisions, which are drawn in this appendix.

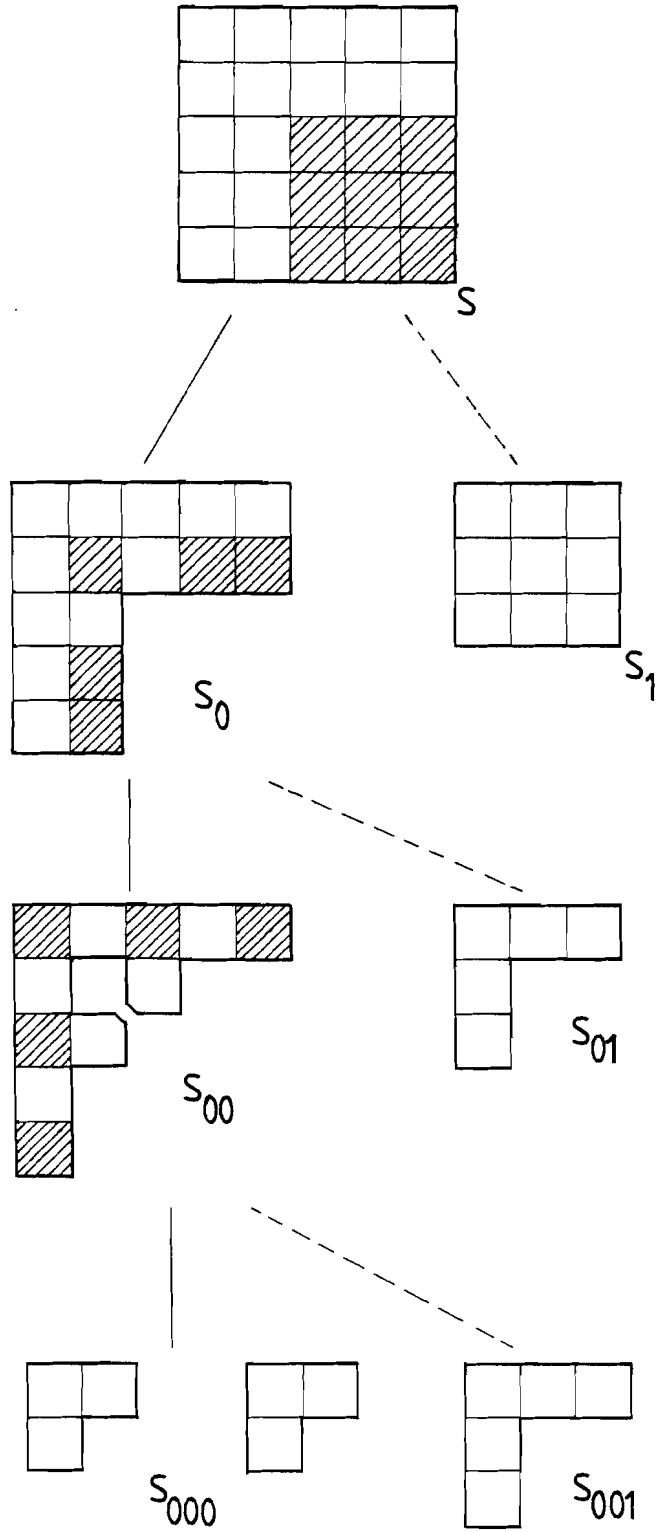
	#(*,*)	Rate	Page
2 x 2	7	.57143	A-3-
3 x 3	24	.59436	A-4-
4 x 4	54	.59259	A-5-
5 x 5	98	.59233	A-6-
6 x 6	157	.59273	A-7-
7 x 7	230	.59809	A-8-
8 x 8	319	.60188	A-10-
9 x 9	427	.60132	A-12-
10 x 10	552	.60180	A-13-
11 x 11	690	.60665	A-14-
12 x 12	848	.60876	A-16-
13 x 13	1031	.60657	A-18-
16 x 16	1681	.60916	A-19-

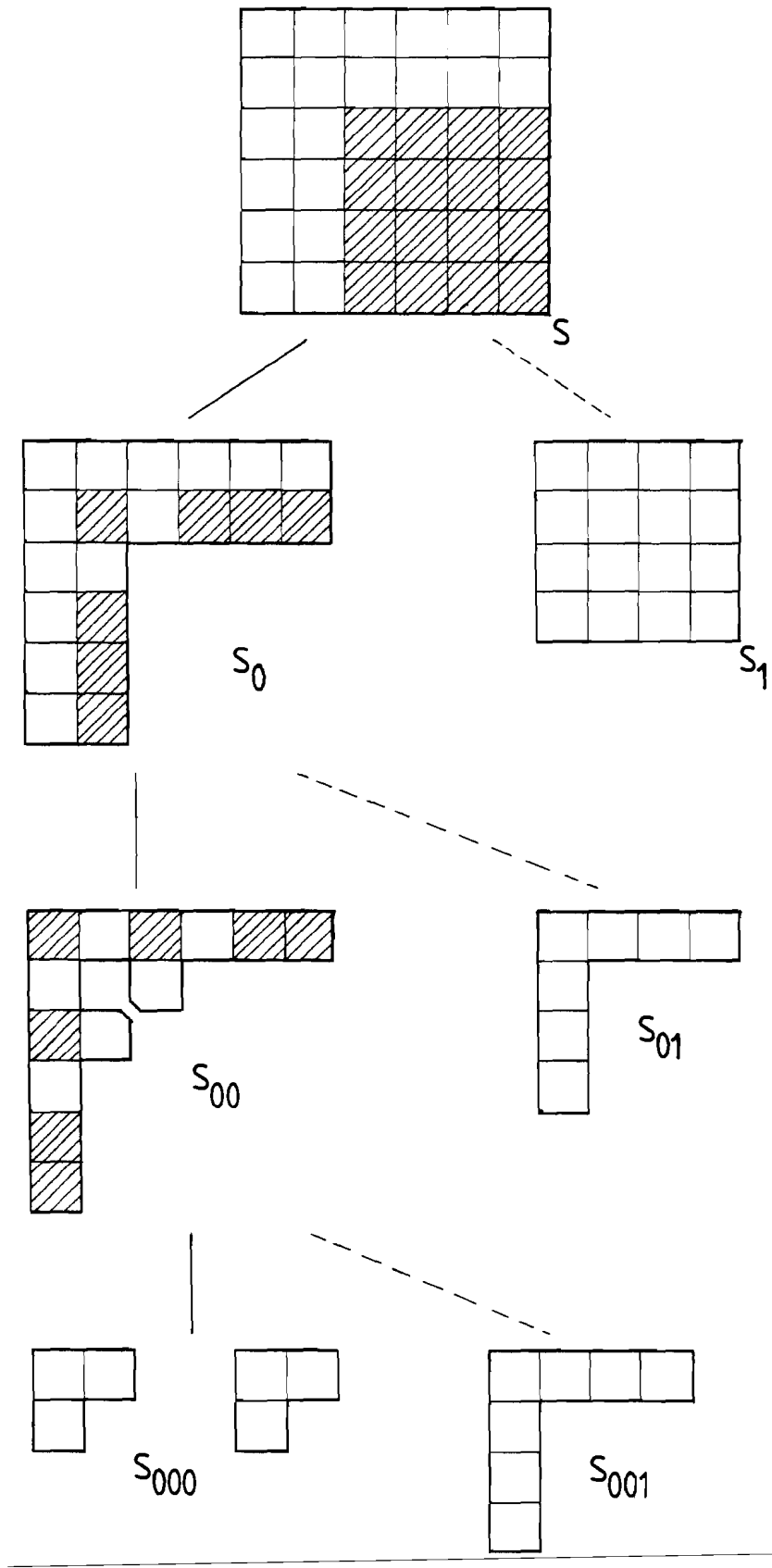
	#(*,*)	Rate	Page
17 x 17	1938	.60953	A-21-
18 x 18	2216	.60968	A-23-
27 x 27	5683	.60994	A-26-
40 x 40	13943	.61070	A-28-
58 x 58	32250	.61105	A-34-

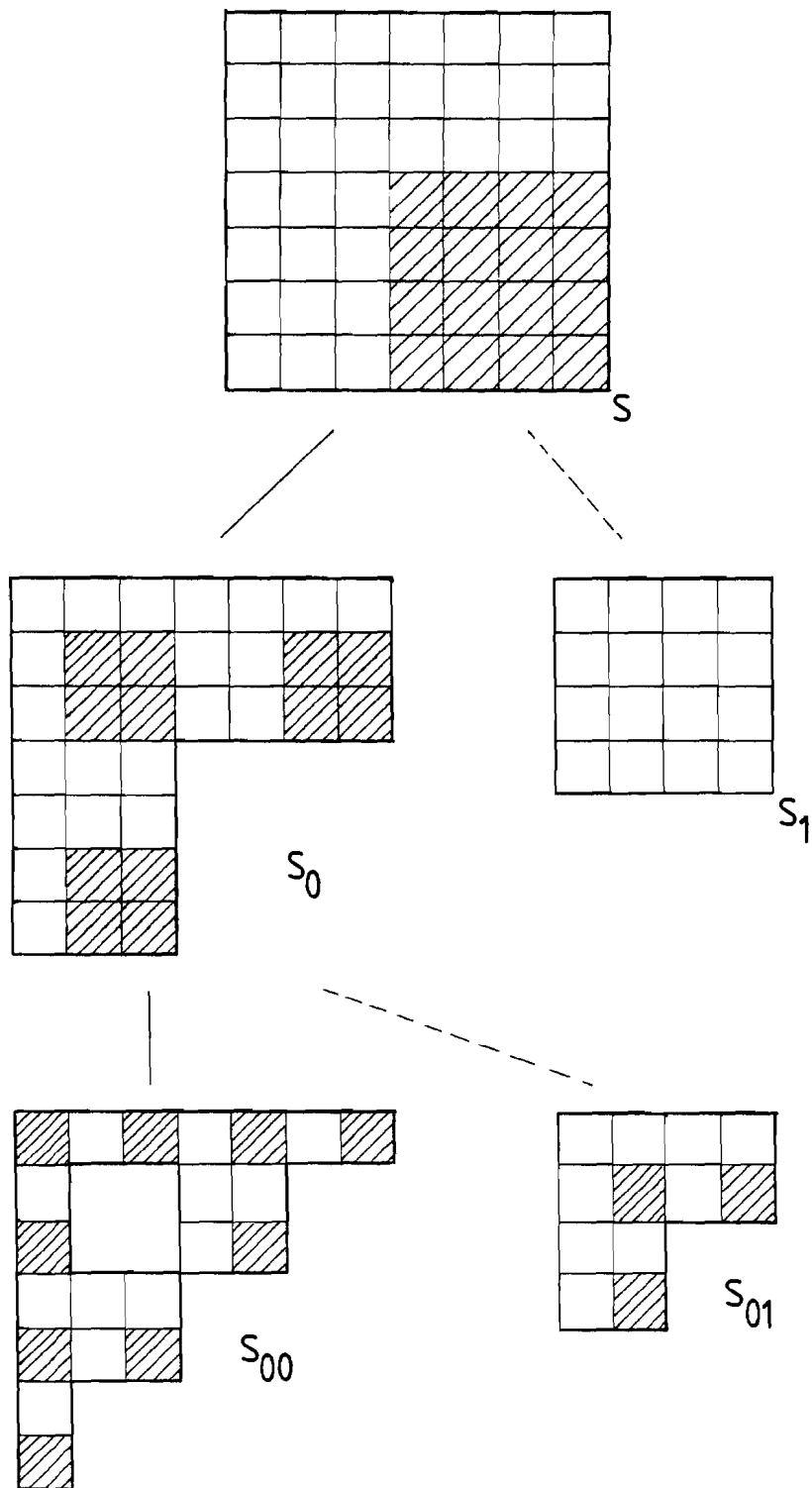


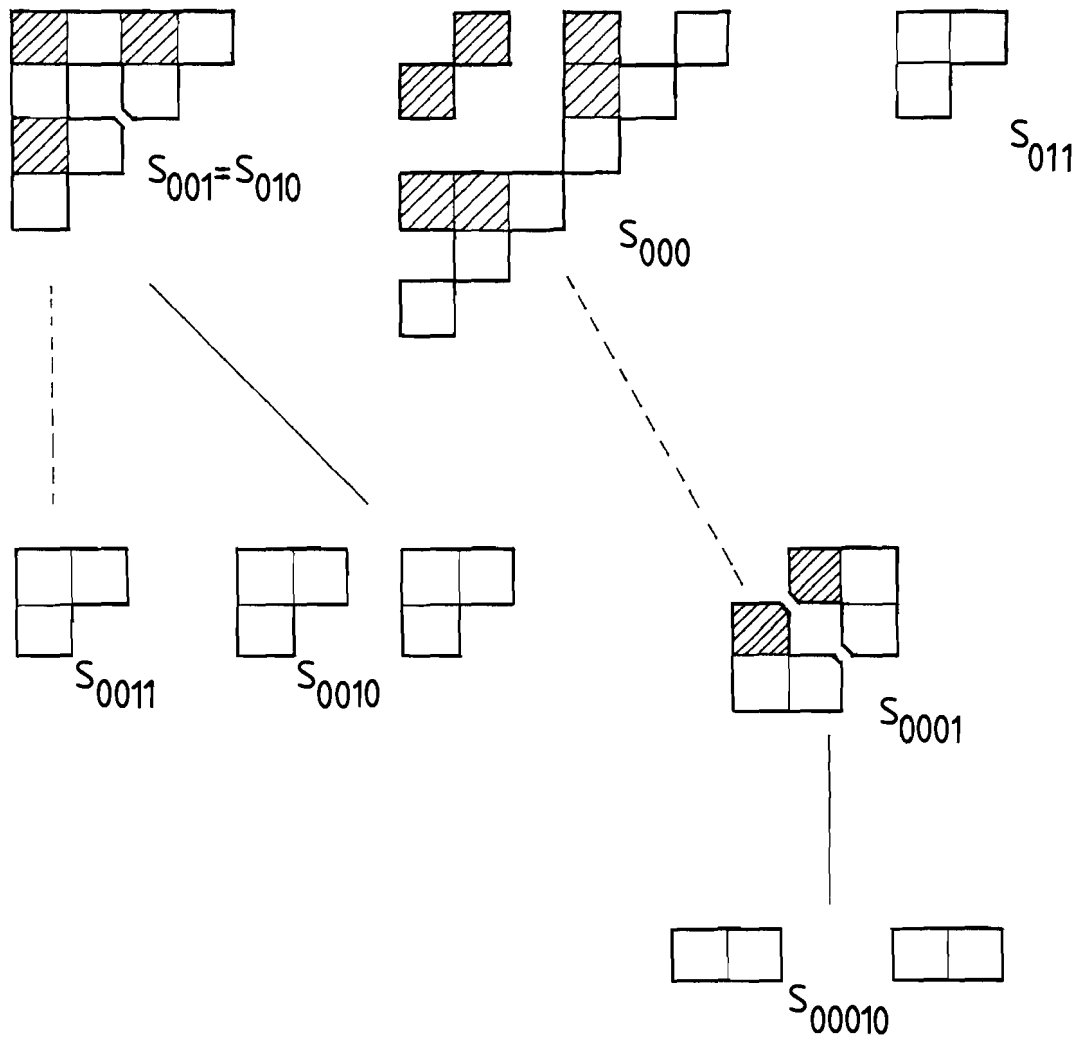


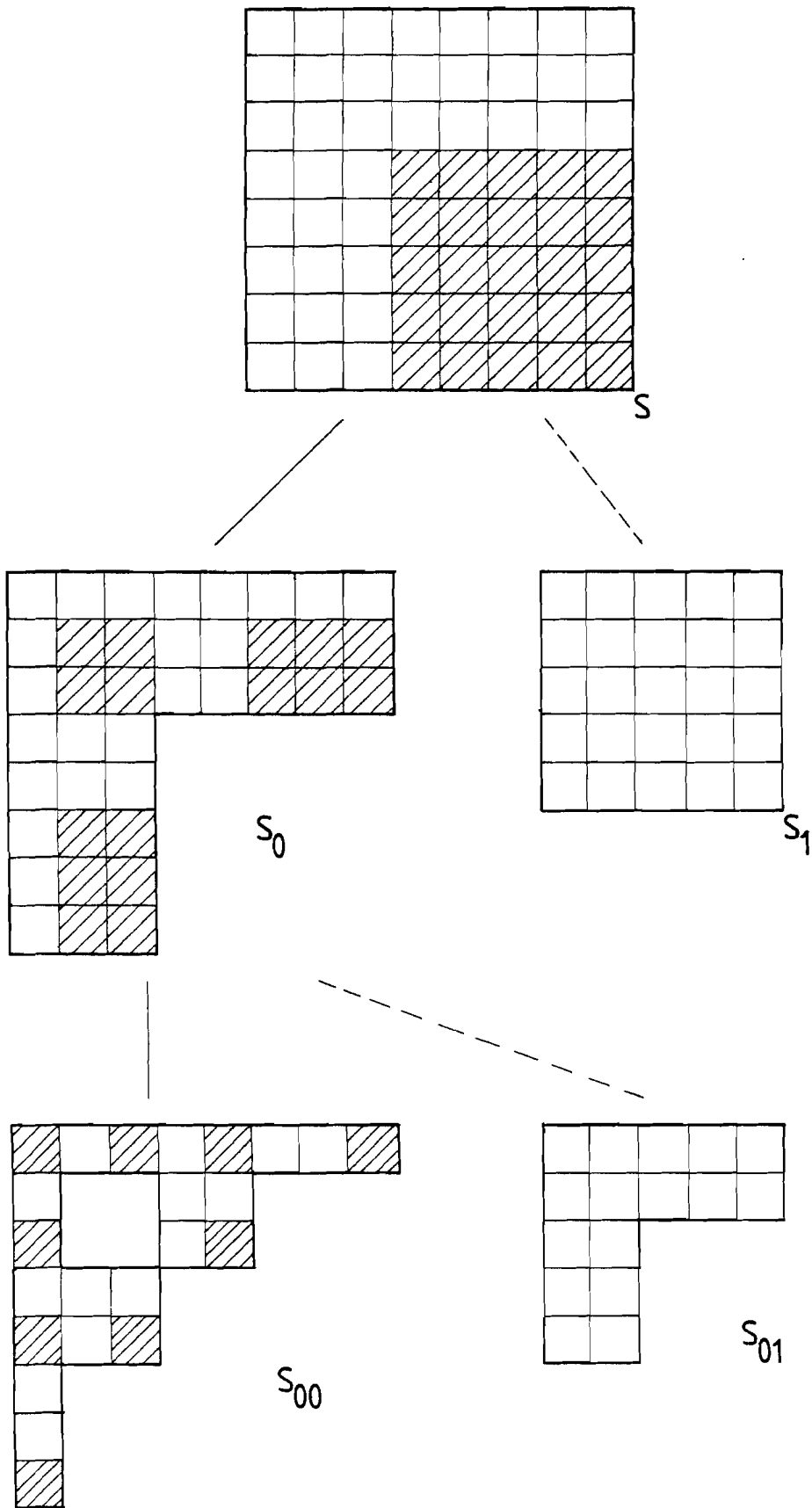


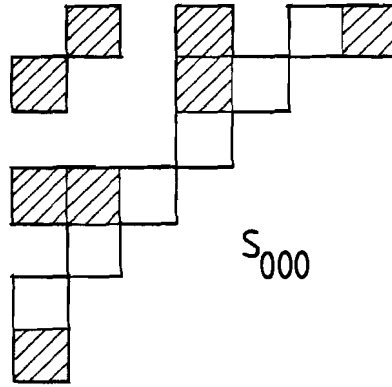
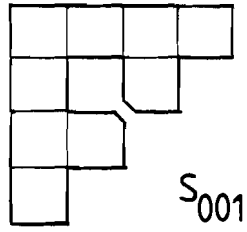




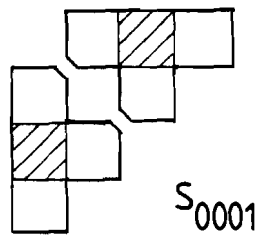




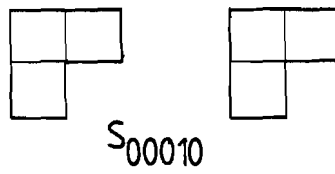


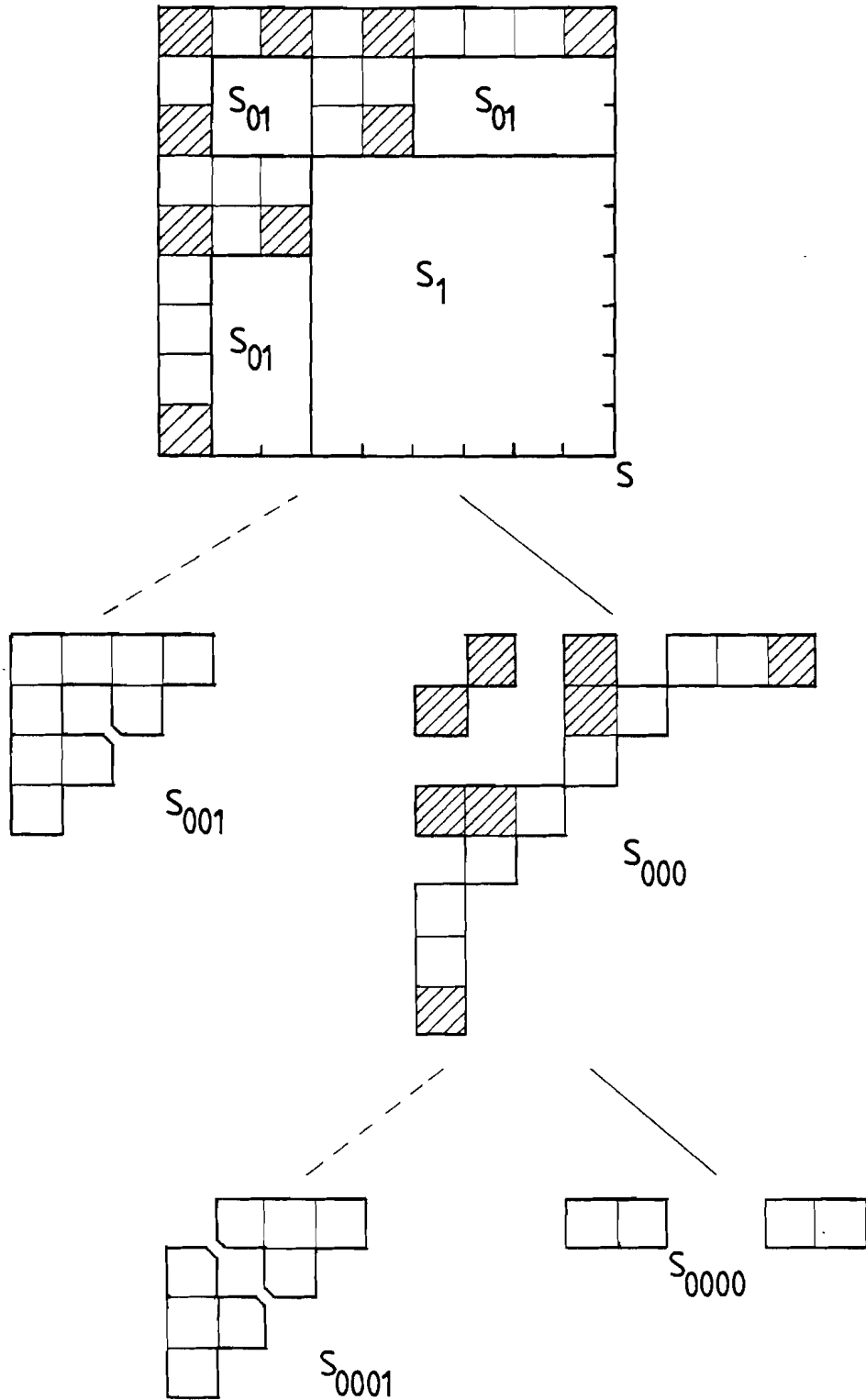


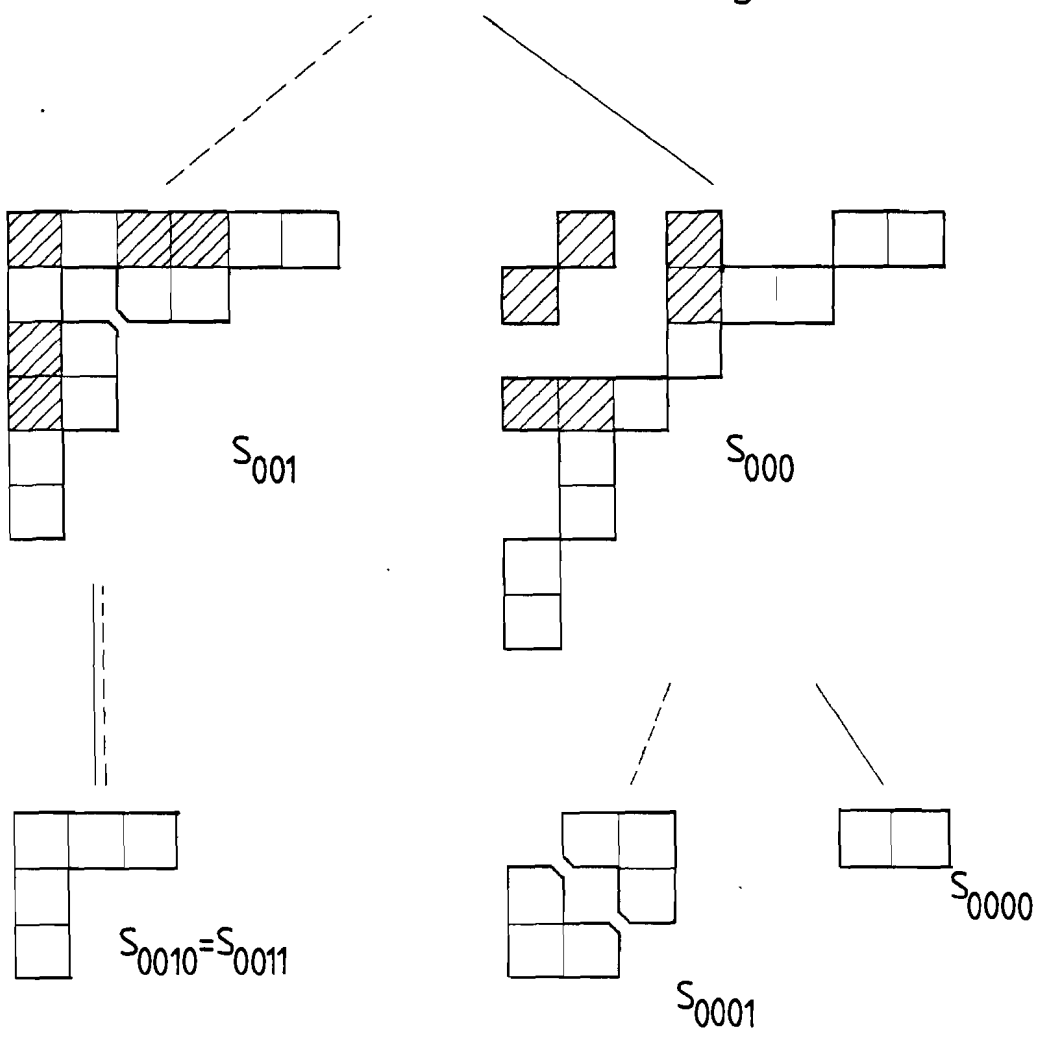
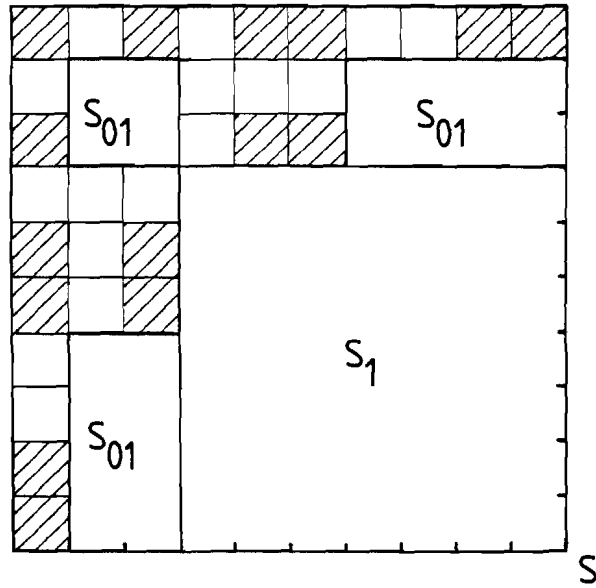
⋮

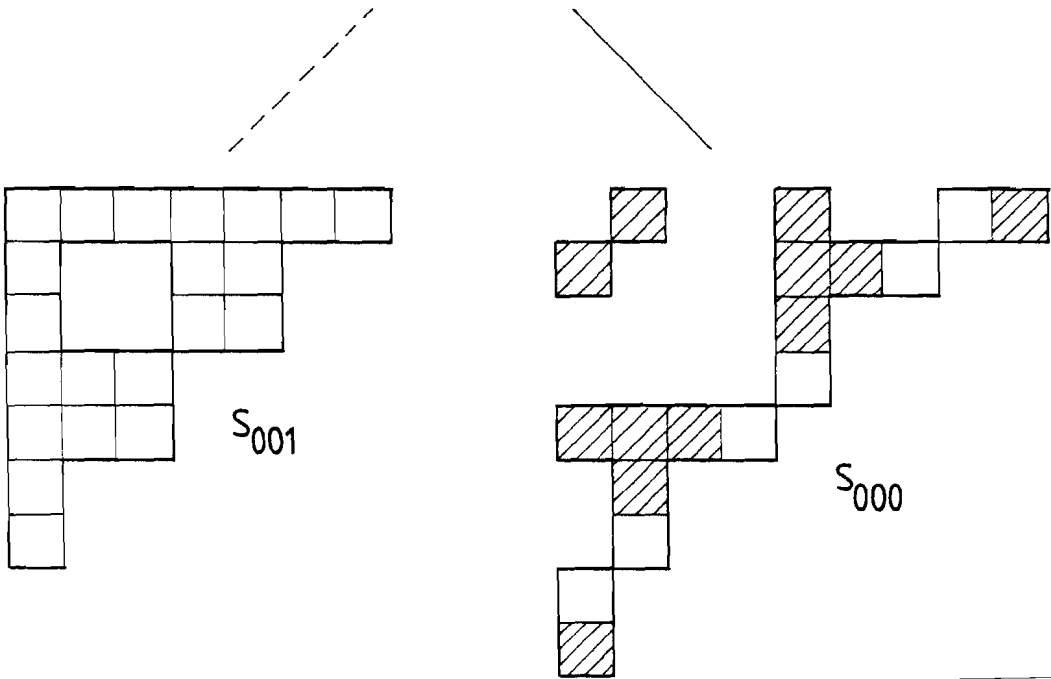
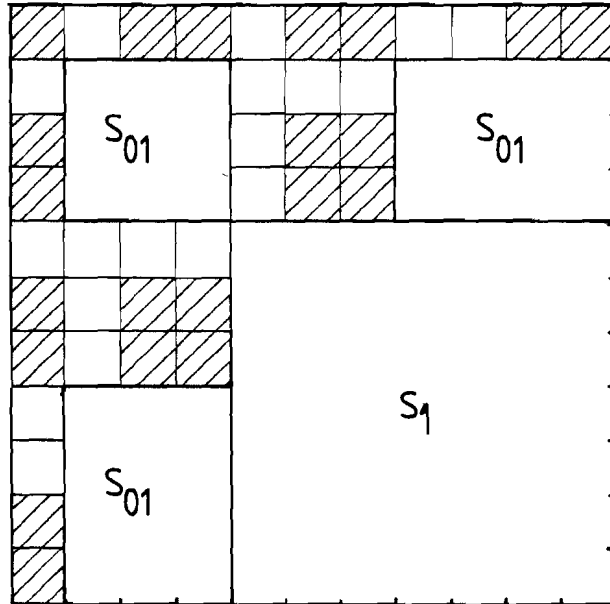


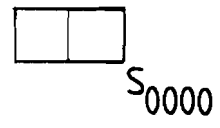
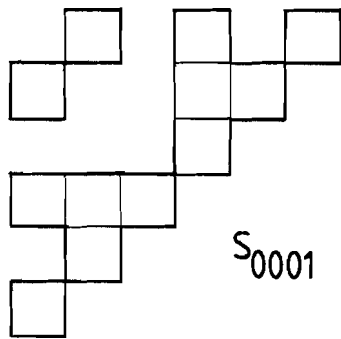
|

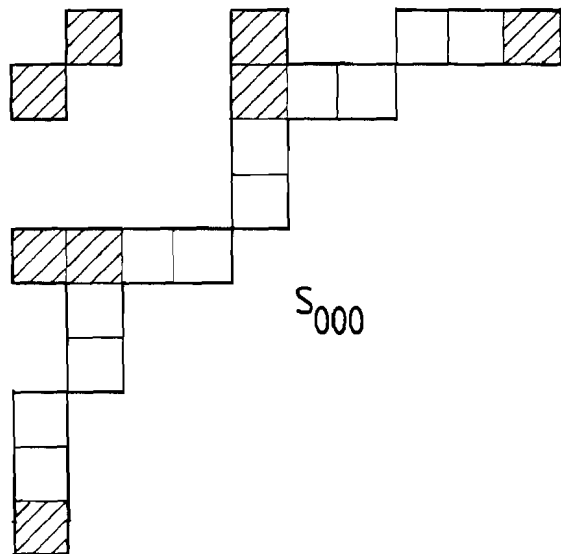
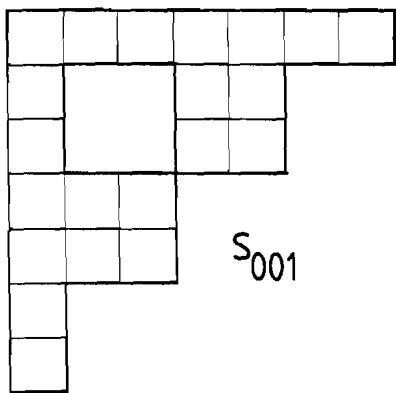
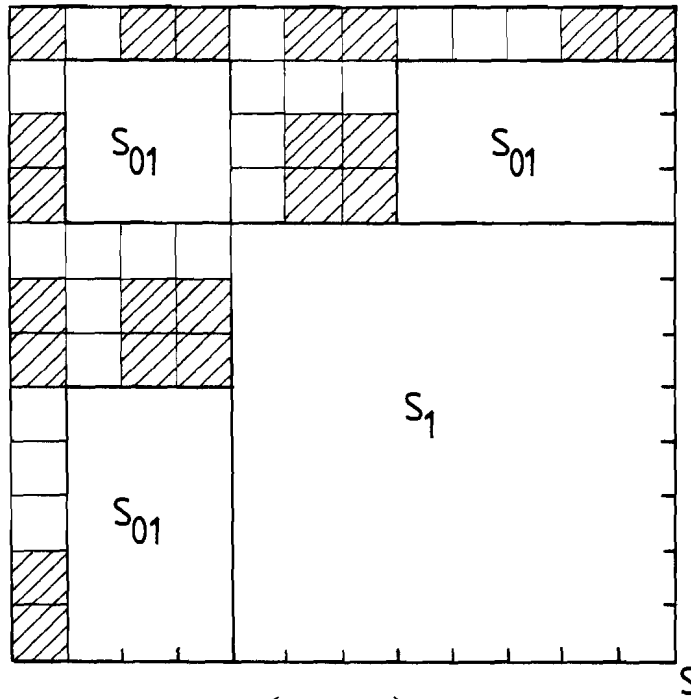


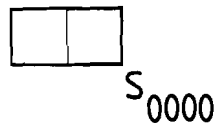
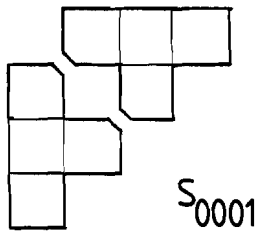


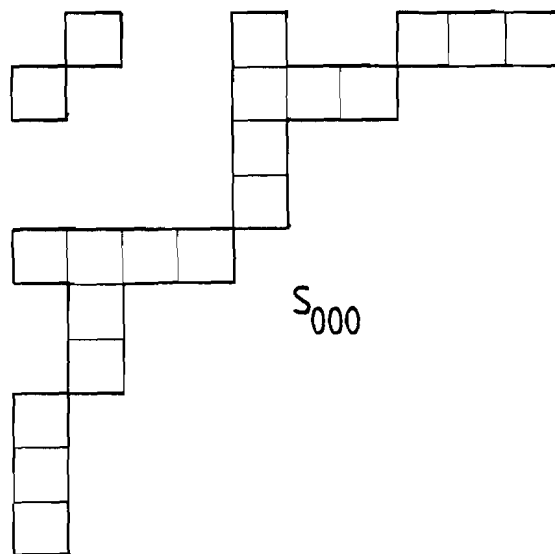
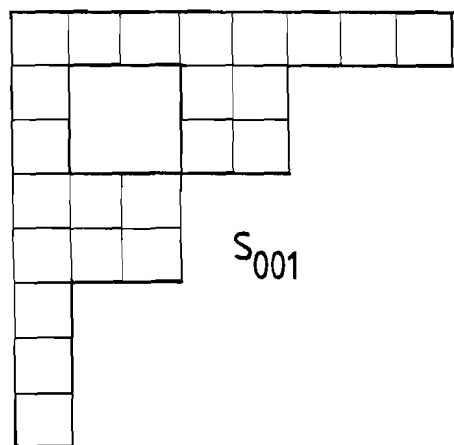
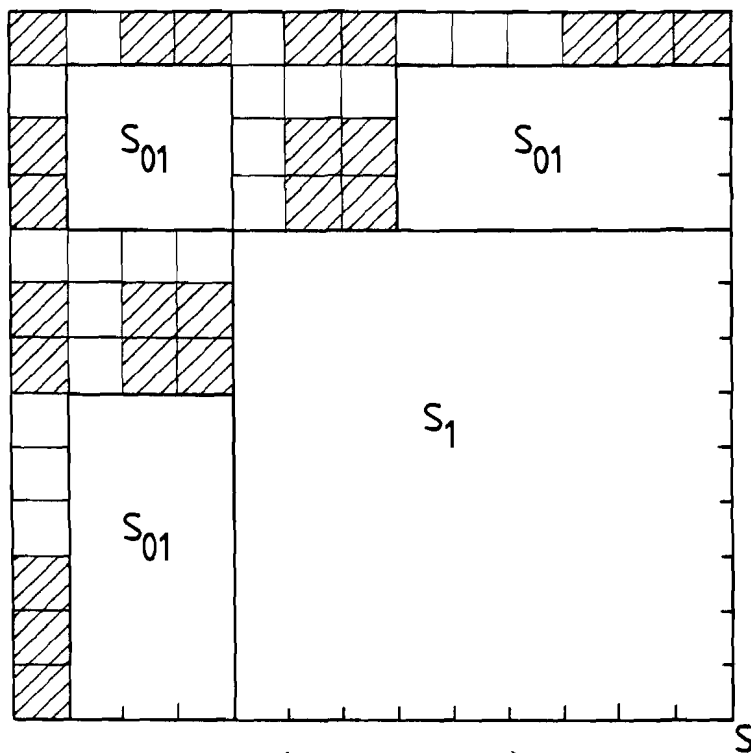


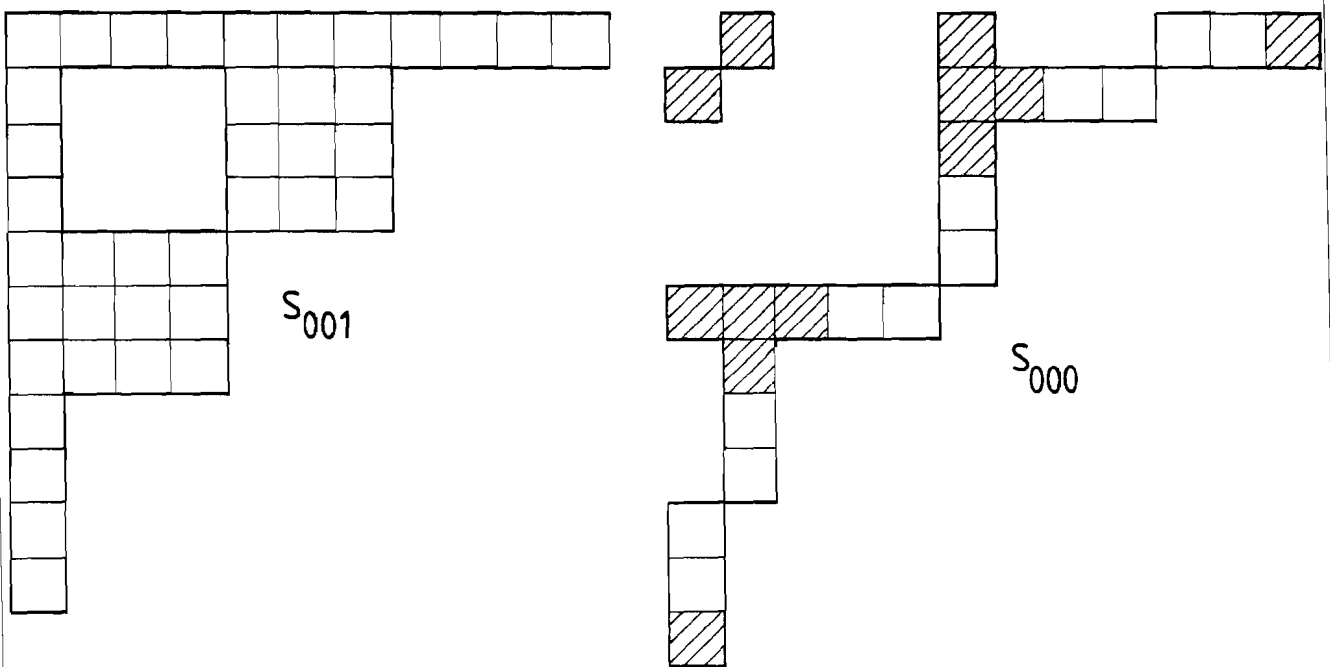
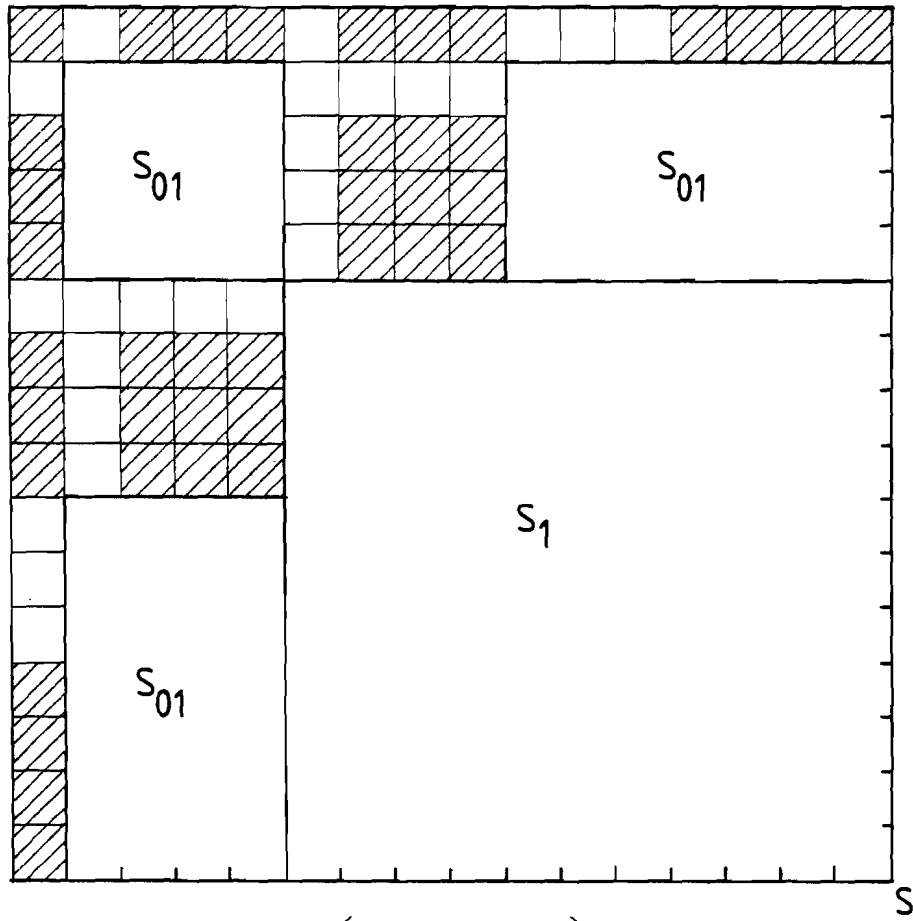


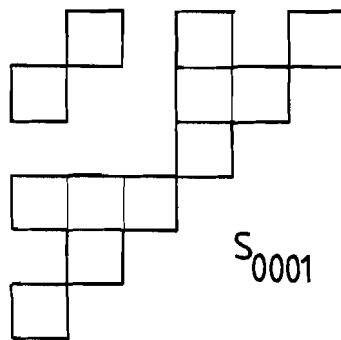


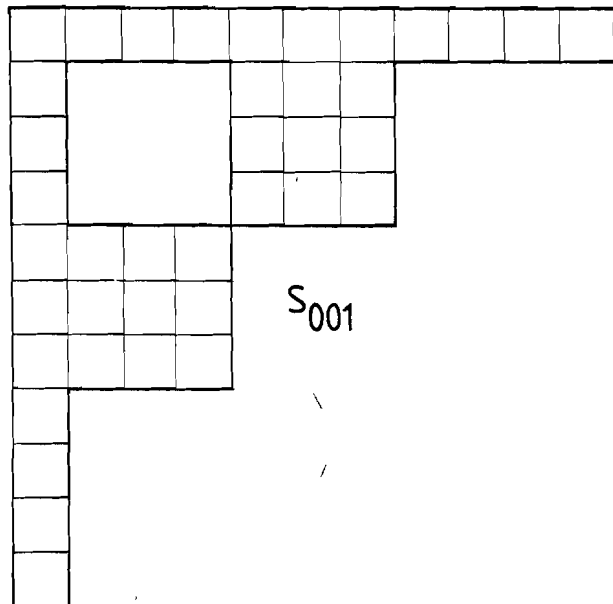
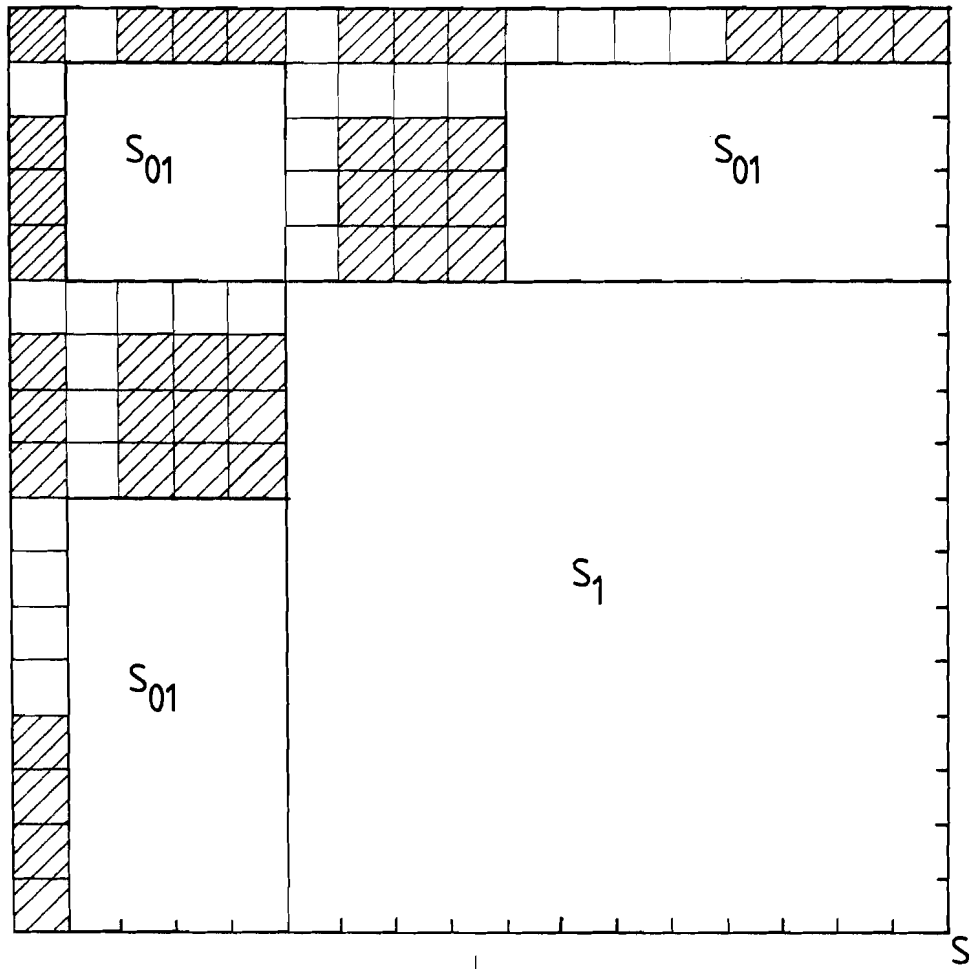


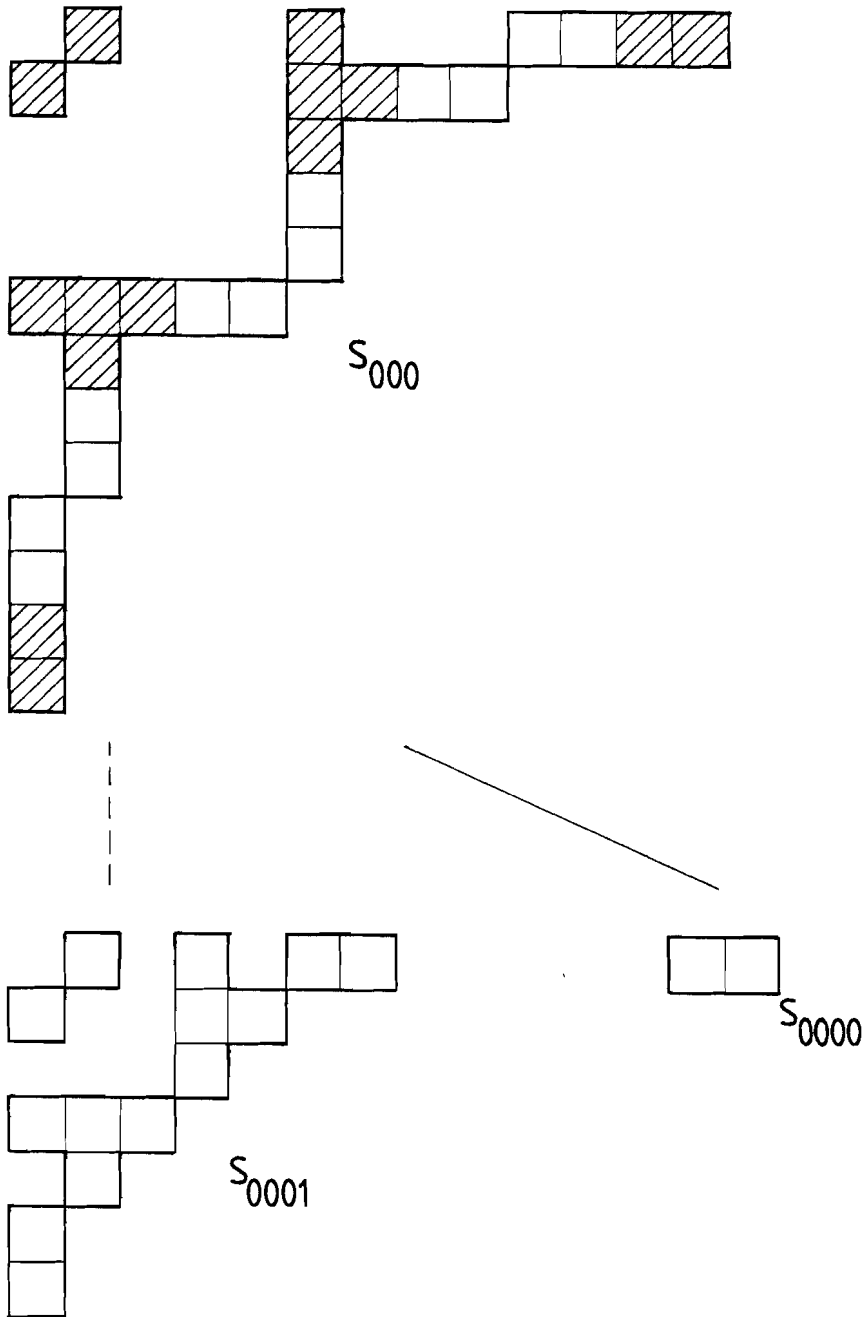


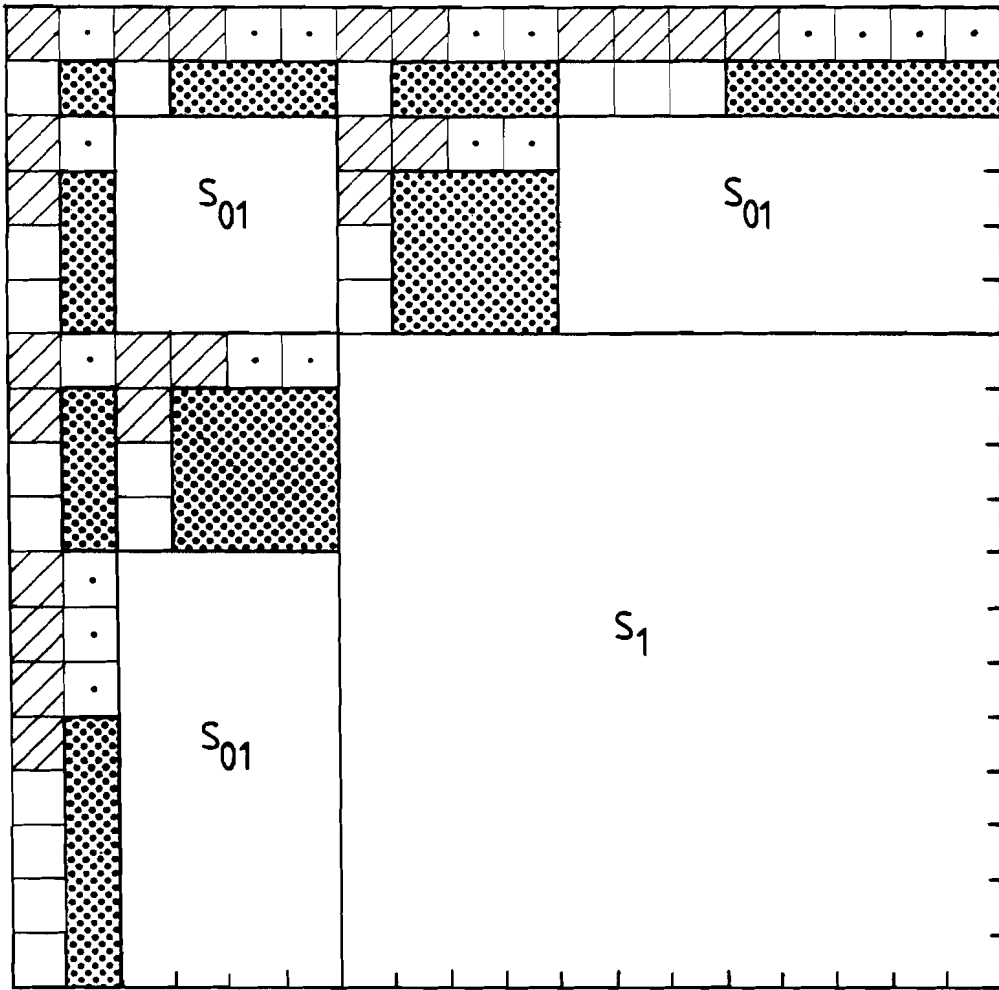




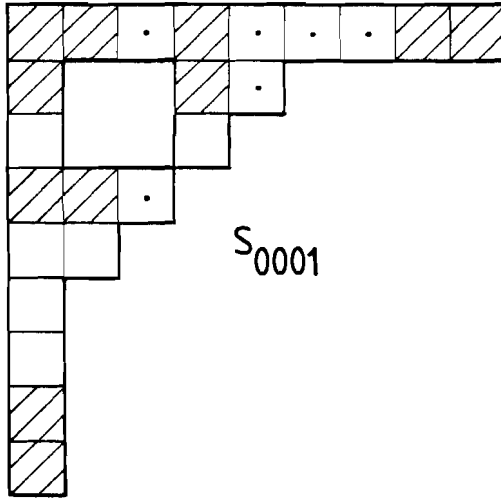




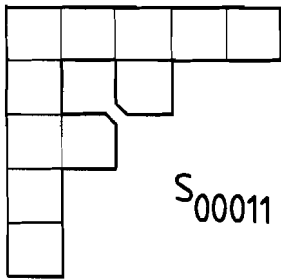




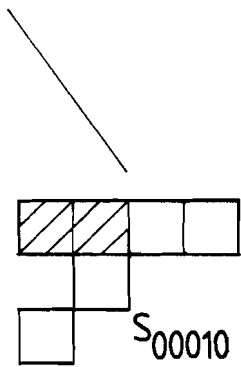
 S₀₀₁



S_{0001}



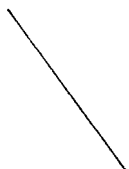
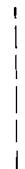
S_{00011}

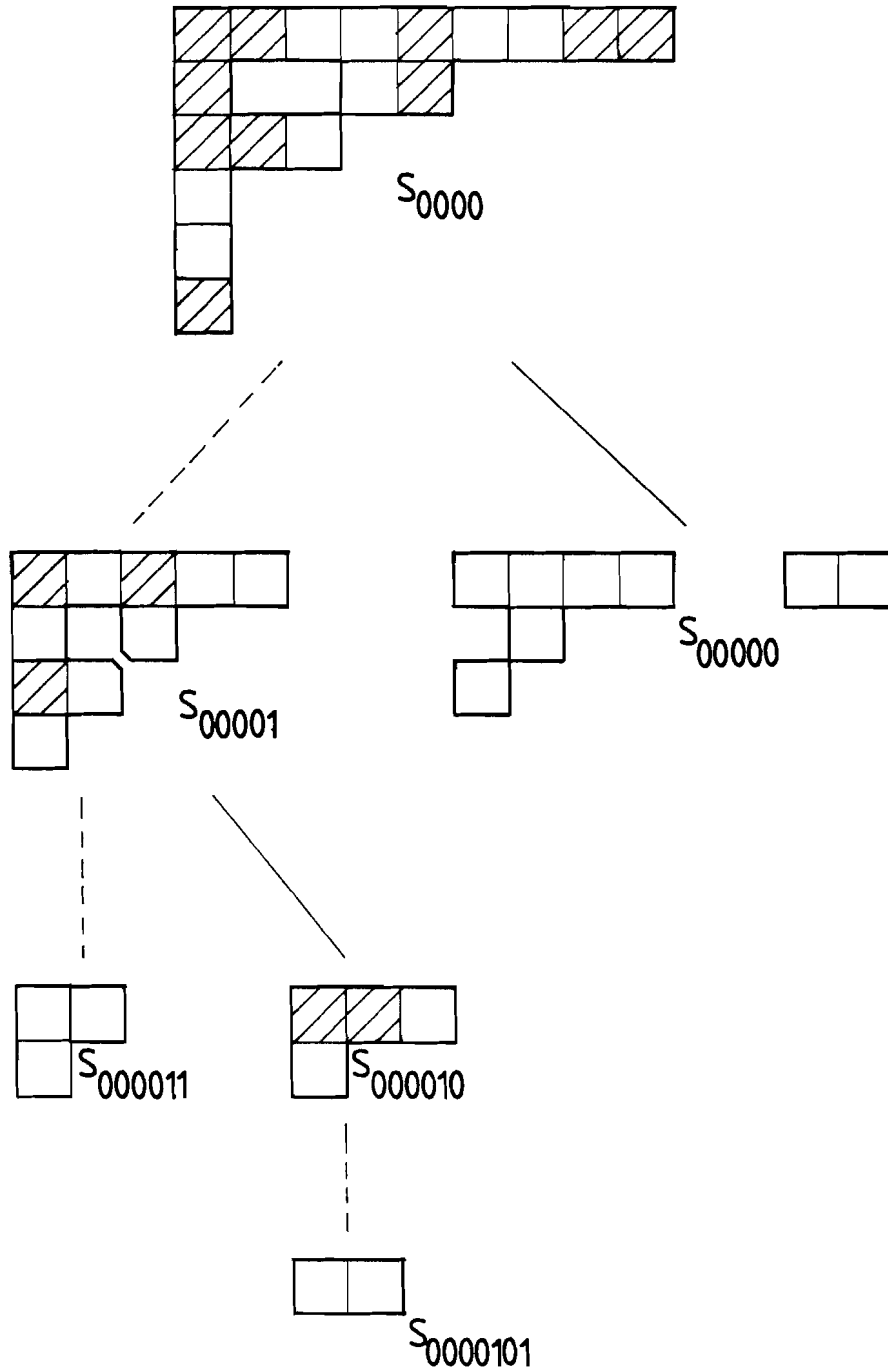


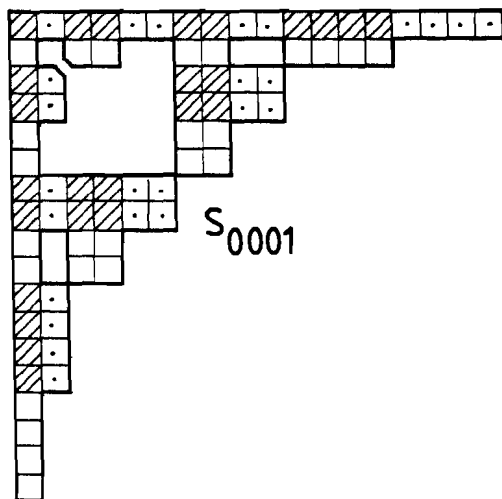
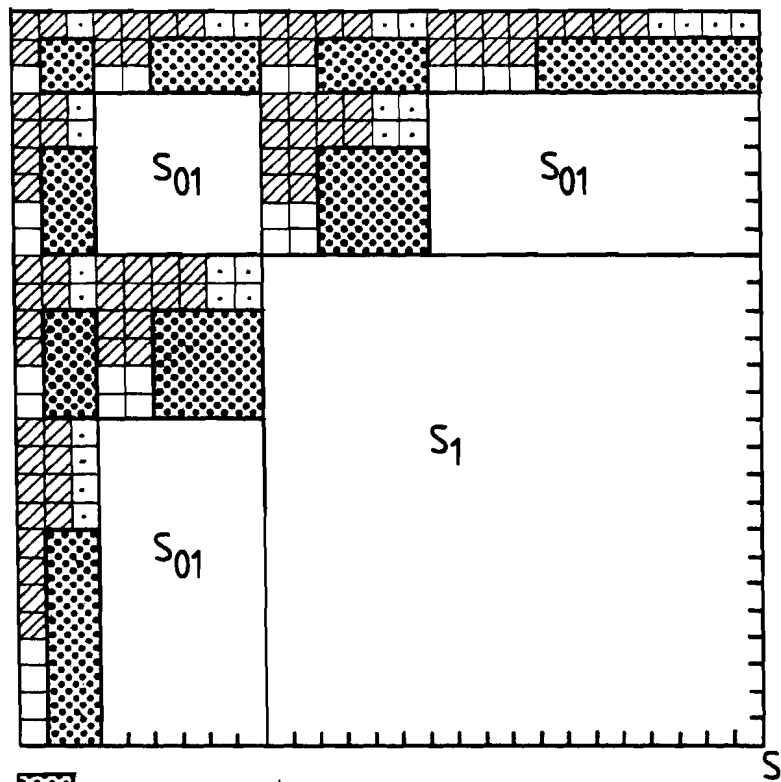
S_{00010}

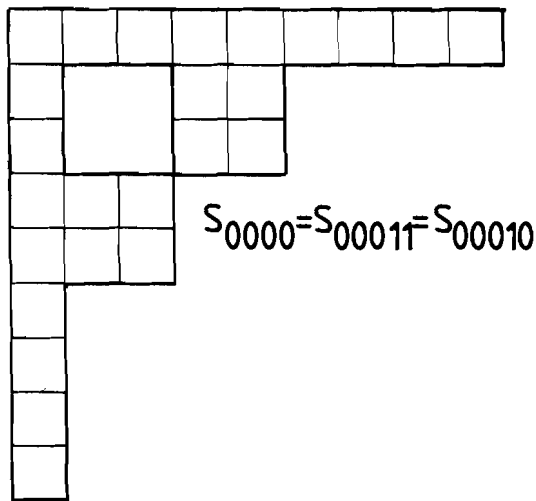


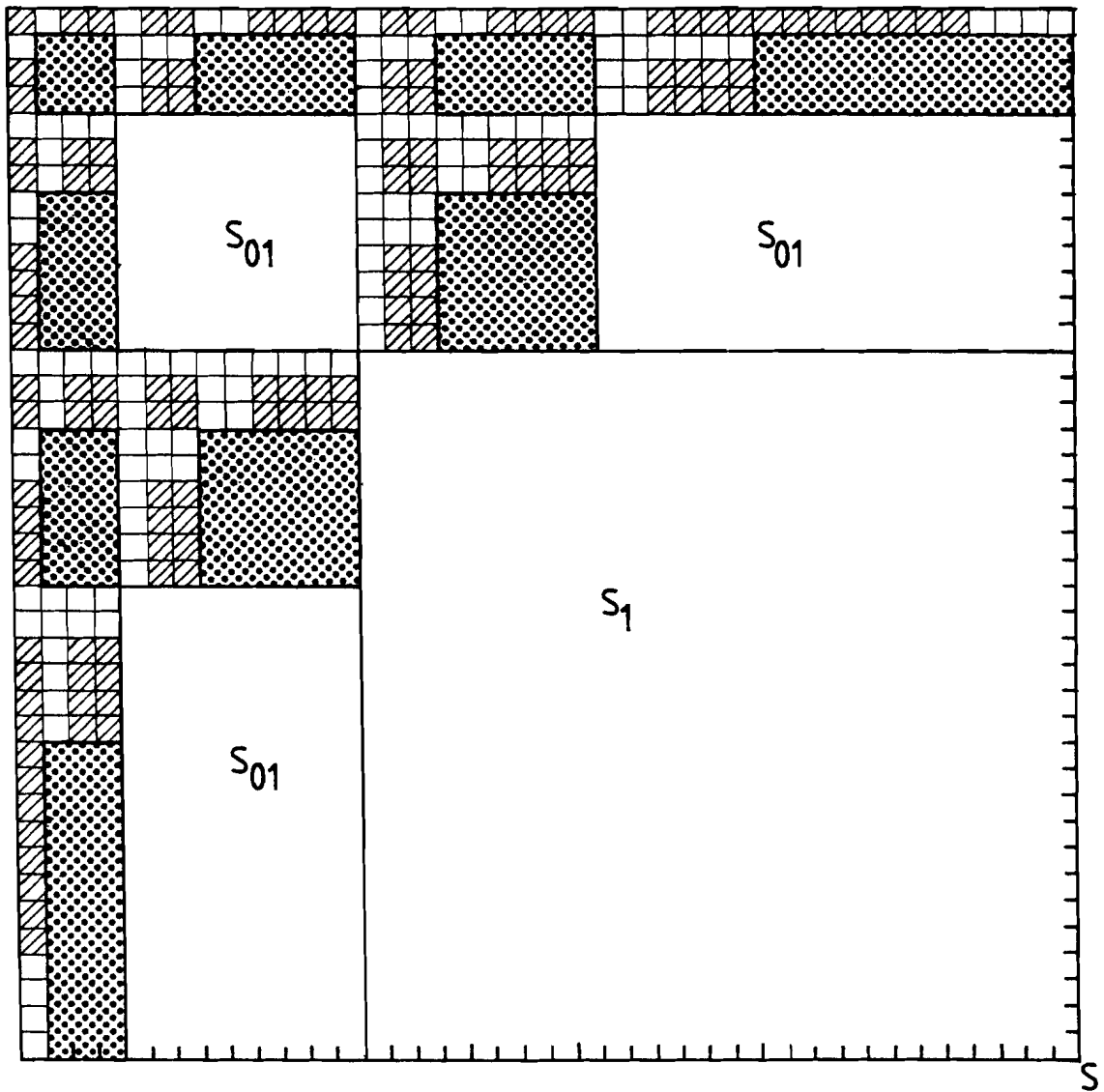
$S_{000100} = S_{000101}$






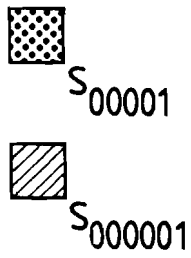
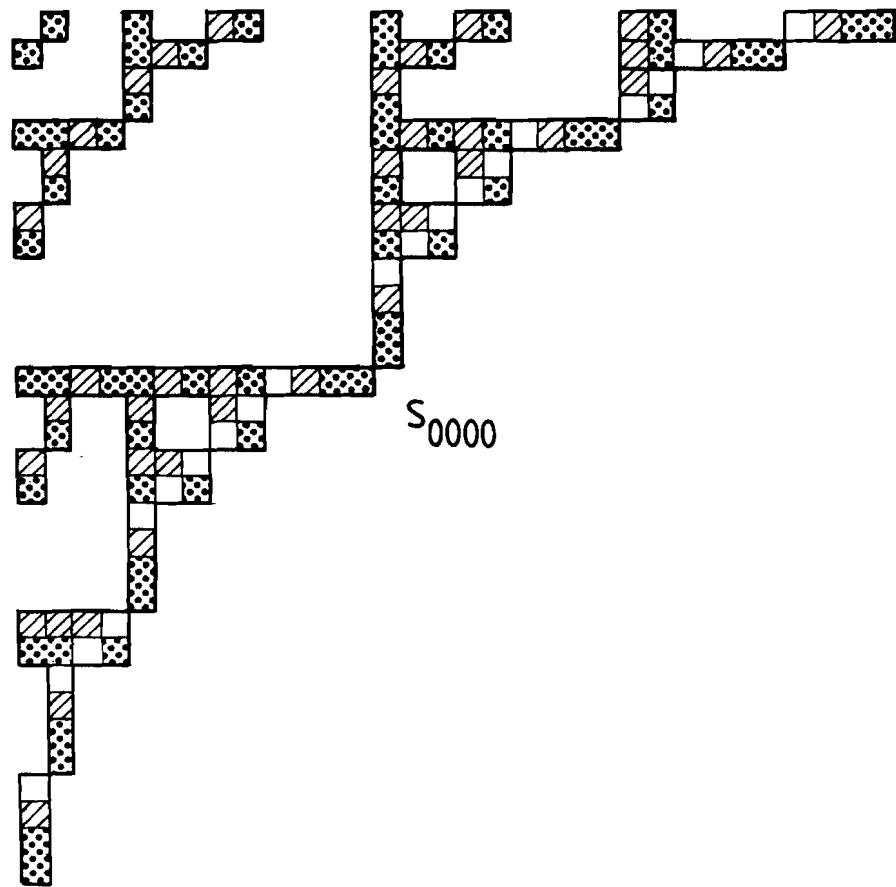


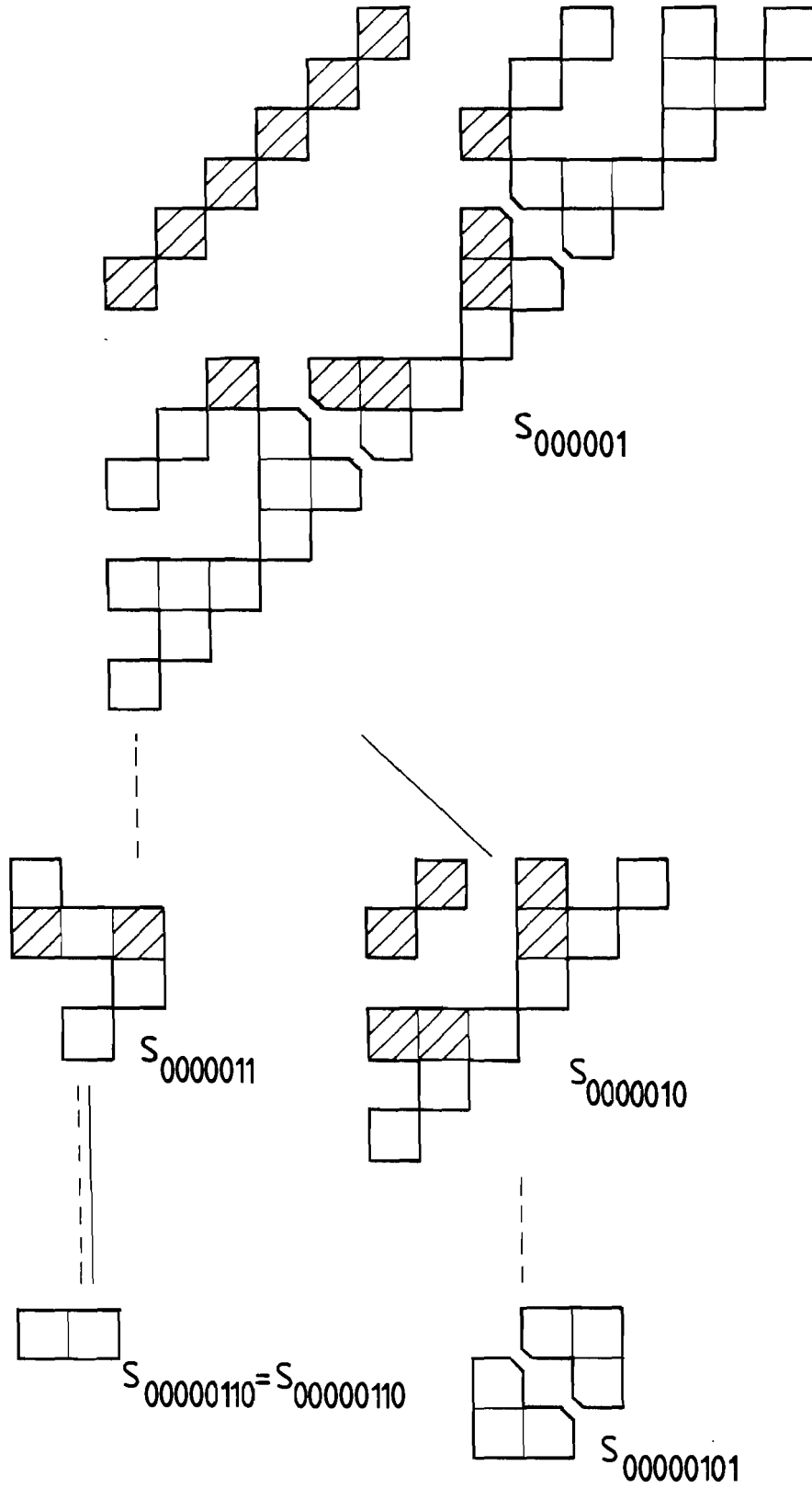


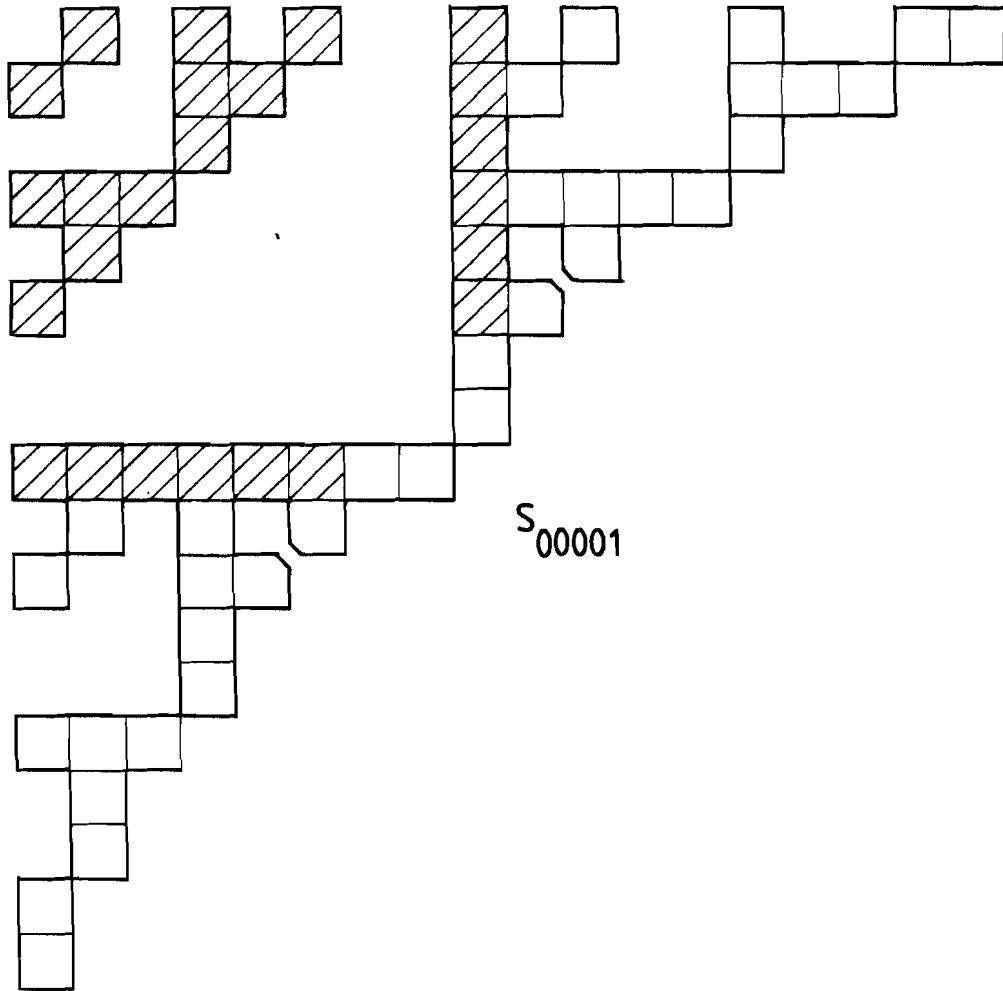


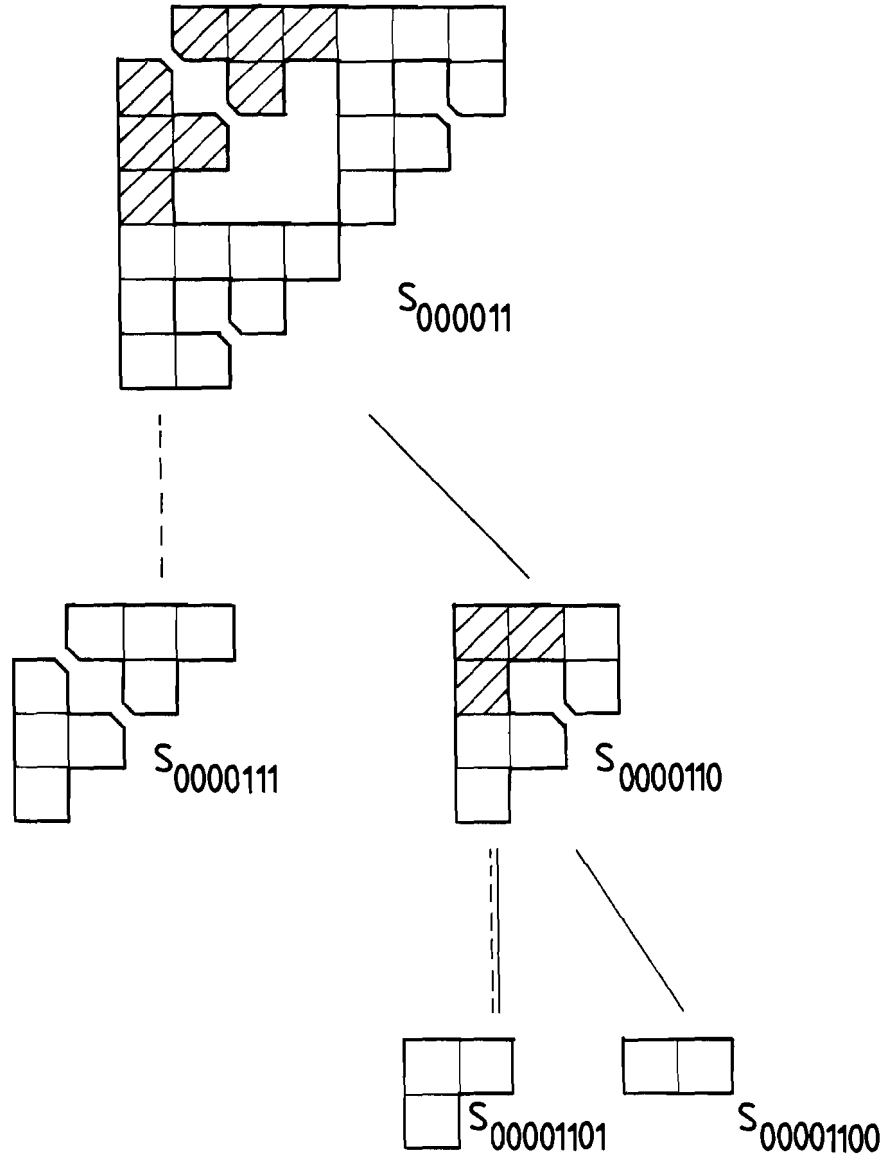
 S_{001}

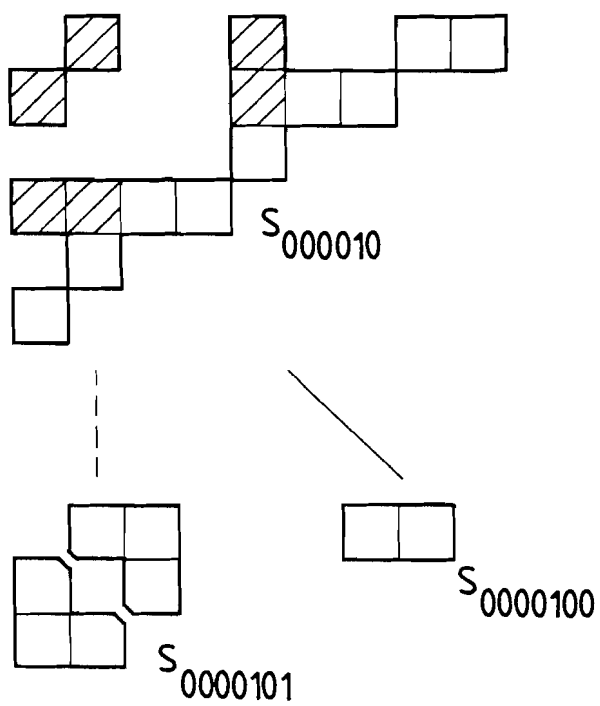
 S_{0001}

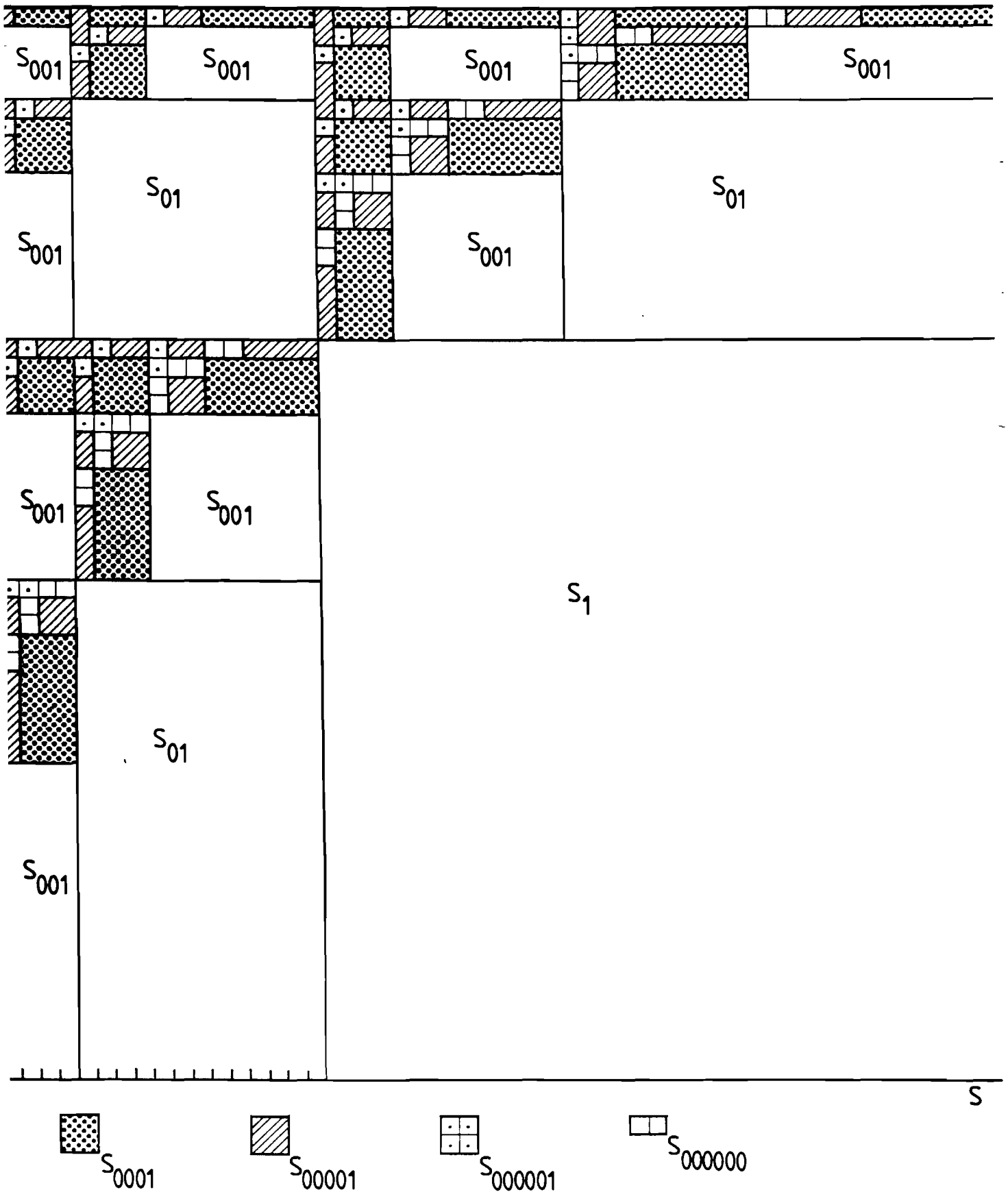


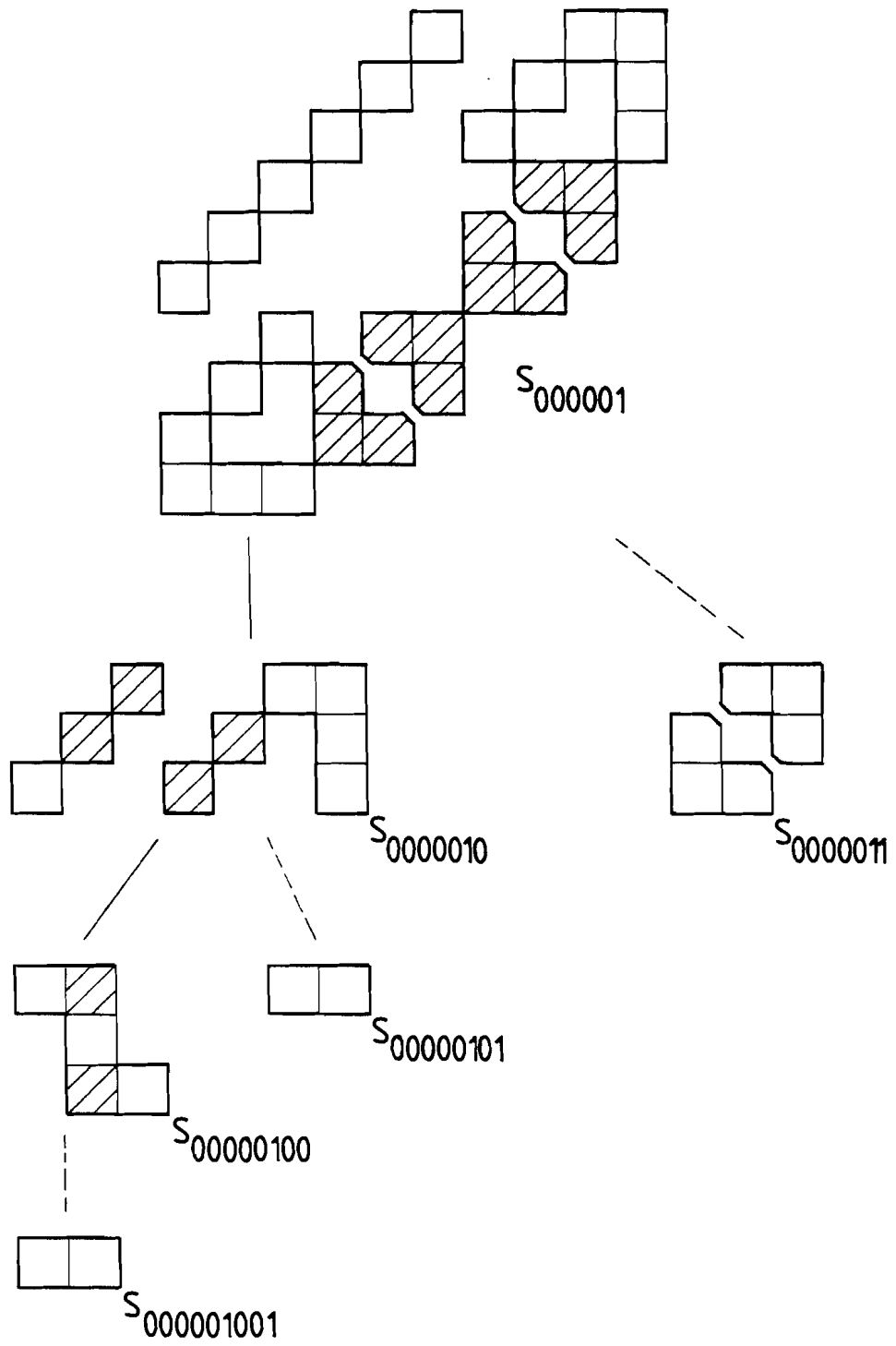












B. HET INTERACTIEVE PROGRAMMA

De tweede appendix van het afstudeerverslag handelt over het interactieve programma waarmee oplossingen gezocht kunnen worden voor de opdeling van de vierkanten. De indeling van de hoofdstukken is als volgt:

- Een algemene beschrijving van het programma
- Speciale opties
- De gedetailleerde beschrijving van het programma
- De beschrijving van de belangrijkste subroutines
- Aanbevelingen voor verbeteringen van het programma

B.1 Een algemene beschrijving van het programma

De opzet was, om een gebruikersvriendelijk programma te schrijven. Dit houdt in, een duidelijke en overzichtelijke interactieve communicatie met de gebruiker. Dit betekent een snelle afwikkeling van oplossingen die bekend zijn, en een duidelijke weergave van de stappen en tussenoplossingen. Dit is als volgt gerealiseerd. Het programma start met het onopgeloste probleem. Het tekent hiervoor het volledige door de gebruiker gekozen, op te lossen vierkant. Het programma bevindt zich nu in de overzichts-fase. Door zijn interactiviteit, is het programma in staat te communiceren met de gebruiker. Het zal dit in deze fase doen door het stellen van de volgende vragen:

1. Which array do you want to solve?
2. Do you know a good solution?
3. Give the number please

Met de eerste vraag kiest u het vierkant of subgebied wat U wilt oplossen. In de beginsituatie kunt U alleen "1" kiezen, omdat er maar 1 figuur aanwezig is. Met de tweede informeert het programma of U een oplossing voor het gebied kent. De praktijk heeft geleerd dat veel (sub-)gebieden regelmatig terugkeren. Een goede realiseerbare oplossing voor zo'n gebied is dan ook bekend bij de gebruiker, en hij kan dit duidelijk maken door op de tweede vraag "y(es)" te antwoorden. Het

programma zal hierop reageren door U te vragen hoeveel transmissies U denkt nodig te hebben om het (sub-)gebied op te lossen. Het programma controleert dit getal niet. Bij een volgende terugkeer in de overzichts-fase zal het programma dit (sub-)vierkant niet meer tekenen. Bij een ontkennend antwoord van vraag 2, ("n(o)") zal het programma de figuur opnieuw tekenen, en U vragen de strategie aan te geven, die U in gedachten heeft. Het programma bevindt zich nu in de strategie-fase. De uitgangssituatie is, dat het programma voor alle berichten van user1 en user2 een "0" op het scherm zet. U kunt deze "0" in een "1" veranderen door de cursor, die inmiddels op het scherm is verschenen, op een bepaalde "0" te plaatsen en de toets "1" in te drukken. Het programma zal de strategie overschrijven, en U om de volgende opdracht vragen. Op dezelfde manier kunt U ook een "1" in een "0" veranderen. Omdat we met een geheugenscherm te maken hebben, kan hierdoor de situatie onoverzichtelijk worden. Daarom is een speciale optie ingebouwd, die de gehele figuur opnieuw tekent, met de dan geldende strategie. Dit kunt U doen door de toets " " in te drukken. De plaats van de cursor is dan onbelangrijk. Het programma blijft door deze handeling in de strategie-fase. Is de strategie die U wenst volledig ingetikt, dan kunt U het programma zijn subgebieden laten uitrekenen, en een overzicht laten geven van de nog op te lossen (sub-)gebieden door de toets "esc" in te drukken. Een (sub-)gebied dat slechts bestaat uit 1 enkel hok (zijnde 1 message pair ($m_{1,r}, m_{2,s}$)) beschouwd het programma als opgelost en wordt niet meer getekend als deelgebied in de overzichts-fase. Wanneer all (sub-)gebieden opgelost zijn, zal het programma het totale aantal transmissies, nodig voor de zojuist ingetikt totale strategie, meedelen op het scherm. Daaronder verschijnt een tabel, waarin de totale, door U gevolgde strategie, staat weergegeven. U kunt hiervan een copie maken.

B.2 Speciale opties

Het programma is in staat in de overzicht-fase, 72 (sub-)gebieden te tekenen. Dit is gerealiseerd door een paginering in te voeren. Elke

pagina bevat 12 tekeningen, en indien het aantal onopgeloste (sub-)gebieden groter of gelijk wordt aan 13, zal in de overzichts-fase automatisch de vraag:

Next page?

op het scherm verschijnen. Door het intikken van "y(es)" of "n(o)" kunt U al dan niet door de pagina's heen stappen en het juiste (sub-)gebied van U keuze uitzoeken. Bij de laatste pagina verschijnt automatisch de vraag:

Previous page?

Beantwoording van deze vraag spreekt voor zichzelf!

Het programma is in staat om de beste oplossing bij te houden. Dit betekent dat het programma automatisch een strategie zal afbreken, wanneer het aantal transmissies van de huidige strategie groter wordt dan het aantal waarvoor geldt, dat het correspondeert met de tot nu toe beste oplossing. Deze optie is ingebouwd, om U niet onnodig lang door te laten zoeken. Maar er schuilt echter wel een gevaar in deze optie. Zoals vermeld, bestaat de mogelijkheid om in de overzichts-fase, (sub-)gebieden direct op te lossen, door het invullen van een getal, dat overeenkomt met het aantal transmissies dat nodig is om deze figuur op te lossen. Dit getal wordt echter niet gecontroleerd door het programma. Hierdoor kan het totale aantal transmissies voor een bepaalde strategie dus op een te lage waarde worden gezet. De gevolgen hiervan zijn funest. Ik raad U dan ook aan, de strategie nogmaals na te rekenen, al dan niet met behulp van het programma.

Als het programma opgestart wordt, zal het in eerste instantie het aantal transmissies van de beste oplossing met bijbehorende strategie opvragen. De vragen die hiervoor gesteld worden, spreken voor zichzelf. Door het invoeren van het aantal transmissies voor de beste oplossing wordt de variabele, die het zoekproces voortijdig moet afbreken, op een bepaalde waarde gezet. Deze waarde is dus weer erg belangrijk. De invoering van de strategie kan in principe vroegtijdig afgebroken worden, daar dit geen wezenlijke informatie voor het goed draaien van het programma oplevert. Dit kan gedaan worden door de vraag:

More?

met "n(o)" te beantwoorden. Hierna zal het programma nog eenmaal de door U ingevoerde gegevens op het scherm plaatsen en U vragen of U het hier mee eens bent. Een bevestigend antwoord, zal dan het zoekprogramma opstarten.

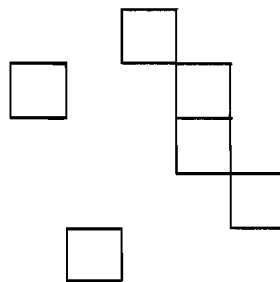
B.3 De gedetailleerde beschrijving van het programma.

In dit hoofdstuk wordt ingegaan op de numerieke verwerking van de (sub-)gebieden door het programma. We zullen dit bespreken aan de hand het programma dat oplossingen moet zoeken voor het 10×10 array. Elke figuur, die voorkomt tijdens het zoeken naar een goede strategie, correspondeert met een 10×10 array wat als volgt is opgebouwd. Stel message pair $(m_{1,r}, m_{2,s})$ komt voor in een figuur. Dan zal in het 10×10 array op de plaats (r,s) een "1" staan. Als message pair $(m_{1,p}, m_{2,q})$ niet voorkomt in de gegeven figuur, zal er op plaats (p,q) in het 10×10 array een "0" staan. Zo correspondeert elk array, met een bepaald figuur. Door gebruik te maken van deze vertaling van arrays naar figuren kan snel een aantal controlerende routines uitgevoerd worden. Bijvoorbeeld: Indien de som van alle plaatsen van het array wordt uitgerekend, weet het programma of het met een "singleton" te maken heeft of niet. Immers als de som gelijk is aan "1", hebben we te maken met een enkele message pair en kunnen we dit beschouwen als opgelost. Ook het S_1 -gebied en het S_0 -gebied kunnen snel berekend worden. Namelijk het S_1 -gebied wordt berekend, door de strategie voor user1 en user2, die door de gebruiker zijn ingevoerd, als volgt te behandelen. Beschouw de user1 strategie als een horizontale vector, en andere als een verticale. Door deze twee vectoren met elkaar te vermenigvuldigen, ontstaat er een matrix. Deze matrix is het S_1 -gebied. Het S_0 -gebied is dan gelijk aan het $(S-S_1)$ -gebied. Tot slot, als de som van alle plaatsen van het array "0" is, dan betekent dit dat in dit array geen figuur zit opgeslagen, en dus gebruikt kan worden om een (sub-)gebied weer te geven. In het voorgaande is verteld, dat we 72 plaatjes kunnen weergeven, daarom is er dan ook een $10 \times 10 \times 72$ array gedefinieerd in het

programma. Het array "ADM" ter lengte 72 houdt bij welke posities in de derde dimensie van het $10 \times 10 \times 72$ array gebruikt zijn. Dit is gedaan om niet telkens de som van de plaatsen te hoeven tellen, bij het zoeken naar een leeg 10×10 array.

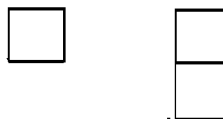
B.4 De beschrijving van de belangrijkste subroutines

U begrijpt dat de belangrijkste subroutine die is, die de "ambiguity classes" van het S_0 -gebied en het S_1 -gebied moet berekenen. Deze subroutine heet "locate" en zal in deze paragraaf beschreven worden. In hoofdstuk 2 is beschreven hoe we de onderlinge samenhang kunnen controleren van die message pairs, die samen in een (sub-)gebied zitten. Namelijk: voor al deze message pairs moet gelden, dat er een ketting te definiëren moet zijn, die alle message pairs bevat, die in zo'n (sub-)gebied zitten. Zo'n ketting mag alleen bestaan uit horizontale en verticale schakels tussen de message pairs. Een message pair, wat dus wel deel uitmaakt van een S_i -gebied ($i=0,1$), maar niet tot de ketting κ behoort, behoort ook niet tot het (sub-)gebied κ . Deze relatie laat zich gemakkelijk vertalen in het programma, omdat we uitgegaan zijn van arrays. Stel we hebben een 5×5 array dat correspondeert met figuur B.4.1.



figuur B.4.1.

Het is duidelijk dat de oplossing van deze figuur als volgt is:



figuur B.4.1.

Hoe vinden we deze nu? We maken hierbij gebruik van 2 referentie vectoren R1 en R2, beide ter lengte 5 en met als begin waarde (0,0,0,0,0). Verder noemen we het array waarin de oorspronkelijke figuur zit opgeslagen A1. De andere benodigde arrays noemen we A2, A3 enz. In eerste instantie kopiëren we de meest rechtse kolom van A1 in R1 en kijken of de som van de plaatsen van de vector groter of gelijk is aan "1". Als dat zo is, wordt de eerste kolom van A1 in de eerste kolom van A2 gecopieerd. Als de som van de plaatsen "0" is, onderzoeken we de tweede kolom enz enz, net zo lang totdat we een vector gevonden hebben waarvan de som van de elementen wel groter of gelijk is aan "1". We bereiken dus altijd een punt, waar 1 kolom van A1 gecopieerd is in A2. Dit punt noemen we "restart search". Vanaf dit punt gaan we de kettingen zoeken. Dit gebeurt als volgt. De eerst volgende linkse kolom van A1 wordt gecopieerd in R2, doch slechts op die plaatsen, waar R1 een "1" heeft staan. Is de som van de elementen van R2 groter of gelijk aan "1", dan wordt ook deze kolom gecopieerd in A2. Is de som "0", dan wordt verder geen actie ondernomen en de volgende linkse kolom bekeken. Nu moeten we er alleen nog voor zorgen dat na elke kopiëring van een kolom van A1 in A2, de vector R1 aangepast wordt. Dit gebeurt, door na zo'n kopiëring de vereniging van R1 en R2 in de vector R1 te plaatsen. Deze handelingen herhalen we tot en met de laatste kolom. Daarna springen we naar "restart search" en herhalen de actie een aantal malen. Wat is nu het stop criterium? Dit is als volgt gedefinieerd:

Nadat we de laatste kolom onderzocht hebben, en dus willen terug springen naar "restart search", wordt steeds de som van de elementen van R1 bepaald. Verandert deze niet meer, dan betekent dit, er geen nieuwe elementen meer toegevoegd zijn aan A2. We hebben dus een volledige (sub-)gebied bepaald. Dit (sub-)gebied elimineren we uit A1, en wordt gedefiniëerd als een nieuw te onderzoeken (sub-)gebied (mits de som van de elementen van het array groter of gelijk is aan "2"). Het resterende gebied dat overeenkomt met A1-A2, wordt gecontroleerd op zijn relevantie (controleer de som). Als dat zo is wordt "locate" opnieuw opgestart, maar dan op A1-A2 enz enz.

B.5 Aanbevelingen voor verbeteringen van het programma

De opzet van dit programma was, om een programma te maken, om snel en foutloos te zoeken naar de oplossingen van een bepaald gebied, om hiermee meer inzicht te krijgen in de oplossingen die optimaal zijn voor de opdelingen van een gebied. De interactiviteit was hierbij een voorwaarde omdat een automatisch zoekend programma al snel enorme hoeveelheden mogelijke oplossingen zou moeten testen, wat enorm veel rekentijd met zich mee zou gaan brengen. Dit laatste is onvermijdelijk, omdat we momenteel nog niet over voldoende afschattende parameters beschikken, waarmee bepaalde opdelingen als onzinnig zouden kunnen worden geclassificeerd. Maar laten we eens kijken naar de strategieën, die we momenteel hanteren. Hierbij liggen een aantal stappen vast, en op het laatste moment schakelen we over op "hand besturing". Op dit moment zou een automatische computer-search wel zinvol zijn, om in ieder geval de optimale opdeling van het restgebied te vinden.

Verder zou de beste opdeling van de tot nu toe gevonden gebieden (d.m.v. het onderzoek van Hollmann, Post en Ligtenberg) opgeslagen kunnen worden in een file. Bij het herkennen van zo'n gebied door het programma tijdens een "run", zou het programma zelf de juiste waarde in kunnen vullen voor het aantal transmissies voor zo'n gebied.

De programma's die door mij geschreven zijn, zijn afhankelijk van een aantal parameters, die gedefiniëerd worden, door de grote van het te onderzoeken gebied. Hierdoor heb ik aparte run-versies van het 3x3, 4x4,, 10x10 programma gemaakt. Een derde aanbeveling is dan ook, om het programma dusdanig te wijzigen, dat door het ingeven van M_1 en M_2 , automatisch een nieuwe run-versie wordt gemaakt, waarna het programma gaat draaien. Dit betekent, dat met het ingeven van enkele parameters, elk willekeurig gebied kan worden onderzocht.

Ten vierde zou onderzocht kunnen worden of het veranderen van de strategie (het aanbrengen van de nullen en enen in de strategie-fase) niet sneller kan gebeuren. Momenteel overschrijft het programma alle nullen en enen, terwijl er slechts 1 verandert. Voor het onderzoek naar grote opdelingen, is het zinvol, dat alleen die 1 of 0 overschreven wordt, die ik wil veranderen.

Tenslotte kan onderzocht worden, of de "grafical" niet bestuurd kan worden vanaf een andere terminal. Hiermee zouden dan alleen de relevante plaatjes op de graphical komen, terwijl de tekst op een gewoon beeldscherm verschijnt.