J. Symbolic Computation (1995) **20**, 471–482

Reduced Idempotents in the Semigroup of Boolean Matrices †

JANUSZ KONIECZNY‡

Department of Mathematics, Mary Washington College, 1301 College Avenue, Fredericksburg, VA 22401-5358, USA

(Received 3 April 1995)

We present an algorithm that generates all reduced idempotents in the semigroup of $n \times n$ Boolean matrices. As a consequence, we obtain a method of listing all partial order relations on a finite set with n elements.

1. Introduction

The semigroup $\mathcal{B}_n$ of Boolean matrices is the set of $n \times n$ matrices over $\{0, 1\}$ with the usual matrix multiplication except that we assume $1 + 1 = 1$. It is isomorphic (in a natural way) to the semigroup $\mathcal{B}_X$ of binary relations on a set X with n elements, where the operation is the composition of relations (see Lallement (1979), p. 4).

In contrast with the semigroups of (full or partial) transformations on X, $\mathcal{B}_n$ is not regular when $n \geq 3$ (see Plemmons and West (1970)). The regular elements of $\mathcal{B}_n$ were characterized by Zaretskii (1962), Schein (1976), and Kim and Roush (1978). A description of the idempotents in $\mathcal{B}_n$ was obtained by Chaudhuri and Mukherjea (1980). Earlier, Plemmons and West (1970) characterized the reduced idempotents in $\mathcal{B}_n$. Their result (see Plemmons and West (1970), Theorem 2.4) states that the set of reduced idempotents in $\mathcal{B}_n$ (viewed as binary relations on a set X with n elements) coincides with the set of partial order relations on subsets of X. For a survey of the theory and applications of Boolean matrices, see Kim (1982).

The purpose of this paper is to present an algorithm for generating all reduced idempotents in $\mathcal{B}_n$.

First, we observe that the set of reduced idempotents is in one-to-one correspondence with the set of matrices that are reduced, regular, and sorted. (A Boolean matrix is sorted if its nonzero rows, viewed as binary numbers, form a nondecreasing sequence and are placed before any zero row.) Next, we generate all reduced, regular, and sorted matrices

† This research was supported by a Mary Washington College Faculty Development Grant.
‡ E-mail: jkoniecz@mwc.edu

0747–7171/95/100471 + 12 $12.00/0

by constructing successively lists $L_0, L_1, \dots, L_n$ of all such matrices of rank $0, 1, \dots, n$, respectively (Algorithm 4.2).

The list L_{r+1} is obtained from L_r. For each α in L_r, we construct the set of all matrices from L_{r+1} that differ from α only at the row $r+1$. (Such matrices are called the extensions of α.) This (essential) part of the construction is based on a characterization of the extensions of a reduced, regular, and sorted Boolean matrix of rank $r < n$ (Theorem 4.1). The proof of the characterization is based on a criterion for a reduced sorted matrix to be regular (Theorem 3.1).

To generate all reduced idempotents in $\mathcal{B}_n$, we construct lists $E_0, E_1, \dots, E_n$ of all such idempotents of rank $0, 1, \dots, n$, respectively (Algorithm 4.3). The list E_r is obtained from L_r by permuting rows of each α in L_r. We show how to obtain a permutation that leads from α to the corresponding reduced idempotent ε (Algorithm 3.2).

The list E_n consists of all reduced idempotents of rank n. It follows from Plemmons and West (1970), Theorem 2.4, that such idempotents (viewed as binary relations on a set X with n elements) coincide with partial order relations on X. Therefore, E_n is a list of all partial order relations on a set with n elements.

Obtaining a formula for the number P_n of all partial order relations on a set with n elements is a long-standing open problem. Some asymptotic formulas for P_n have been found, for example by Kleitman and Rothschild (1975). On the other hand, computers have been used to calculate P_n for small n. Erné and Stege (1991) give values of P_n for $n \leq 14$. Their method is based on an algorithm developed by Culberson and Rawlins (1991) for generating unlabeled posets (isomorphic classes of partial order relations) and on some recursive formulas. Since the value of P_n for $n = 14$ is greater than 10^{22}, it seems that an algorithm based solely on generating all partial order relations cannot be used to compute higher values of P_n. For a comprehensive list of references concerning enumeration of partial order relations on a finite set, see Erné and Stege (1991).

2. Standard Definitions and Results

An equivalence relation $\mathcal{L}$ ($\mathcal{R}$) on a semigroup S is defined by the rule that $a\mathcal{L}b$ ($a\mathcal{R}b$) if and only if a and b generate the same principal left (right) ideal in S, i.e., if and only if $S^1a = S^1b$ ($aS^1 = bS^1$), where S^1 is the semigroup S with an identity adjoined. The relations $\mathcal{L}$ and $\mathcal{R}$ commute, and the composition $\mathcal{L} \circ \mathcal{R} = \mathcal{R} \circ \mathcal{L}$ is called $\mathcal{D}$. These equivalence relations are called **Green's equivalences** (see Howie (1976), p. 38). For $a \in S$, we denote the equivalence classes of a with respect to $\mathcal{L}$, $\mathcal{R}$, and $\mathcal{D}$ by L_a, R_a, and D_a, respectively.

An element a of a semigroup S is called **regular** if $a = axa$ for some $x \in S$. If a $\mathcal{D}$-class D contains a regular element, then all elements of D are regular, and D is called a **regular $\mathcal{D}$-class**. In a regular $\mathcal{D}$-class, every $\mathcal{L}$-class and every $\mathcal{R}$-class contains an idempotent (an element e with $e^2 = e$).

Turning to the semigroup $\mathcal{B}_n$ of Boolean matrices, we observe that the rows and columns of any $\alpha \in \mathcal{B}_n$ come from the set V_n of all n-tuples over $\{0, 1\}$. Elements of V_n, called **Boolean vectors**, can be added coordinate-wise using Boolean addition ($1 + 1 = 1$). The zero vector will be denoted by $\mathbf{0}$, and the ith coordinate of $v \in V_n$ by v_i ($1 \leq i \leq n$). For $u, v \in V_n$, we write $u \leq v$ if u is less than or equal to v when u and v are viewed as binary numbers. We say that u is **included** in v, and write $u \sqsubseteq v$, if $u_i = 1$ implies $v_i = 1$ ($1 \leq i \leq n$). For example, if $u = 0011$, $v = 0110$, and $w = 1011$ are vectors in V_4, then $u \leq v \leq w$ and $u \sqsubseteq w$. Finally, for $i \in \{1, \ldots, n\}$, we denote by e_i the vector in V_n whose ith coordinate is 1 and any other coordinate is 0.

For $\alpha \in \mathcal{B}_n$, the **row space** of α, denoted by $V(\alpha)$, is the subset of V_n consisting of the zero vector $\mathbf{0}$ and all possible sums of (one or more) nonzero rows of α. The **row basis** of α, denoted by $r(\alpha)$, is the set of all nonzero vectors in $V(\alpha)$ that are not sums of other vectors in $V(\alpha)$. Note that each vector in $r(\alpha)$ must be a row of α. The cardinality of $r(\alpha)$ is called the **row rank** of α. Note that the zero matrix has the empty set as its row basis, and is of rank 0.

For example, for

$$\alpha = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

$V(\alpha) = \{0000, 1010, 0101, 1100, 1110, 1101, 1111\}$, $r(\alpha) = \{1010, 0101, 1100\}$, and row rank of $\alpha = 3$.

We have the dual notions of the **column space** $W(\alpha)$, **column basis** $c(\alpha)$, and **column rank** of α. If α is regular, then the row and column ranks of α are the same, and this common value is called the **rank** of α.

For $\alpha \in \mathcal{B}_n$ and indices i, j (by an index we mean an element of the set $\{1, \ldots, n\}$), the (i, j)-entry of α is denoted by α_{ij}, and the ith row (column) of α by α_{i*} (α_{*i}). The sets of indices $\mathbf{d}\alpha = \{i : \alpha_{i*} \neq \mathbf{0}\}$ and $\mathbf{r}\alpha = \{i : \alpha_{*i} \neq \mathbf{0}\}$ are called, respectively, the **domain** and **range** of α.

A matrix $\alpha \in \mathcal{B}_n$ is called **row reduced (column reduced)** if no nonzero row (column) of α is a sum of (one or more) other rows (columns) of α; α is **reduced** if it is both row and column reduced. Note that the row rank of a reduced α is the number of nonzero rows of α.

A matrix $\alpha \in \mathcal{B}_n$ is called **sorted** if there is k, $0 \leq k \leq n$, such that

$$\mathbf{0} \neq \alpha_{1*} \leq \alpha_{2*} \leq \ldots \leq \alpha_{k*} \quad \text{and} \quad \alpha_{k+1,*} = \alpha_{k+2,*} = \ldots = \alpha_{n*} = \mathbf{0}.$$

For example, the matrix

$$\alpha = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

is row (but not column) reduced and sorted.

The following two lemmas are combinations of results proved by Plemmons and West (1970) (Lemma 1.2, Lemma 1.7, Theorem 1.9, and Theorem 2.4). Result (1) in Lemma 2.1 was first proved by Zaretskii (1963). The symbol S_n denotes the symmetric group of permutations on the set $\{1, 2, \ldots, n\}$.

LEMMA 2.1. *For any matrices $\alpha, \beta \in \mathcal{B}_n$:*

(1) *α and β are in the same $\mathcal{L}$ ($\mathcal{R}$)-class if and only if they have the same row (column) space.*

(2) *If α and β are row reduced and $\alpha \mathcal{L} \beta$, then there is a permutation $p \in S_n$ such that*

$$\alpha_{p(i),*} = \beta_{i*} \quad (1 \leq i \leq n).$$

(3) *If* α *is row (column) reduced, then every member of* R_α (L_α) *is row (column) reduced.*
(4) *If* α *is reduced and regular, then* L_α *contains exactly one reduced idempotent.* □

LEMMA 2.2. *For any idempotent* $\varepsilon \in \mathcal{B}_n$*:*

(1) *If* $\varepsilon_{ij} = 1$, *then* $\varepsilon_{j*} \sqsubseteq \varepsilon_{i*}$ $(1 \le i, j \le n)$.
(2) *If* ε *is reduced and* $\varepsilon_{i*} \ne \mathbf{0}$, *then* $\varepsilon_{ii} = 1$ $(1 \le i \le n)$.
(3) *If* ε *is reduced, then* $\mathbf{d}\varepsilon = \mathbf{r}\varepsilon$. □

We shall also need the following result.

LEMMA 2.3. *Assume that* $\alpha \in \mathcal{B}_n$ *is reduced and regular,* $\varepsilon \in \mathcal{B}_n$ *is a reduced idempotent in* L_α, *and* $p \in S_n$ *is a permutation such that*

$$\varepsilon_{p(i),*} = \alpha_{i*} \quad (1 \le i \le n).$$

Then for all indices u *and* v*:*

$$\varepsilon_{p(u),p(v)} = 1 \quad \textit{implies} \quad \alpha_{v*} \sqsubseteq \alpha_{u*}.$$

PROOF. By (1) of Lemma 2.2, $\varepsilon_{p(u),p(v)} = 1$ implies $\alpha_{v*} = \varepsilon_{p(v),*} \sqsubseteq \varepsilon_{p(u),*} = \alpha_{u*}$. □

3. A Criterion for Regularity

This section presents a criterion for a reduced sorted Boolean matrix to be regular (Theorem 3.1), and, as an application of the criterion, an algorithm for constructing a reduced idempotent in the $\mathcal{D}$-class of a regular $\alpha \in \mathcal{B}_n$.

Let $\alpha \in \mathcal{B}_n$ and let i, j be indices such that $\alpha_{ij} = 1$. The smallest index h such that $\alpha_{hj} = 1$ will be denoted by m_{ij}^α. Note that $m_{ij}^\alpha \le i$ and $m_{ij}^\alpha = i$ means that the column α_{*j} has only 0s above the 1 in the entry (i, j). For example, if

$$\alpha = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

then $m_{31}^\alpha = 1$, $m_{34}^\alpha = 3$, and $m_{43}^\alpha = 2$.

The following theorem states that a reduced and sorted $\alpha \in \mathcal{B}_n$ is regular if and only if each nonzero row α_{i*} contains exactly one 1 with only 0s above and for any other 1 in α_{i*}, the highest row with 1 in the same column is included in α_{i*}. The words "above" and "highest" refer to visual directions when α is written down on paper.

THEOREM 3.1. *Let* $\alpha \in \mathcal{B}_n$ *be reduced and sorted. Then* α *is regular if and only if the following conditions are satisfied:*

(1) *For each index* i *with* $\alpha_{i*} \ne \mathbf{0}$, *there is exactly one index* z *such that* $\alpha_{iz} = 1$ *and* $m_{iz}^\alpha = i$;
(2) *For all indices* i, j*:*

$$\alpha_{ij} = 1 \textit{ and } h = m_{ij}^\alpha \quad \textit{imply} \quad \alpha_{h*} \sqsubseteq \alpha_{i*}.$$

PROOF. Suppose α is regular. By (4) of Lemma 2.1, L_α contains a reduced idempotent ε. Further, by (2) of Lemma 2.1, there is a permutation $p \in S_n$ such that

$$\varepsilon_{p(i),*} = \alpha_{i*} \quad (1 \le i \le n).$$

To prove (1), assume $\alpha_{i*} \ne \mathbf{0}$, and set $z = p(i)$. Then $\varepsilon_{z*} = \alpha_{i*} \ne \mathbf{0}$, which gives $\alpha_{iz} = \varepsilon_{zz} = 1$ by (2) of Lemma 2.2. Setting $h = m^\alpha_{iz}$, we have $\varepsilon_{p(h),p(i)} = \alpha_{hz} = 1$, which gives $\alpha_{i*} \sqsubseteq \alpha_{h*}$ by Lemma 2.3. Since α is reduced and sorted, $\alpha_{i*} \sqsubseteq \alpha_{h*}$ implies $i \le h$. Since $h = m^\alpha_{iz} \le i$, it follows that $\alpha_{iz} = 1$ and $m^\alpha_{iz} = i$.

Turning to uniqueness, suppose t is another index with $\alpha_{it} = 1$ and $m^\alpha_{it} = i$. Set $s = p^{-1}(t)$ and observe that it suffices to show that $i = s$ (since then $z = p(i) = p(s) = t$). The fact that $\varepsilon_{p(i),p(s)} = \varepsilon_{zt} = \alpha_{it} = 1$ and Lemma 2.3 give $\alpha_{s*} \sqsubseteq \alpha_{i*}$, which implies $s \le i$. On the other hand, $\varepsilon_{zt} = \alpha_{it} = 1$ shows that $t \in \mathrm{r}\varepsilon = \mathbf{d}\varepsilon$, and so (2) of Lemma 2.2 gives $\alpha_{st} = \varepsilon_{tt} = 1$. By the definition of m^α_{it}, $\alpha_{st} = 1$ implies $i = m^\alpha_{it} \le s$, which concludes the proof of (1).

To prove (2), assume $\alpha_{ij} = 1$ and $h = m^\alpha_{ij}$. By the definition of m^α_{ij}, $\alpha_{hj} = 1$ and $m^\alpha_{hj} = h$. Consequently, $j = p(h)$ by the proof of (1). Now, $\varepsilon_{p(i),p(h)} = \alpha_{ij} = 1$, implying $\alpha_{h*} \sqsubseteq \alpha_{i*}$ by Lemma 2.3.

Conversely, suppose (1) and (2) are satisfied. If α is the zero matrix, we're done. Otherwise, it suffices to construct an idempotent ε in L_α. Since α is sorted, there is k $(1 \le k \le n)$ such that $\alpha_{i*} \ne \mathbf{0}$ for each $i \le k$ and $\alpha_{i*} = \mathbf{0}$ for each $i > k$. By (1), for each $i \le k$, there is a unique index z_i such that $\alpha_{iz_i} = 1$ and $m^\alpha_{iz_i} = i$. Note that $z_1, \dots, z_k$ are distinct, and let $z_{k+1} < \dots < z_n$ be the remaining indices. Define a permutation $p \in S_n$ by

$$p(i) = z_i \quad (1 \le i \le n),$$

and a Boolean matrix $\varepsilon \in \mathcal{B}_n$ by

$$\varepsilon_{p(i),*} = \alpha_{i*} \quad (1 \le i \le n).$$

By (1) of Lemma 2.1, $\varepsilon \in L_\alpha$. It remains to show $\varepsilon^2 = \varepsilon$. If $i > k$, then $\varepsilon_{p(i),*} = \alpha_{i*} = \mathbf{0}$, implying $\varepsilon^2_{p(i),*} = \mathbf{0}$. Assume $i \le k$. First, $\varepsilon_{p(i),p(i)} = \alpha_{i,p(i)} = \alpha_{iz_i} = 1$ gives $\varepsilon_{p(i),*} \sqsubseteq \varepsilon^2_{p(i),*}$. For the reverse inclusion, assume that j is an index with $\varepsilon_{p(i),j} = \alpha_{ij} = 1$, and set $h = m^\alpha_{ij}$. Apply the definition of m^α_{ij} to get $\alpha_{hj} = 1$ and $m^\alpha_{hj} = h$, and then the definition of p to get $p(h) = z_h = j$. Finally, by (2), $\varepsilon_{j*} = \varepsilon_{p(h),*} = \alpha_{h*} \sqsubseteq \alpha_{i*} = \varepsilon_{p(i),*}$, which shows $\varepsilon^2_{p(i),*} \sqsubseteq \varepsilon_{p(i),*}$ and concludes the proof. □

To illustrate, for the reduced and sorted

$$\alpha = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \in \mathcal{B}_3,$$

(1) of Theorem 3.1 is satisfied, but (2) is not ($\alpha_{31} = 1$ and $2 = m^\alpha_{31}$, but $\alpha_{2*} \not\sqsubseteq \alpha_{3*}$). Therefore, α is not regular.

Remark. There are three known criteria for regularity in $\mathcal{B}_n$:

(1) (Zaretskii (1962)). A matrix $\alpha \in \mathcal{B}_n$ is regular if and only if its row space $V(\alpha)$ is a distributive lattice under inclusion.

(2) (Schein (1976)) A matrix $\alpha \in \mathcal{B}_n$ is regular if and only if $\alpha = \alpha(\alpha^T\alpha'\alpha^T)'\alpha$, where

α^T is the transpose of α ($\alpha^T_{ij} = 1 \iff \alpha_{ji} = 1$) and α' is the complement of α ($\alpha'_{ij} = 1 \iff \alpha_{ij} = 0$).

(3) (Kim and Roush (1978)) A matrix $\alpha \in \mathcal{B}_n$ is regular if and only if for each row α_{i*} in the row basis $r(\alpha)$, there is at least one Boolean vector $u_i \in V_n$ such that:

(a) u_i contains exactly one 1;
(b) $u_i \sqsubseteq \alpha_{i*}$;
(c) For any $\alpha_{j*} \in r(\alpha)$, $u_i \sqsubseteq \alpha_{j*}$ implies $\alpha_{i*} \sqsubseteq \alpha_{j*}$.

Theorem 3.1 is essentially Kim and Roush's criterion applied to reduced and sorted matrices. (Suppose that α is reduced and sorted. Assuming (1) and (2) of Theorem 3.1, we define $u_i = e_z$, where z is as in (1), and obtain (a)–(c) of the criterion. Conversely, assuming (a)–(c), we first prove the uniqueness of u_i and then define z as the index for which $(u_i)_z = 1$.) However, while Kim and Roush's proof is based on Zaretskii's criterion, the proof of Theorem 3.1 is elementary. Moreover, it reveals a correspondence between reduced idempotents and reduced, regular, and sorted matrices. This correspondence will be needed later and is recorded in the following algorithm.

An input for the algorithm is any reduced, regular, and sorted $\alpha \in \mathcal{B}_n$.

ALGORITHM 3.2. **(for constructing a reduced idempotent in L_α).**

1. Find k ($0 \leq k \leq n$) such that $\alpha_{i*} \neq \mathbf{0}$ for each $i \leq k$ and $\alpha_{i*} = \mathbf{0}$ for each $i > k$.
2. For every index $i \leq k$, find the unique index z_i such that $\alpha_{iz_i} = 1$ and $m^\alpha_{iz_i} = i$. By the definition of m^α_{ij}, the indices $z_1, \dots, z_k$ are distinct.
3. Order the remaining indices: $z_{k+1} < \dots < z_n$.
4. Construct a permutation $p \in S_n$ by:

$$p(i) = z_i \quad (1 \leq i \leq n),$$

and a Boolean matrix $\varepsilon \in \mathcal{B}_n$ by:

$$\varepsilon_{p(i),*} = \alpha_{i*} \quad (1 \leq i \leq n).$$

By (1) and (3) of Lemma 2.1 and the second part of the proof of Theorem 3.1, the matrix ε is a reduced idempotent in L_α. □

Note that by (4) of Lemma 2.1, the construction $\alpha \to \varepsilon$ presented in Algorithm 3.2 provides a one-to-one correspondence between the set of reduced, regular, and sorted matrices in $\mathcal{B}_n$ and the set of reduced idempotents in $\mathcal{B}_n$.

Plemmons and West (1970), Lemma 2.3, showed that each regular $\mathcal{D}$-class in $\mathcal{B}_n$ contains a reduced idempotent. An extension of Algorithm 3.2 gives an algorithm for constructing such an idempotent.

An input for the following algorithm is any regular $\alpha \in \mathcal{B}_n$.

ALGORITHM 3.3. **(for constructing a reduced idempotent in D_α).**

1. Find indices $i_1, \dots, i_k$ such that the rows $\alpha_{i_1,*}, \dots, \alpha_{i_k,*}$ form a row basis of α, and replace the remaining rows of α with $\mathbf{0}$. The resulting matrix (call it α_1) is row reduced and lies in L_α.

2. Find indices $j_1, \ldots, j_l$ such that the columns $(\alpha_1)_{*,j_1}, \ldots, (\alpha_1)_{*,j_l}$ form a column basis of α_1, and replace the remaining columns of α_1 with $\mathbf{0}$. The resulting matrix (call it α_2) is reduced (by (3) of Lemma 2.1) and lies in $R_{\alpha_1} \subseteq D_\alpha$.
3. Sort the rows of α_2 so that the resulting matrix (call it α_3) is sorted. Note that α_3 is still reduced and $\alpha_3 \in L_{\alpha_2} \subseteq D_\alpha$.
4. Use Algorithm 3.2 with α_3 as an input to construct a reduced idempotent $\varepsilon \in L_{\alpha_3} \subseteq D_\alpha$. □

Remark. If we apply Algorithm 3.3 to any $\alpha \in \mathcal{B}_n$, we can tell whether or not α is regular by checking if α_3 produced in 3. satisfies conditions (1) and (2) of Theorem 3.1.

To illustrate Algorithm 3.3, let

$$\alpha = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \in \mathcal{B}_4. \qquad \text{Then,}$$

$$\alpha_1 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \quad \alpha_2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad \alpha_3 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Note that the reduced and sorted matrix α_3 satisfies (1) and (2) of Theorem 3.1, so α_3 and α are regular. For α_3, we have $k = 3$ and $z_1 = 2$, $z_2 = 4$, and $z_3 = 1$. Consequently, *the one remaining index is* $z_4 = 3$, *the permutation (written as a sequence of images) is* $p = 2413$, and the reduced idempotent $\varepsilon \in D_\alpha$ is obtained by permuting the rows of α_3 according to p:

$$\varepsilon = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

4. Generating Reduced Idempotents

In this section, we present algorithms for generating: (i) all reduced, regular, and sorted matrices in $\mathcal{B}_n$, and (ii) all reduced idempotents in $\mathcal{B}_n$.

The second algorithm is a simple extension of the first one. The first algorithm is based on a characterization of the extensions of a reduced, regular, and sorted $\alpha \in \mathcal{B}_n$ (Theorem 4.1). An extension of such an α is defined below.

Let α be a reduced, regular, and sorted matrix of rank $r < n$. We say that $\alpha' \in \mathcal{B}_n$ is *an* **extension** *of α if α' is a reduced, regular, and sorted matrix of rank $r+1$ such that* $\alpha'_{i*} = \alpha_{i*}$ for every $i \neq r+1$. For example, if

$$\alpha = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \alpha' = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

then α' is an extension of α.

For $\alpha \in \mathcal{B}_n$, an index i, and $v \in V_n$, we denote by $\alpha[i, v]$ the matrix obtained from α

by replacing the row α_{i*} with v. Recall that e_i denotes the vector whose ith coordinate is 1 and any other coordinate is 0. Verify that for the matrices α, α' from the example above and for $v = 0110$, $\alpha' = \alpha[3, v + e_1]$.

THEOREM 4.1. *Let $\alpha \in \mathcal{B}_n$ be a reduced, regular, and sorted matrix of rank $r < n$. Then, $\alpha' \in \mathcal{B}_n$ is an extension of α if and only if $\alpha' = \alpha[r+1, v+e_z]$ for some $v \in V(\alpha)$ and some index z such that $\alpha_{*z} = \mathbf{0}$ and $\alpha_{r*} < v + e_z$.*

PROOF. Suppose α' is an extension of α. Then $\alpha' = \alpha[r+1, \alpha'_{r+1,*}]$, $\alpha_{r*} = \alpha'_{r*} < \alpha'_{r+1,*}$, and so it suffices to show that $\alpha'_{r+1,*} = v + e_z$, where v and e_z are as in the statement of the theorem. By (1) of Theorem 3.1, there is a unique index z with $\alpha'_{r+1,z} = 1$ and $\alpha_{*z} = \mathbf{0}$. Let $j_1, \ldots, j_k$ be the remaining indices (other than z) such that $\alpha_{r+1,j_1} = \cdots = \alpha_{r+1,j_k} = 1$. Defining $h_q = m^{\alpha'}_{r+1,j_q}$, $1 \le q \le k$, (note that $h_q < r+1$) and setting $v = \alpha_{h_1,*} + \cdots + \alpha_{h_k,*}$, we get $v \in V(\alpha)$ and $\alpha'_{r+1,*} = v + e_z$.

Conversely, suppose $\alpha' = \alpha[r+1, v+e_z]$, where $v \in V(\alpha)$, $\alpha_{*z} = \mathbf{0}$, and $\alpha_{r*} < v + e_z$. Since $\alpha_{*z} = \mathbf{0}$ and $\alpha'_{r+1,z} = 1$, the row $\alpha'_{r+1,*}$ is not a sum of other rows of α' and it is not included in any other row. It follows that α' is row reduced. Note that $\alpha'_{*z} \ne \mathbf{0}$ cannot be a sum of other columns of α'. For any other column $\alpha'_{*j} \ne \mathbf{0}$, we have $\alpha_{*j} \ne \mathbf{0}$ and for any indices $j_1, \ldots, j_k$:

$$\alpha'_{*j} = \alpha'_{*j_1} + \ldots + \alpha'_{*j_k} \Longrightarrow \alpha_{*j} = \alpha_{*j_1} + \ldots + \alpha_{*j_k}.$$

It follows that α' is also column reduced, and so it is a reduced matrix with row and column rank $r+1$. Since $\alpha_{r*} < v + e_z = \alpha'_{r+1,*}$, the matrix α' is sorted. Therefore, it remains to show that α' satisfies conditions (1) and (2) of Theorem 3.1 for the index $i = r+1$. Since $\alpha' = \alpha[r+1, v+e_z]$ with $v \in V(\alpha)$ and $\alpha_{*z} = \mathbf{0}$, z is the only index such that $\alpha'_{r+1,z} = 1$ and $m^{\alpha'}_{r+1,z} = r+1$. This shows (1). To show (2), assume $\alpha'_{r+1,j} = 1$ for $j \ne z$, and let $h = m^{\alpha'}_{r+1,j}$. Since $\alpha'_{r+1,*} = v + e_z$, there is an index $i \le r$ such that $\alpha_{ij} = 1$ and $\alpha_{i*} \sqsubseteq \alpha'_{r+1,*}$. Since $h = m^{\alpha'}_{r+1,j} = m^{\alpha}_{ij}$, we have $\alpha'_{h*} = \alpha_{h*} \sqsubseteq \alpha_{i*} \sqsubseteq \alpha'_{r+1,*}$, which concludes the proof. □

The following algorithm constructs successively lists $L_0, L_1, \ldots, L_n$ of all reduced, regular, and sorted matrices in $\mathcal{B}_n$ of rank $0, 1, \ldots, n$, respectively. The list L_{r+1} is obtained from L_r by constructing the extensions of each $\alpha \in L_r$.

ALGORITHM 4.2. **(for generating all reduced, regular, and sorted $\alpha \in \mathcal{B}_n$).**

1. Begin with constructing lists L_0 and L_1 of all reduced, regular, and sorted matrices of rank, respectively, 0 and 1:

$$L_0: \quad \alpha_1^{(0)}, \qquad L_1: \quad \alpha_1^{(1)},\ \alpha_2^{(1)},\ \ldots,\ \alpha_n^{(1)},$$

   where $\alpha_1^{(0)}$ is the zero matrix, and $\alpha_q^{(1)}$ $(1 \le q \le n)$ is the matrix with 1 at the entry $(1, q)$ and 0 at all other entries.
2. Set $r = 1$ (r will run through $1, 2, \ldots, n$).
3. At this point, a list L_r of all reduced, regular, and sorted matrices of rank r has already been constructed:

$$L_r: \quad \alpha_1^{(r)},\ \alpha_2^{(r)},\ \ldots,\ \alpha_{\ell_r}^{(r)}.$$

   Set $i_r = 1$ (i_r will run through $1, 2, \ldots, \ell_r$) and $L_{r+1} = \emptyset$.

4. Set $\alpha = \alpha^{(r)}_{i_r}$.
5. Construct the row space $V(\alpha)$.
6. Construct the set $M(\alpha)$ of all indices z such that $\alpha_{*z} = 0$.
7. For each $v \in V(\alpha)$ and each $z \in M(\alpha)$ do the following:
   (a) Check if $\alpha_{r*} < v + e_z$. If not, skip (b) and (c), otherwise continue.
   (b) Construct the matrix $\alpha' = \alpha[r+1, v+e_z]$. By Theorem 4.1, α' is an extension of α.
   (c) Add α' to the list L_{r+1}.
8. Increase i_r by 1. If $i_r \le \ell_r$, go to 4., otherwise continue.
9. Increase r by 1. If $r < n$, go to 3., otherwise the lists $L_0, L_1, \ldots, L_n$ have been constructed. □

An essential part of Algorithm 4.2 is the computation of all extensions of a given $\alpha \in L_r$ (steps 5.–7.). To illustrate this part, assume that $n = 4$ and

$$\alpha = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \in L_2.$$

Then $V(\alpha) = \{v_1, v_2, v_3\}$, where $v_1 = 0000$, $v_2 = 1000$, $v_3 = 1010$, and $M(\alpha) = \{2, 4\}$. There are six possible sums $v + e_z$ ($v \in V(\alpha)$, $z \in M(\alpha)$), of which three: $v_2 + e_2 = 1100$, $v_3 + e_2 = 1110$, and $v_3 + e_4 = 1011$ are greater than $\alpha_{2*} = 1010$. Therefore, α has three extensions:

$$\alpha' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \alpha'' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad \alpha''' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

where $\alpha' = \alpha[3, v_2 + e_2]$, $\alpha'' = \alpha[3, v_3 + e_2]$, and $\alpha''' = \alpha[3, v_3 + e_4]$. The algorithm has two loops. The inner loop (4.–8.–4.) computes the extensions of all $\alpha \in L_r$, which is equivalent to constructing L_{r+1}. The outer loop (3.–9.–3.) repeats the inner loop $n-1$ times, with $r = 1, 2, \ldots, n-1$, to construct successively: L_2 from L_1, L_3 from L_2, ..., L_n from L_{n-1}. Finally, since each $\beta \in L_{r+1}$ $(1 \le r < n)$ is an extension of exactly one $\alpha \in L_r$, namely $\alpha = \beta[r+1, \mathbf{0}]$, the algorithm generates all reduced, regular, and sorted matrices, and each such matrix is generated exactly once.

Combining Algorithm 4.2 and Algorithm 3.2, we obtain an algorithm that constructs lists $E_0, E_1, \ldots, E_n$ of all reduced idempotents in $\mathcal{B}_n$ of rank $0, 1, \ldots, n$, respectively.

ALGORITHM 4.3. **(for generating all reduced idempotents in $\mathcal{B}_n$).**

1. *Use Algorithm 4.2 to construct lists $L_0, L_1, \ldots, L_n$ of all reduced, regular, and sorted matrices in $\mathcal{B}_n$ of rank $0, 1, \ldots, n$, respectively.*
2. *Set $r = 0$ (r will run through $0, 1, \ldots, n$).*
3. *Set $E_r = \emptyset$.*
4. *For each $\alpha \in L_r$ do the following:*
   (a) Use Algorithm 3.2 with α as an input to construct a reduced idempotent $\varepsilon \in L_\alpha$.
   (b) Add ε to E_r.

5. Increase r by 1. If $r \leq n$, go to 3., otherwise the lists $E_0, E_1, \dots, E_n$ have been constructed. □

By the comment after Algorithm 3.2, Algorithm 4.3 generates all reduced idempotents in $\mathcal{B}_n$, and each such idempotent is generated exactly once. Every $\varepsilon \in E_r$ is obtained by permuting rows of the corresponding $\alpha \in L_r$.

Remark. There is a natural one-to-one correspondence between $n \times n$ Boolean matrices and binary relations on a set $X = \{x_1, x_2, \dots, x_n\}$:

$$\alpha \longrightarrow A, \quad \text{where} \quad (x_i, x_j) \in A \iff \alpha_{ij} = 1.$$

By Plemmons and West (1970), Theorem 2.4, a Boolean matrix $\alpha \in \mathcal{B}_n$ is a reduced idempotent of rank n if and only if the corresponding binary relation A is a partial order relation on X. It follows that the list E_n constructed by Algorithm 4.3 consists of all (matrix representations of) partial order relations on X.

The most elementary method of generating a list E of all reduced idempotents in $\mathcal{B}_n$ would consist of the following two steps: (1) start a procedure generating all $n \times n$ Boolean matrices; (2) for each new matrix α generated in (1), check if α is a reduced idempotent and, if it is, add α to E. An analysis of the complexity of Algorithm 4.3 should consist in a comparison with the naive algorithm described in (1) and (2).

To generate the reduced idempotents, Algorithm 4.3 uses the reduced, regular, and sorted matrices. Each such matrix α is processed as follows: (i) all extensions of α are computed (steps 5.–7. of Algorithm 4.2) ((i) is performed only if rank $\alpha < n$); and (ii) the reduced idempotent ε in L_α is constructed (Algorithm 3.2). Since the number of reduced, regular, and sorted matrices is the same as the number R_n of reduced idempotents, Algorithm 4.3 processes R_n matrices. On the other hand, the naive algorithm would process 2^{n^2} matrices.

We now estimate the number of steps required by each algorithm to process a single matrix. (By a step we mean an operation performed on two Boolean values.) Let α be a reduced, regular, and sorted matrix of rank r. To estimate the cost of computing the extensions of α (this cost is zero if $r = n$), look at steps 5.–7. of Algorithm 4.2. To save time, the sets $V(\alpha)$ and $M(\alpha)$ may be computed and stored right after $\alpha = \beta[r, v + e_z]$ is constructed as an extension of $\beta \in L_{r-1}$: $V(\alpha) = V(\beta) \cup \{(v + e_z) + w : w \in V(\beta)\}$ and $M(\alpha) = M(\beta) - \{z\}$. To construct $V(\alpha)$, we need to compute the sums $(v + e_z) + w$ (which requires $|V(\beta)|n$ steps) and eliminate repeated sums (at most $(1 + 2 + \cdots + (|V(\beta)| - 1))n$ steps). Thus, computing $V(\alpha)$ requires $|V(\beta)|n + \frac{1}{2}(|V(\beta)| - 1)|V(\beta)|n = \frac{1}{2}|V(\beta)|(|V(\beta)| + 1)n \leq 2^{r-2}(2^{r-1} + 1)n$ steps. The cost of computing $M(\alpha) = M(\beta) - \{z\}$ is negligible. Finally, (a) and (b) in 7. of Algorithm 4.2 require at most ((b) is sometimes skipped) $n + n^2$ steps and are performed $|V(\alpha)||M(\alpha)| \leq 2^r(n - r)$ times (since $|M(\alpha)| = n - r$). Therefore, the computation of extensions requires at most $\frac{1}{2}|V(\beta)|(|V(\beta)| + 1)n + |V(\alpha)|(n - r)(n + n^2)$ steps. (In the worst case, which happens when α is equivalent to the $r \times r$ identity matrix, this number is $2^{r-2}(2^{r-1} + 1)n + 2^r(n - r)(n + n^2)$.) Since the construction of the reduced idempotent $\varepsilon \in L_\alpha$ (Algorithm 3.2) requires $O(n^2)$ steps, the cost of processing α is at most $\frac{1}{2}|V(\beta)|(|V(\beta)| + 1)n + |V(\alpha)|(n - r)(n + n^2) + O(n^2)$. (The cost is $O(n^2)$ if $r = n$). As to the naive algorithm, it would require $O(n^3)$ steps to check if a given matrix is a reduced idempotent.

To summarize: Algorithm 4.3 processes R_n matrices (where R_n is the number of reduced idempotents in $\mathcal{B}_n$) and the cost of processing a single α of rank r is at most

$\frac{1}{2}|V(\beta)|(|V(\beta)|+1)n + |V(\alpha)|(n-r)(n+n^2) + O(n^2)$ where $\beta = \alpha[r, 0]$. (The cost is $O(n^2)$ if $r = n$.) The naive algorithm would process 2^{n^2} matrices at the cost $O(n^3)$ each. It is the fact that R_n is much smaller than 2^{n^2} that makes Algorithm 4.3 more efficient. For example, a computer program based on Algorithm 4.3, written in Turbo Pascal and run on a PC 486, took 23 seconds to generate the 159, 126 reduced idempotents in $\mathcal{B}_6$. On the other hand, for $n = 6$, the naive algorithm would have to process $2^{36} = 68,719,476,736$ matrices.

We should also point out that storing $V(\alpha)$ and $M(\alpha)$ will not take much memory if the computation is organized as follows. Let L be a list to store reduced, regular, and sorted matrices α (together with $V(\alpha)$ and $M(\alpha)$). Initially $L = L_1$ (see Algorithm 4.2). While L is not empty, remove the last matrix α from L, process α in the sense of (i) and (ii) above, and for each extension α' of α, either add α' to L (if rank $\alpha' < n$) or process α' in the sense of (ii) only (if rank $\alpha' = n$). We found out, by running the computer program mentioned above, that out of 159, 126 reduced, regular, and sorted matrices in $\mathcal{B}_6$, no more than 52 were stored in memory at any single moment.

As an example, we apply Algorithm 4.3 to generate all reduced idempotents in $\mathcal{B}_3$. The first step is to use Algorithm 4.2 to generate all reduced, regular, and sorted matrices in $\mathcal{B}_3$ (lists L_0, L_1, L_2, and L_3). Then, by permuting rows of these matrices, the reduced idempotents are generated (lists E_0, E_1, E_2, and E_3).

$$L_0: \begin{pmatrix} 0&0&0\\0&0&0\\0&0&0 \end{pmatrix}. \qquad L_1: \begin{pmatrix} 1&0&0\\0&0&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&0\\0&0&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&0&0\\0&0&0 \end{pmatrix}.$$

$$L_2: \begin{pmatrix} 1&0&0\\1&1&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 1&0&0\\1&0&1\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&0\\1&0&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&0\\1&1&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&0\\0&1&1\\0&0&0 \end{pmatrix},$$
$$\begin{pmatrix} 0&0&1\\1&0&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&1\\1&0&1\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&1&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&1&1\\0&0&0 \end{pmatrix}.$$

$$L_3: \begin{pmatrix} 1&0&0\\1&1&0\\1&1&1 \end{pmatrix}, \begin{pmatrix} 1&0&0\\1&0&1\\1&1&0 \end{pmatrix}, \begin{pmatrix} 1&0&0\\1&0&1\\1&1&1 \end{pmatrix}, \begin{pmatrix} 0&1&0\\1&0&0\\1&0&1 \end{pmatrix}, \begin{pmatrix} 0&1&0\\1&0&0\\1&1&1 \end{pmatrix},$$
$$\begin{pmatrix} 0&1&0\\1&1&0\\1&1&1 \end{pmatrix}, \begin{pmatrix} 0&1&0\\0&1&1\\1&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&0\\0&1&1\\1&1&0 \end{pmatrix}, \begin{pmatrix} 0&1&0\\0&1&1\\1&1&1 \end{pmatrix}, \begin{pmatrix} 0&0&1\\1&0&0\\1&1&0 \end{pmatrix},$$
$$\begin{pmatrix} 0&0&1\\1&0&0\\1&1&1 \end{pmatrix}, \begin{pmatrix} 0&0&1\\1&0&1\\1&1&1 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&1&0\\1&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&1&0\\1&0&1 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&1&0\\1&1&0 \end{pmatrix},$$
$$\begin{pmatrix} 0&0&1\\0&1&0\\1&1&1 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&1&1\\1&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&1&1\\1&0&1 \end{pmatrix}, \begin{pmatrix} 0&0&1\\0&1&1\\1&1&1 \end{pmatrix}.$$

$$E_0: \begin{pmatrix} 0&0&0\\0&0&0\\0&0&0 \end{pmatrix}. \qquad E_1: \begin{pmatrix} 1&0&0\\0&0&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&0\\0&1&0\\0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&0\\0&0&0\\0&0&1 \end{pmatrix}.$$

$$E_2: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

$$E_3: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

The list E_3 consists of all (matrix representations of) partial order relations on a set with three elements.

With Algorithm 4.3 as a starting point, one can find all column reduced idempotents in $\mathcal{B}_n$: For each reduced idempotent α, construct all possible matrices β that can be obtained from α by replacing each zero row of α with a vector v (possibly zero) from the row space $V(\alpha)$. A similar construction (using zero columns and column space) yields all row reduced idempotents. The above "row" construction applied to the row reduced idempotents (or the "column" construction applied to the column reduced idempotents) would yield all idempotents, but it would not be efficient since the same idempotent could be generated many times.

References

Chaudhuri, R., Mukherjea, A. (1980). Idempotent Boolean matrices. *Semigroup Forum* **21**, 273–282.

Culberson, J.C., Rawlins, G.J.E. (1991). New results from an algorithm for counting posets. *Order* **7**, 361–374.

Erné, M., Stege, K. (1991). Counting finite posets and topologies. *Order* **8**, 247–265.

Howie, J.M. (1976). *An Introduction to Semigroup Theory*. London: Academic Press.

Kim, K.H. (1982). *Boolean Matrix Theory and Applications*. New York: Marcel Dekker.

Kim, K.H., Roush, F.W. (1978). Inverses of Boolean matrices. *J. Linear Alg. Appl.* **22**, 247–262.

Kleitman, D.J., Rothschild, B.L. (1975). Asymptotic enumeration of partial orders on a finite set. *Trans. Amer. Math. Soc.* **205**, 205–220.

Lallement, G. (1979). *Semigroups and Combinatorial Applications*. New York: John Wiley and Sons.

Plemmons, R.J., West, M.T. (1970). On the semigroup of binary relations. *Pacific J. Math.* **35**, 743–753.

Schein, B.M. (1976). Regular elements of the semigroup of all binary relations. *Semigroup Forum* **13**, 95–102.

Zaretskii, K.A. (1962). Regular elements in the semigroup of binary relations. *Uspekhi Mat. Nauk* **17**, 177–179. (Russian)

Zaretskii, K.A. (1963). The semigroup of binary relations. *Mat. Sb.* **61**, 291–305. (Russian)