Threshold Logic

The Reduction Method And Its Applications

By A.J.J. Regeer

DRAFT VERSION

January 2009

TO DO List

- Chapter(s) 6
- Literatuurlijst
- Titlepage en de rest
- Index
- Verwijzingen nakijken
- Inhoudelijk nakijken
- Grammatica
- Vormgeving
- Structuur paragrafen
- Zinsconstructie

- Thesis Claude E. Shannon
- Boeken over schrijven
- Vergeet niet Draft Mode uit te zetten bij finale productie

1 Introduction

CHAPTER

In digital logic design we are accustomed to using Boolean gates when designing logic circuits. In threshold logic we propose another kind of gate with which to construct the circuits.

Threshold logic is an alternative way for implementing Boolean functions. In Digital logic design we use the familiar Boolean gates such as AND, OR, and NOT, etc. to construct circuits that are the implementation of Boolean functions we require.

In threshold logic we have instead only one building block, namely the threshold logic unit, or TLU for short. It has been shown in the literature that with this building block we can implement all Boolean functions. In other words it is as expressive as are the Boolean gates.

When working with the TLU some questions arise naturally. How can we determine if a TLU can implement some Boolean function. If it can implement Boolean function how do we find the proper weights and threshold value for the TLU.

It has been shown that these questions can be answered with the simplex method. But in this thesis we propose another method for determining these questions. This

method offers the added bonus of giving us a way to implement a Boolean function in a structured way.

2 Vectors

CHAPTER

2.1 Vectors

We start with the definition of a vector. In linear algebra it is customary to define the vector by its operations and properties, but that would be too general for our purposes. So, instead we have chosen to define its form(s) and operations in terms of intgers and its operations and properties.

Definition **2.1** A *vector* is an object of the form

$$\begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

where $a_1, a_2, \dots, a_n$ are elements of the set of integers, that is $a_i \in \mathbb{Z}$, for $1 \leq i \leq n$. The form on the left is called a *row vector* and the form on the right is called a

column vector. The row vector is said to have a *horizontal orientation*, and the column vector is said to have a *vertical orientation*. The elements $a_1, a_2, \ldots, a_n$ are called the *entries*, *components*, or *coordinates* of the vector. The number a_i is called the i^{th} entry of the vector. The *size* or *length* of a vector is the number of entries it contains. The row vector $\begin{pmatrix} 0 & 0 & \cdots & 0 \end{pmatrix}$ is the zero row vector and is denoted by 0. Also the column vector for which all entries are zero is called the zero column vector and is also denoted by 0. □

Readers who are familiar with linear algebra should note that in this text the entries of vectors are integers and not, as is usual in linear algebra, elements of a field. As will become clear in the remainder of this text, this restriction will have some significant consequences as many results in linear algebra can not be obtained in this text.

From now on we will use the notation $v = (v_1, v_2, \ldots, v_n)$ when we either mean a column vector or a row vector. This can be used in situations where it either is clear from the context which form is implied, or in situations where we do not care what the exact form is. Note that the notation $(a_1, a_2, \ldots, a_n)$ is different from the notation for a row vector: $\begin{pmatrix} a_1 & a_2 & \ldots & a_n \end{pmatrix}$, where there are no commas. The next definition introduces the operations that are defined for vectors.

DEFINITION **2.2** ***(addition and multiplication)***

1. **Addition**

   If $x = \begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix}$ is a row vector of size n and $y = \begin{pmatrix} y_1 & y_2 & \ldots & y_n \end{pmatrix}$ is another row vector of size n then their *sum* $x + y$ is the row vector defined by

   $$x + y = \begin{pmatrix} [x_1 + y_1] & [x_2 + y_2] & \ldots & [x_n + y_n] \end{pmatrix}$$

   If $x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ and $y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$ are both column vectors of size n then their *sum* $x + y$ is the column vector defined by

   $$x + y = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{pmatrix}$$

2. **Multiplication by an integer c**

If $x = (\ a_1 \quad a_2 \quad \cdots \quad a_n\ )$ is a row vector of size n and c is an integer, then the *product* of c and x, denoted by cx, is the row vector defined by

$$cx = (\ cx_1 \quad cx_2 \quad \cdots \quad cx_n\ )$$

If $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ is a column vector of size n and c is an integer, then the *product* of c and x, denoted by cx, is the column vector defined by

$$cx = \begin{pmatrix} cx_1 \\ cx_2 \\ \vdots \\ cx_n \end{pmatrix}$$

□

Note that addition as defined above only applies to vectors that have the same orientation and are of the same length. In other cases addition has no meaning.

The following theorem lists a number of properties of addition and scalar multiplication which follow immediately from the definitons, as the reader will easily verify. We shall use these properties freely in what follows without giving explicit reference after each use. The theorem shall be given without proof.

THEOREM **2.1** Let x, y and z be vectors such that they are of equal length and have the same orientation. Further, let a and b be integers. Then we have the following properties for addition and multiplication.

(P 1) $x + y = y + x$ (commutativity of addition.)

(P 2) $(x + y) + z = x + (y + z)$ (associativity of addition.)

(P 3) $x + 0 = x$

(P 4) There is a unique vector, denoted $-x$, such that $x + (-x) = 0$

(P 5) $1 \cdot x = x$

(P 6) $(ab)x = a(bx)$

(P 7) $a(x + y) = ax + ay$

(P 8) $(a + b)x = ax + bx$

■

DEFINITION **2.3** Let $\mathcal{C}^n$ be the set of all column vectors of length n and let $\mathcal{R}^n$ be the set of all row vectors of length n. We call $\mathcal{C}^n$ and $\mathcal{R}^n$ *vector spaces*, where $\mathcal{C}^n$ is a column space and $\mathcal{R}^n$ is a row space. Let $\mathcal{C}$ be the set of all column spaces, that is $\mathcal{C} = \bigcup_{n=1}^{\infty}\{\mathcal{C}^n\}$, and let $\mathcal{R}$ be the set of all row spaces, that is $\mathcal{R} = \bigcup_{n=1}^{\infty}\{\mathcal{R}^n\}$. Finally, let $\mathcal{V}$ be the set of all vector spaces, that is $\mathcal{V} = \mathcal{R} \cup \mathcal{C}$. □

DEFINITION **2.4** Let $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_m)$ be vectors of length n and m respectively. Then a and b are *equal*, denoted $a = b$, if

1. They have the same orientation
2. They have the same length, that is $n = m$
3. All their corresponding entries are aqual, that is

$$a_i = b_i \quad \text{for} \quad 1 \leq i \leq \min(n, m)$$

□

Note that two vectors are never equal if their lengths are not equal, nor are they equal if one of the vectors is a row vector and the other is a column vector.

DEFINITION **2.5** Let $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_m)$ be vectors of length n and m respectively. Then a is less then b, denoted $a < b$, if

1. They have the same orientation
2. They have the same length, that is $n = m$
3. Each entry in a must be less then its corresponding entry in b, that is

$$a_i < b_i \quad \text{for} \quad 1 \leq i \leq \min(n, m)$$

□

DEFINITION **2.6** Let $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_m)$ be vectors of length n and m respectively. Then a is greater then b, denoted $a > b$, if

1. They have the same orientation
2. They have the same length, that is $n = m$

3. Each entry in a is greater then its corresponding entry in b, that is

$$a_i > b_i \quad \text{for} \quad 1 \leq i \leq \min(n, m)$$

□

DEFINITION **2.7** Let $v = (\ v_1 \quad v_2 \quad \dots \quad v_n\ )$ be a row vector and let $w = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$ be a column vector. Both are of equal length. Then multiplication between v and w, denoted $v \cdot w$ or vw, is defined by

$$v \cdot w = \sum_{i=1}^{n} v_i \cdot w_i$$

□

DEFINITION **2.8** Let $v = (v_1, v_2, \dots, v_n)$ be a vector of length n. Then v is called

1. *positive* if all elements of v are positive, i.e.

$$v_i > 0 \quad \text{for } 1 \leq i \leq n,$$

2. *semipositive* if all elements of v are either zero or positive with at least one entry positive, i.e.

$$v_i = 0 \text{ or } v_i > 0 \quad \text{for } 1 \leq i \leq n,$$

and there is an i with $1 \leq i \leq n$ such that $v_i > 0$.

3. *negative* if all elements of v are negative, i.e.

$$v_i < 0 \quad \text{for } 1 \leq i \leq n,$$

4. *seminegative* if all elements of v are either zero or negative with at least one entry negative, i.e.

$$v_i = 0 \text{ or } v_i < 0 \quad \text{for } 1 \leq i \leq n,$$

and there is an i with $1 \leq i \leq n$ such that $v_i < 0$.

5. *nonnegative* if all elements of v are either zero or positive, i.e.

$$v_i = 0 \text{ or } v_i > 0 \quad \text{for } 1 \leq i \leq n$$

6. *mixed* if there are positive and negative entries in v, i.e. there exist i and j with $1 \leq i, j \leq n$ such that $v_i < 0$ and $v_j > 0$.

7. *paved* if there is a $q \in \mathbb{Z}$, with $q > 0$, such that

$$|a_i| = q \text{ or } a_i = 0 \text{ for } 1 \leq i \leq n.$$

□

2.2 Scalar Product, Norm, Transpose

DEFINITION **2.9** ***(transpose)***
Let $v = (v_1, v_2, \ldots, v_n)$ be a vector. Then the *transpose* of v, denoted by v^{T}, is defined by

$$v^{\mathsf{T}} = \begin{cases} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} & \text{if } v \text{ is a horizontal vector} \\ \\ \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix} & \text{if } v \text{ is a vertical vector} \end{cases}$$

□

THEOREM **2.2** Let x and y be vectors such that they are of equal length and have the same orientation. Further, let a be an integer. Then we have the following properties for the transpose.

(P 1) $(a \cdot x)^{\mathrm{T}} = a \cdot x^{\mathrm{T}}$

(P 2) $(x + y)^{\mathrm{T}} = x^{\mathrm{T}} + y^{\mathrm{T}}$

■

DEFINITION **2.10** ***(inner product)***
Let $v = (v_1, v_2, \ldots, v_n)$ and $w = (w_1, w_2, \ldots, w_n)$ be two vectors of equal length and with the same orientation. Then the scalar or *inner product*, denoted by $v \star w$, is defined by

$$v \star w = \begin{cases} v^{\mathsf{T}} \cdot w & \text{if } v \text{ is a vertical vector} \\ v \cdot w^{\mathsf{T}} & \text{if } v \text{ is a horizontal vector} \end{cases}$$

□

THEOREM **2.3** Let x, y and z be vectors such that they are of equal length and have the same orientation. Further, let a be an integer. Then we have the following properties for the scalar product.

(P 1) $x \star (y + z) = x \star y + x \star z$ (distributivity)

(P 2) $x \star y = y \star x$ (commutativity)

(P 3) $x \star (a \cdot y) = a \cdot (x \star y)$

PROOF

We will prove property 3:

$$\begin{aligned} x \star (a \cdot y) &= \sum_{i=1}^{n} x_i (ay)_i \\ &= \sum_{i=1}^{n} x_i (a y_i) \\ &= \sum_{i=1}^{n} (x_i a) y_i \\ &= \sum_{i=1}^{n} (a x_i) y_i \\ &= \sum_{i=1}^{n} a (x_i y_i) \\ &= a \sum_{i=1}^{n} (x_i y_i) \\ &= a (x \star y) \end{aligned}$$

■

THEOREM **2.4** Let a and b be vectors of length n such that $a > b$. Let x be a semipositive vector of length n. Then

$$x \star a > x \star b$$

PROOF

The proof is by induction on the length of the vectors. If the length of x is 1 then x is an integer and because x is a semipositive vector it must be a positive integer. So we have $a_1 > b_1$ and it follows that $x_1 a_1 > x_1 b_1$, which results in

$$x \star a > x \star b$$

Now suppose the theorem is correct for $n-1$, then

$$xa = \sum_{i=1}^{n} x_i a_i = \sum_{i=1}^{n-1} + x_n a_n$$

and

$$xb = \sum_{i=1}^{n} x_i b_i = \sum_{i=1}^{n-1} x_i b_i + x_n b_n$$

Now we have $x_n = 0$ or $x_n > 0$. If $x_n = 0$ then we must have that one of the other entries of x must be positive, because x is semipositive. Now we form the vector x' that exists of the first $n-1$ entries of x, i.e. $x'_i = x_i$ for $1 \leq i \leq n-1$. We also define the vectors a' and b' which are defined similarly. That is $a'_i = a_i$ for $1 \leq i \leq n-1$ and $b'_i = b_i$ for $1 \leq i \leq n-1$. With x' a semipositive vector we can apply the theorem inductively to a', b' and x', which results in

$$x'a' > x'b'$$

Now we have

$$x'a' = \sum_{i=1}^{n-1} x'_i a'_i = \sum_{i=1}^{n-1} x_i a_i$$

and

$$x'b' = \sum_{i=1}^{n-1} x'_i b'_i = \sum_{i=1}^{n-1} x_i b_i$$

So we have

$$\begin{aligned}
\sum_{i=1}^{n-1} x_i a_i &> \sum_{i=1}^{n-1} x_i b_i \\
\sum_{i=1}^{n-1} x_i a_i + 0 &> \sum_{i=1}^{n-1} x_i b_i + 0 \\
\sum_{i=1}^{n-1} x_i a_i + 0 \cdot a_n &> \sum_{i=1}^{n-1} x_i b_i + 0 \cdot b_n \\
\sum_{i=1}^{n-1} x_i a_i + x_n \cdot a_n &> \sum_{i=1}^{n-1} x_i b_i + x_n \cdot b_n \\
\sum_{i=1}^{n} x_i a_i &> \sum_{i=1}^{n} x_i b_i \\
x \star a &> x \star b
\end{aligned}$$

If $x_n > 0$ then it is possible that the first $n-1$ entries of x are all zero. Then we have

$$\sum_{i=1}^{n-1} x_i a_i = 0 = \sum_{i=1}^{n-1} x_i b_i$$

We know that $a_n > b_n$, so we also have $x_n a_n > x_n b_n$. Now, we have

$$\begin{aligned} x_n a_n &> x_n b_n \\ x_n a_n + 0 &> x_n b_n + 0 \\ x_n a_n + \sum_{i=1}^{n-1} x_i a_i &> x_n b_n + \sum_{i=1}^{n-1} x_i b_i \\ \sum_{i=1}^{n} x_i a_i &> \sum_{i=1}^{n} x_i b_i \\ x \star a &> x \star b \end{aligned}$$

When the first $n-1$ entries of x are not all zero then we can apply the theorem inductively again to a', b', and x', which results in

$$x'a' > x'b'$$

Now we have that

$$\begin{aligned} a_n &> b_n \\ x_n a_n &> x_n b_n \\ x'a' + x_n a_n &> x'a' + x_n b_n \end{aligned}$$

and we have

$$\begin{aligned} x'a' &> x'b' \\ x'a' + x_n b_n &> x'b' + x_n b_n \end{aligned}$$

and so by the transitive law for integers we have

$$x'a' + x_n a_n > x'b' + x_n b_n$$

which is equivalent to saying that

$$\begin{aligned} \sum_{i=1}^{n-1} x'_i a'_i + x_n a_n &> \sum_{i=1}^{n-1} x'_i b'_i + x_n b_n \\ \sum_{i=1}^{n-1} x_i a_i + x_n a_n &> \sum_{i=1}^{n-1} x_i b_i + x_n b_n \\ \sum_{i=1}^{n} x_i a_i &> \sum_{i=1}^{n} x_i b_i \\ x \star a &> x \star b \end{aligned}$$

And so the theorem is proved. ■

We will next define the norm of a vector. Readers who are familiar with the norm of a vector in linear algebra should be careful to note that this norm as it is defined in this text is different from the norm in linear algebra.

The norm is a measure for its 'bigness'. A vector with large entries has a large norm.

DEFINITION **2.11** ***(norm)***
Let $v = (v_1, v_2, \ldots, v_n)$ be a vector of length n. Then the *norm* of v, denoted by $\|v\|$, is defined by

$$\|v\| = v \star v$$

□

THEOREM **2.5** Let x be a vector. Further, let a be an integer. Then we have the following properties for the norm.

(P 1) $\|ax\| = a^2 \|x\|$

(P 2) $\|x\| \geq 0$

(P 3) if $\|x\| = 0$ then $x = 0$

■

DEFINITION **2.12** Let v and w be vectors of equal length and with the same orientation. We define the vectors $\min(v, w)$ and $\max(v, w)$ as follows:

$$\min(v, w) = \begin{cases} v & \text{if } \|v\| \leq \|w\| \\ w & \text{if } \|v\| > \|w\| \end{cases}$$

$$\max(v, w) = \begin{cases} v & \text{if } \|v\| \geq \|w\| \\ w & \text{if } \|v\| < \|w\| \end{cases}$$

□

2.3 Division

In this section we introduce division of vectors. It is similar to what we are used to in integer division. Then we introduce some concepts that allow us to prove some results that will be used in later chapters.

DEFINITION **2.13** Given an integer z and a vector v, we say that z divides v to mean that there is a vector q such that $v = zq$, and we call z a *divisor* of v. □

EXAMPLE **1** Let $v = (\ 3 \quad 15 \quad 12 \quad 6 \quad 21\ )$ and let $q = (\ 1 \quad 5 \quad 4 \quad 2 \quad 7\ )$, then $d = 3$ divides v. □

DEFINITION **2.14** Let v be a vector and let the integer d be a divisor of v, then let the vector q be such that $v = dq$. We call q its *quotient*, and it is denoted by v/d. □

EXAMPLE **2** In example 1 the quotient is q. □

DEFINITION **2.15** ***(greatest divisor)***
Let v be a vector which is not equal to zero. Then the *greatest divisor* of v is the unique positive integer d such that

1. d is a divisor of v,
2. d is greater than or equal to every other divisor of v, i.e.

$$\text{if } c \text{ divides } v \text{ then } c \leq d$$

We denote the greatest divisor of the vector v by

$$\mathrm{gd}(v).$$

□

DEFINITION **2.16** ***(fundamental vector)***
Let v be a vector not equal to the zero vector. Then v is a *fundamental* vector if the largest divisor of v is equal to 1, i.e. $\mathrm{gd}(v) = 1$ □

EXAMPLE **3** In example 1 the vector$q$ is a fundamental vector. □

DEFINITION **2.17** ***(base vector)***
Let v be a vector not equal to the zero vector, and let the positive integer d be its greatest divisor, i.e. $d = gd(v)$, then the vector v/d is called its *base vector*, denoted by v^*. □

THEOREM **2.6** Let v and w be vectors of equal length and with the same orientation. Furthermore, the vectors v and w are both not equal to the zero vector. If $(v \star w)^2 = \|v\|\,\|w\|$ then

$$(v^* \star w)^2 = \|v^*\|\,\|w\|$$

PROOF

$$(v \star w)^2 = \|v\|\,\|w\|$$

With $v = av^*$ for some positive integer a we get

$$(v \star w)^2 = ((av^*) \star w)^2 = (a(v^*) \star w)^2 = a^2(v^* \star w)^2$$

and

$$\|v\| \, \|w\| = \|av^*\| \, \|w\| = a^2 \, \|v^*\| \, \|w\|$$

So we get

$$a^2(v^* \star w)^2 = a^2 \, \|v^*\| \, \|w\|$$

which results in

$$(v^* \star w)^2 = \|v^*\| \, \|w\|$$

■

THEOREM **2.7** Let v be a vector which is not equal to the zero vector. Let w be its base vector, i.e. $w = v^*$. Then $\|w\| \leq \|v\|$.

PROOF

We have $v = aw$ for some positive integer a, so

$$a \geq 1 \qquad \Rightarrow \qquad a^2 \geq 1 \qquad \Rightarrow \qquad a^2 \, \|w\| \geq \|w\|$$

and

$$\|v\| = \|aw\| = a^2 \, \|w\| \geq \|w\| \text{ with the previous result}$$

So we have $\|w\| \leq \|v\|$, as required. ■

THEOREM **2.8** Let v and w be vectors, both not equal to the zero vector, that are of equal orientation and have the same length. Then $(v \star w)^2 = \|v\| \, \|w\|$ if and only if $(v^* \star w^*)^2 = \|v^*\| \, \|w^*\|$.

PROOF

Let the positive integer c be the greatest divisor of v and let the positive integer d be the greatest divisor of w. Further let the vector p be the base vector of v, i.e. $p = v^*$, and let the vector q be the base vector of w, i.e. $q = w^*$. Suppose $(v \star w)^2 = \|v\| \, \|w\|$. Then, because $v = cp$ and $w = dq$, we have

$$(v \star w)^2 = ((cp) \star (dq))^2 = ((cd)(p \star q))^2 = (cd)^2(p \star q)^2 = c^2d^2(p \star q)^2 \tag{2.1}$$

and we have

$$\|v\| = c^2 \, \|p\| \quad \text{and} \quad \|w\| = d^2 \, \|q\| \tag{2.2}$$

So

$$\|v\| \, \|w\| = c^2 \, \|p\| \, d^2 \, \|q\| = c^2d^2 \, \|p\| \, \|q\| \tag{2.3}$$

So from eq. (2.1) and eq. (2.3) we have

$$c^2d^2(p \star q)^2 = c^2d^2 \|p\| \|q\|$$

which results in

$$(p \star q)^2 = \|p\| \|q\|$$

which is equal to

$$(v^* \star w^*)^2 = \|v^*\| \|w^*\|$$

Now suppose $(v^* \star w^*)^2 = \|v^*\| \|w^*\|$. This is equal to saying $(p \star q)^2 = \|p\| \|q\|$, then

$$c^2d^2(p \star q)^2 = c^2d^2 \|p\| \|q\| = (c^2 \|p\|)(d^2 \|q\|)$$

which, with equation (2.2), means

$$((cd)(p \star q))^2 = \|v\| \|w\| \tag{2.4}$$

We can rewrite the left part of the above equation as follows

$$\begin{aligned} [(cd)(p \star q)]^2 &= [d((cp) \star q)]^2 \\ &= [(cp) \star (dq)]^2 \\ &= [v \star w]^2 \end{aligned}$$

with which equation (2.4) becomes

$$(v \star w)^2 = \|v\| \|w\|$$

■

THEOREM **2.9** Let v be a vector that is not equal to the zero vector, and let q be its base vector, i.e. $q = v^*$. Then q is a fundamental vector.

PROOF

Let the positive integer d be defined by $d = \text{gd}(v)$, then $q = v/d$. Suppose q is not a fundamental vector. Then its greatest divisor is not equal to one, i.e. $\text{gd}(q) \neq 1$. Thus $\text{gd}(q) = c$ with $c > 1$ and c divides q. So there is a vector p such that $q = cp$. But $v = dq$. So we get $v = d(cp) = (dc)p$. So the positive integer $k = (dc)$ is a divisor of v. We found earlier that $c > 1$, which means $cd > d$, and which finally means that $k > d$. Because d is the greatest divisor of v, we have for all divisors h of v that $h \leq d$. So, because we have found that k divides v, we must have

$$k \leq d \tag{2.5}$$

But we found earlier that $k > d$. This contradicts with eq. (2.5), so we must conclude that q is a fundamental vector. ■

THEOREM **2.10** Let v and w be vectors of equal orientation and with the same length. Let $q = \min(v - w, v + w)$. If $(v \star w)^2 = \|v\| \, \|w\|$ then $(q \star w)^2 = \|q\| \, \|w\|$.

PROOF

$$q = v - w \quad \text{or} \quad q = v + w$$

Suppose $q = v - w$. Then

$$\begin{aligned} ((v - w) \star w)^2 &= (v \star w - w \star w)^2 \\ &= (v \star w)^2 - 2(v \star w)(w \star w) + (w \star w)^2 \\ &= \|v\| \, \|w\| - 2(v \star w) \, \|w\| + \|w\|^2 \end{aligned}$$

The last equation, because $(v \star w)^2 = \|v\| \, \|w\|$ as was given in the theorem. We continue

$$\begin{aligned} ((v - w) \star w)^2 &= \|v\| \, \|w\| - 2(v \star w) \, \|w\| + \|w\|^2 \\ &= \{\|v\| - 2(v \star w) + \|w\|\} \, \|w\| \\ &= \{v \star v - v \star w - v \star w + w \star w\} \, \|w\| \\ &= \{v \star v - v \star w - w \star v + w \star w\} \, \|w\| \\ &= \{(v - w) \star (v - w)\} \, \|w\| \\ &= \|v - w\| \, \|w\| \end{aligned}$$

So we have $((v - w) \star w)^2 = \|v - w\| \, \|w\|$. With $q = v - w$ it follows that

$$(q \star w)^2 = \|q\| \, \|w\|$$

Now suppose $q = v + w$, then

$$\begin{aligned} ((v + w) \star w)^2 &= (v \star w + w \star w)^2 \\ &= (v \star w)^2 + 2(v \star w)(w \star w) + (w \star w)^2 \\ &= \|v\| \, \|w\| + 2(v \star w) \, \|w\| + \|w\|^2 \end{aligned}$$

The last equation by $(v \star w)^2 = \|v\| \, \|w\|$ as given in the theorem. We continue

$$\begin{aligned} ((v + w) \star w)^2 &= \|v\| \, \|w\| + 2(v \star w) \, \|w\| + \|w\|^2 \\ &= \{\|v\| + 2(v \star w) + \|w\|\} \, \|w\| \\ &= \{\|v\| + v \star w + v \star w + \|w\|\} \, \|w\| \\ &= \{\|v\| + v \star w + w \star v + \|w\|\} \, \|w\| \\ &= \{v \star v + v \star w + w \star v + w \star w\} \, \|w\| \\ &= \{(v + w) \star (v + w)\} \, \|w\| \\ &= \|v + w\| \, \|w\| \end{aligned}$$

So we have $((v + w) \star w)^2 = \|v + w\| \, \|w\|$. With $q = v + w$ it follows that $(q \star w)^2 = \|q\| \, \|w\|$. With which the theorem is proved. ■

THEOREM **2.11** Let v and w be vectors of equal length and with the same orientation. If $(v \star w)^2 = \|v\| \, \|w\|$ and $\|v\| = \|w\|$, then $\min(v - w, v + w) = 0$.

PROOF

We have $(v \star w)^2 = \|v\| \, \|w\|$. With $\|v\| = \|w\|$ it follows that

$$(v \star w)^2 = \|v\| \, \|w\| = \|v\| \, \|v\| = \|v\|^2$$

From which it follows that

$$(v \star w) = \|v\| \quad \text{or} \quad (v \star w) = - \|v\|$$

Now let $q = \min(v - w, v + w)$, then $q = v - w$ or $q = v + w$.

Suppose $q = v - w$, then

$$\begin{aligned} \|q\| &= \|v - w\| \\ &= (v - w) \star (v - w) \\ &= v \star v - v \star w - w \star v + w \star w \\ &= v \star v - v \star w - v \star w + w \star w \\ &= \|v\| - 2(v \star w) + \|w\| \\ &= \|v\| - 2(v \star w) + \|v\| \end{aligned} \tag{2.6}$$

We have $(v \star w) = \|v\|$ or $(v \star w) = - \|v\|$.

Now suppose $(v \star w) = \|v\|$, then it follows that

$$-2(v \star w) = -2 \|v\| \tag{2.7}$$

So from eq. (2.6) en eq. (2.7) we get

$$\begin{aligned} \|q\| &= \|v\| - 2(v \star w) + \|v\| \\ &= \|v\| - 2 \|v\| + \|v\| \\ &= 2 \|v\| - 2 \|v\| = (2 - 2) \|v\| = (0) \|v\| = 0 \end{aligned}$$

Now suppose $(v \star w) = - \|v\|$, from which it follows that

$$2(v \star w) = -2 \|v\| \tag{2.8}$$

Because $q = (v - w)$ and because of the properties of the min function we have

$$\|q\| = \|v - w\| \leq \|v + w\| \tag{2.9}$$

Now

$$\begin{aligned} \|v + w\| &= (v + w) \star (v + w) \\ &= v \star v + v \star w + w \star v + w \star w \\ &= v \star v + v \star w + v \star w + w \star w \\ &= \|v\| + 2(v \star w) + \|w\| \end{aligned} \tag{2.10}$$

So from eq. (2.9) with eq. (2.10) we have

$$\begin{aligned}
\|q\| &= \|v-w\| \leq \|v+w\| \\
&= \|v\| + 2(v \star w) + \|w\| \\
&= \|v\| + 2(v \star w) + \|v\| \,(\text{ because of } \|v\| = \|w\|\,) \\
&= \|v\| - 2\,\|v\| + \|v\| \\
&= 2\,\|v\| - 2\,\|v\| = (2-2)\,\|v\| = 0\,\|v\| = 0
\end{aligned}$$

So we have $\|q\| \leq 0$. But for all vectors we have by property **(P 2)** of theorem 2.5 that the norm of a vector is nonnegative. So from $\|q\| \leq 0$ and $\|q\| \geq 0$, we conclude that $\|q\| = 0$.

Now we have for both cases that $\|q\| = 0$, so we conclude $\|q\| = 0$. From $\|q\| = 0$ it follows by property **(P 3)** of theorem 2.5 that $q = 0$, which is equivalent to $\min(v-w, v+w) = 0$.

We are now ready to prove the theorem for the case where $q = v + w$. Then again we have that $(v \star w) = \|v\|$ or $(v \star w) = -\|v\|$. Suppose $(v \star w) = \|v\|$. Then it follows that

$$-2(v \star w) = -2\,\|v\| \tag{2.11}$$

Because $q = v + w$ and because of the properties of the min function we have

$$\begin{aligned}
\|q\| &= \|v+w\| \leq \|v-w\| \\
&= (v-w) \star (v-w) \\
&= v \star v - v \star w - w \star v + w \star w \\
&= v \star v - v \star w - v \star w + w \star w \\
&= \|v\| - 2(v \star w) + \|w\| \\
&= \|v\| - 2(v \star w) + \|v\| \\
&= \|v\| - 2\,\|v\| + \|v\| \quad \text{(by eq.2.11)} \\
&= 2\,\|v\| - 2\,\|v\| \\
&= 0
\end{aligned}$$

So we have $\|q\| \leq 0$, and we have $\|q\| \geq 0$ by property **(P 2)** of theorem 2.5. We conclude that $\|q\| = 0$ by property **(P 3)** of theorem 2.5.

Now suppose $(v \star w) = -\|v\|$, then it follows that

$$2(v \star w) = -2\,\|v\| \tag{2.12}$$

Now

$$\begin{aligned}
\|q\| &= \|v+w\| \\
&= (v+w) \star (v+w) \\
&= v \star v + v \star w + w \star v + w \star w
\end{aligned}$$

$$\begin{aligned} &= v \star v + v \star w + v \star w + w \star w \\ &= \|v\| + 2(v \star w) + \|w\| \\ &= \|v\| + 2(v \star w) + \|v\| \\ &= \|v\| - 2\|v\| + \|v\| \quad \text{(by eq.2.12)} \\ &= 2\|v\| - 2\|v\| \\ &= 0 \end{aligned}$$

So we have $\|q\| = 0$. From property **(P 3)** of theorem 2.5 it follows that $q = 0$, which is equivalent to $\min(v - w, v + w) = 0$. Which proves the theorem. ■

COROLLARY **2.12** Let v and w be vectors of equal length and with the same orientation. If $(v \star w)^2 = \|v\| \|w\|$ and $\|v\| = \|w\|$, then $v = w$ or $v = -w$.

PROOF

From theorem 2.11 it follows that $\min(v - w, v + w) = 0$. From the definition of the min function it follows that $\min(v - w, v + w) = v - w$ or $\min(v - w, v + w) = v + w$. So $v - w = 0$ or $v + w = 0$, from which it follows that $v = w$ or $v = -w$ as required. ■

THEOREM **2.13** Let v and w be vectors of equal length and orientation and both not equal to zero. Let $q = \min(v - w, v + w)$ and let v and w be such that $\|v\| > \|w\|$. Then

$$\text{If } (v \star w)^2 = \|v\| \|w\| \text{ then } \|q\| < \|v\|$$

PROOF

With $\|v\| > \|w\|$ we have $(v \star w)^2 = \|v\| \|w\| > \|w\|^2$. From which it follows that

$$(v \star w) > \|w\| \quad \text{or} \quad (v \star w) < -\|w\|$$

We have $q = v - w$ or $q = v + w$. Suppose $q = v - w$. Now we have $(v \star w) > \|w\|$ or $(v \star w) < -\|w\|$. Suppose

$$(v \star w) > \|w\| \tag{2.13}$$

then

$$\begin{aligned} \|q\| &= \|v - w\| \\ &= (v - w) \star (v - w) \\ &= v \star v - v \star w - w \star v + w \star w \\ &= \|v\| - 2(v \star w) + \|w\| \end{aligned}$$

From eq. (2.13) we have $-2(v \star w) < -2\|w\|$. It follows that

$$\begin{aligned} \|q\| &= \|v\| - 2(v \star w) + \|w\| < \|v\| - 2\|w\| + \|w\| \\ &= \|v\| - \|w\| \end{aligned}$$

We have $\|w\| \geq 0$ if and only if $-\|w\| \leq 0$ if and only if $\|v\| - \|w\| \leq \|v\|$. So we have $\|q\| < \|v\| - \|w\| \leq \|v\|$, and so $\|q\| < \|v\|$.

Now, suppose $(v \star w) < -\|w\|$. Because $q = v - w$, we have from the definition of the min function that

$$\|v - w\| \leq \|v + w\|$$

and

$$\begin{aligned} \|v + w\| &= (v + w) \star (v + w) \\ &= v \star v + v \star w + w \star v + w \star w \\ &= v \star v + v \star w + v \star w + w \star w \\ &= \|v\| + 2(v \star w) + \|w\| \end{aligned}$$

From $(v \star w) < -\|w\|$ it follows that $2(v \star w) < -2\|w\|$. We have

$$\begin{aligned} \|q\| &= \|v - w\| \leq \|v + w\| \\ &= \|v\| + 2(v \star w) + \|w\| < \|v\| - 2\|w\| + \|w\| \\ &= \|v\| - \|w\| \leq \|v\| \end{aligned}$$

and so $\|q\| < \|v\|$.

Now suppose $q = v + w$, then

$$\begin{aligned} \|q\| &= \|v + w\| \\ &= (v + w) \star (v + w) \\ &= v \star v + v \star w + w \star v + w \star w \\ &= v \star v + v \star w + v \star w + w \star w \\ &= \|v\| + 2(v \star w) + \|w\| \end{aligned}$$

Now, again we have $(v \star w) > \|w\|$ or $(v \star w) < -\|w\|$. Suppose $(v \star w) > \|w\|$, then because $q = v + w$ and because of the definition of the min function we have

$$\|v + w\| \leq \|v - w\|$$

We have

$$\begin{aligned} \|v - w\| &= (v - w) \star (v - w) \\ &= v \star v - v \star w - w \star v + w \star w \\ &= v \star v - v \star w - v \star w + w \star w \\ &= \|v\| - 2(v \star w) + \|w\| \end{aligned}$$

So we have $\|q\| = \|v + w\| \leq \|v - w\| = \|v\| - 2(v \star w) + \|w\|$. We have from $(v \star w) > \|w\|$ that $-2(v \star w) < -2\|w\|$. So it follows that

$$\begin{aligned} \|q\| &\leq \|v\| - 2(v \star w) + \|w\| \\ &< \|v\| - 2\|w\| + \|w\| \\ &= \|v\| - \|w\| \\ &\leq \|v\| \end{aligned}$$

and so $\|q\| < \|v\|$.

Now suppose $(v \star w) < -\|w\|$ then we have

$$2(v \star w) < -2\|w\|$$

So we have

$$\begin{aligned} \|q\| &= \|v\| + 2(v \star w) + \|w\| \\ &< \|v\| - 2\|w\| + \|w\| \\ &= \|v\| - \|w\| \leq \|v\| \end{aligned}$$

and so $\|q\| < \|v\|$.

So we have $\|q\| < \|v\|$ in both situations and that proves our result, as required. ■

THEOREM **2.14** Let v and w be fundamental vectors. If $(v \star w)^2 = \|v\|\,\|w\|$ then $v = w$ or $v = -w$.

PROOF

The proof is by strong mathematical induction on the norm m of the maximum of the two vectors. The induction begins with $m = 1$, i.e. $\|v\| = 1$ if we assume for the moment, without loss of generality, that $v = \max(v, w)$. Because a fundamental vector cannot be zero, we must have for w as well $\|w\| = 1$, because that is the minimum value the norm of a fundamental vector can have. Remember that a fundamental vector is unequal to zero by definition.

So we have for this case

$$(v \star w)^2 = \|v\|\,\|w\| = 1 \cdot 1 = 1 \tag{2.14}$$

Because $\|v\| = \|w\| = 1$ we have $v_i = -1$ or $v_i = 1$ for some i, and $v_j = 0$ for all $j \neq i$. And the same for w, i.e. $w_{i'} = -1$ or $w_{i'} = 1$ for some i', and $w_{j'} = 0$ for all $j' \neq i'$.

Suppose $i \neq i'$. Then $v_p w_p = 0$ for all $1 \leq p \leq n$, and so we would have

$$\sum_{p=1}^{n} v_p w_p = 0 \tag{2.15}$$

From $(v \star w)^2 = (\sum_{p=1}^{n} v_p w_p)^2 = 1$, we have

$$\sum_{p=1}^{n} v_p w_p = 1 \quad \text{or} \quad \sum_{p=1}^{n} v_p w_p = -1$$

This contradicts with equation (2.15). So we have to conclude that $i = i'$. So we have

$$\begin{aligned} v \star w &= \sum_{p=1}^{n} v_p w_p \\ &= v_i w_i + \sum_{p=1, p\neq i}^{n} v_p w_p \\ &= v_i w_i + 0 \\ &= v_i w_i \end{aligned}$$

And so by equation (2.14) we have $(v_i w_i)^2 = 1$.

Now $v_i w_i = 1$ or $v_i w_i = -1$. In the first case we would have $v_i = w_i = 1$ or $v_i = w_i = -1$ and so we would have

$$v = w \tag{2.16}$$

In the second case, i.e. $v_i w_i = -1$, we would have

$$v_i = 1 \text{ and } w_i = -1 \quad \text{or} \quad v_i = -1 \text{ and } w_i = 1$$

From which it follows that

$$v = -w \tag{2.17}$$

From eq.(2.16) and eq. (2.17) we have

$$v = w \quad \text{or} \quad v = -w$$

as required.

Now suppose the theorem is true for all m for which $m < k$. We will prove that the theorem is true for $m = k$.

Again, as with the case for $m = 1$, we assume without loss of generality, that $v = \max(v, w)$. Suppose $\|\max(v, w)\| = \|v\| = k$, and that $(v \star w)^2 = \|v\| \, \|w\|$.

Let $q = \min(v - w, v + w)$ Let's suppose that $q = 0$. We have $q = v - w$ or $q = v + w$. If $q = v - w$ then $v - w = 0$, which results in $v = w$. If $q = v + w$ then $v + w = 0$, which results in $v = -w$. So we have $v = w$ or $v = -w$ as required.

Now, suppose $q \neq 0$. Suppose

$$\|v\| \neq \|w\| \tag{2.18}$$

Then we have

$$\|v\| = k \quad \text{and} \quad \|w\| < \|v\| = k$$

Now v and w are both fundamental vectors, so we can apply theorem 2.10, which results in

$$(q \star w)^2 = \|q\| \, \|w\| \tag{2.19}$$

Furthermore, by theorem 2.13 we have that

$$\|q\| < \|v\| \tag{2.20}$$

Let the vector p be the base vector of q, i.e.

$$p = q^*$$

It exists because we have assumed that $q \neq 0$. Then by theorem 2.9 p is also a fundamental vector, and by theorem 2.7 and with equation 2.19 we have

$$(p \star w)^2 = \|p\| \, \|w\| \tag{2.21}$$

Now, both p and w are fundamental vectors, and we have $\|p\| \leq \|q\| < \|v\|$, the former by theorem 2.7, and the latter by equation (2.20). So $\|p\| < k$. We already had $\|w\| < \|v\|$, so $\|w\| < k$.

So, with the results obtained above and with equation (2.21) we can apply the inductive hypothesis to p and w to obtain:

$$p = w \quad \text{or} \quad p = -w \tag{2.22}$$

We have by definition that $q = ap$ for some positive integer a.

$$\boxed{\text{suppose } p = w}$$

Then $q = ap = aw$.

If $q = v - w$ then $v - w = aw$ from which it follows that $v = (a + 1)w$. So $a + 1$ divides v, and $a + 1 > 1$, because a is a positive integer. But this contradicts with the fact that v is a fundamental vector, for which its greatest divisor is equal to one.

If $q = v + w$ then $v + w = aw$, from which it follows that $v = (a - 1)w$. If $a = 1$ then $v = 0$, and we have a contradiction because v is a fundamental vector (and fundamental vectors are not equal to the zero vector by definition.)

If $a = 2$ then we have $v = w$. But then $\|v\| = \|w\|$, which contradicts with the fact that $\|v\| \neq \|w\|$, by equation (2.18).

If $a > 2$ then again we have a divisor, i.e. $(a - 1)$, for v which is greater than 1, which contradicts with the fact that v is a fundamental vector for which the greatest divisor is equal to 1.

$$\boxed{\text{suppose } p = -w}$$

Then $q = ap = -aw$.

If $q = v - w$ then $v - w = -aw$ from which it follows that $v = (1 - a)w = (a - 1)(-w)$.

If $a = 1$, we have $v = 0$, which contradicts with the fact that v is a fundamental vector which is by definition not equal to the zero vector.

If $a = 2$ we have $v = (2-1)(-w) = 1(-w) = -w$. But this would mean that $\|v\| = \|w\|$ which contradicts with the fact that $\|v\| \neq \|w\|$, by equation (2.18).

If $a > 2$ we have that $a - 1$ divides v and $a - 1 > 1$, but this contradicts with the fact that v is a fundamental vector for which its greatest divisor is equal to 1.

If $q = v + w$ then $v + w = -aw$, from which it follows that $v = -(a+1)w = (a+1)(-w)$. We have that $a+1$ divides v and $a+1 > 1$. But this contradicts with the fact that v is a fundamental vector for which the greatest divisor is equal to 1.

Now we have reached contradictions for both possibilities: $p = w$ and $p = -w$, so we can conclude that

$$\|v\| = \|w\|$$

We can now immediately apply the corollary to theorem 2.11, which results in

$$v = w \quad \text{or} \quad v = -w$$

With which we have proved the theorem. ■

COROLLARY **2.15** Let v and w be vectors, both not equal to the zero vector, and let them be of equal orientation and let them have the same length. If $(v \star w)^2 = \|v\| \, \|w\|$ then $v^* = w^*$ or $v^* = -w^*$.

PROOF

Suppose $(v \star w)^2 = \|v\| \, \|w\|$, then with theorem 2.8 we get $(v^* \star w^*)^2 = \|v^*\| \, \|w^*\|$. Now according to theorem 2.9 v^* and w^* are both fundamental vectors, so we can apply theorem 2.14 which results in $v^* = w^*$ or $v^* = -w^*$. ■

THEOREM **2.16** Let v and w be vectors, both not equal to the zero vector, and let them be of equal length and let them have the same orientation. If $(v \star w)^2 = \|v\| \, \|w\|$ then there is an integer a and a positive integer c such that

$$av = cw$$

PROOF

Suppose $(v \star w)^2 = \|v\| \, \|w\|$. According to corollary 2.15 we have that $v^* = w^*$ or $v^* = -w^*$. Let k be the greatest divisor of v and let j be the greatest divisor of w, then

$$v = kv^* \quad \text{and} \quad w = jw^*$$

Suppose $v^* = w^*$ (1A), then

$$v = kv^* \tag{2.23}$$

and

$$w = jw^* \tag{2.24}$$

Multiply eq. (2.23) by j and eq. (2.24) by k:

$$jv = kjv^*$$

and

$$kw = kjw^* \overset{(1A)}{=} kjv^*$$

So we have $jv = kw$, with k a positive integer and j an integer.

Now suppose $v^* = -w^*$ (1B), then

$$v = kv^* \tag{2.25}$$

and

$$w = jw^* \tag{2.26}$$

Multiply eq. (2.25) by j and eq. (2.26) by k:

$$jv = kjv^* \tag{2.27}$$

and

$$kw = kjw^* = kj(-v^*) = -(kj)v^* \tag{2.28}$$

Multiply eq. (2.27) by -1

$$-jv = -kjv^*$$

Let $j' = -j$ then we get

$$j'v = -kjv^* \tag{2.29}$$

Now we have by eq. (2.28) and eq. (2.29):

$$j'v = kw$$

So we have an integer j' and a positive integer k such that

$$j'v = kw$$

And so we have proved the theorem. ■

THEOREM **2.17** Let v and w be vectors of equal length and orientation, such that $(v \star w)^2 \neq \|v\| \, \|w\|$. Then there is a vector z with the same orientation and length as x and y, such that $v \star z = 0$ and $w \star z \neq 0$.

PROOF

Let $z = (v \star v)w - (w \star v)v$. Then

$$\begin{aligned} v \star z &= v \star \{(v \star v)w - (w \star v)v\} \\ &= v \star ((v \star v)w) - v \star ((w \star v)v) \\ &= (v \star v)(v \star w) - (w \star v)(v \star v) \\ &= (v \star v)(v \star w) - (v \star v)(v \star w) \\ &= 0 \end{aligned}$$

and

$$\begin{aligned} w \star z &= w \star \{(v \star v)w - (w \star v)v\} \\ &= w \star ((v \star v)w) - w \star ((w \star v)v) \\ &= (v \star v)(w \star w) - (w \star v)(w \star v) \\ &= (v \star v)(w \star w) - (v \star w)(v \star w) \\ &= \|v\| \, \|w\| - (v \star w)^2 \\ &\neq 0 \quad \text{(which was given in the theorem.)} \end{aligned}$$

Which proves the theorem. ■

3 Linear Equations and Inequalities

CHAPTER

3.1 Matrices

DEFINITION **3.1** An $m \times n$ *matrix* A with entries from the set of integers $\mathbb{Z}$ is a rectangular array of the form

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

where each entry a_{ij} $(1 \leq i \leq m, 1 \leq j \leq n)$ is an integer. The entries a_{ij} are called the *elements* of the matrix. The elements $a_{i1}, a_{i2}, \dots, a_{in}$ compose the ith *row* of the matrix, and the elements $a_{1j}, a_{2j}, \dots, a_{mj}$ compose the jth *column* of the matrix. The element that lies in row i and in column j is denoted A_{ij}. An element A_{ij} with $i = j$ is called a *diagonal* element. When $m = n$ we call the matrix A *square*. We call the matrix with all its elements equal to zero the *zero matrix* and is denoted by ZM. □

Of course matrices with entries from other sets are also possible, but these are not considered in this thesis. From now on we shall simply use the phrase matrix without mentioning that the elements are from the set of integers and it is assumed that the reader is aware that a matrix with entries from the set of integers is implied.

DEFINITION **3.2** Two $m \times n$ matrices A and B are called *equal*, denoted $A = B$, if all their corresponding elements are equal, that is, $A_{ij} = B_{ij}$ for $1 \leq i \leq m$ and $1 \leq j \leq n$. Note that two matrices that differ in their number of rows or in their number of columns are not equal. Two matrices A and B that are not equal are said to be unequal, denoted $A \neq B$. □

DEFINITION **3.3** Let A be an $m \times n$ matrix, then we define the column vectors and row vectors of A as follows

1. Let the integer j be defined such that $1 \leq j \leq n$. Then the j^{th} column vector of A, denoted by $\mathsf{I}A\mathsf{I}_j$, is the column vector obtained as follows
$$\left(\mathsf{I}A\mathsf{I}_j\right)_i = A_{ij}, \text{ for } 1 \leq i \leq m \ ,$$
2. Let the integer i be defined such that $1 \leq i \leq m$, then the i^{th} row vector of A, denoted by $\underline{\overline{A}}_i$, is the row vector obtained as follows
$$\left(\underline{\overline{A}}_i\right)_j = A_{ij}, \text{ for } 1 \leq j \leq n \ .$$

□

3.2 Operations on Matrices

DEFINITION **3.4** Let A and B be two $m \times n$ matrices. Then we define the addition of these two matrices, denoted $A + B$, as
$$(A + B)_{ij} = A_{ij} + B_{ij} \text{ for } 1 \leq i \leq m \text{ and } 1 \leq j \leq n.$$

□

DEFINITION **3.5** Let A be an $m \times p$ matrix and B be a $p \times n$ matrix. Then we define the product of these matrices, denoted AB, as
$$(AB)_{ij} = \sum_{x=1}^{p} A_{ix} \cdot B_{xj} \text{ for } 1 \leq i \leq m, \text{ and } 1 \leq j \leq n.$$

□

DEFINITION **3.6** Let z be an integer and let A be the $m \times n$ matrix. Then we define the product of this scalar and matrix, denoted zA, as
$$(zA)_{ij} = z \cdot A_{ij} \text{ for } 1 \leq i \leq m, \text{ and } 1 \leq j \leq n.$$

□

3.3 Matrices and Vectors

DEFINITION **3.7** Let A be an $m \times n$ matrix, let $w = (\ w_1 \quad w_2 \quad \dots \quad w_m\ )$ be a row vector of length m, and let $v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$ be a column vector of length n. Then we define

1. Multiplication between w and A, denoted as wA, as

$$wA = \sum_{i=1}^{m} w_i \underline{A}_i$$

2. Multiplication between v and A, denoted as Av, as

$$Av = \sum_{i=1}^{n} v_i |A|_i$$

□

THEOREM **3.1** Let A be an $m \times n$ matrix, let x be a row vector of length m, and let y be a column vector of length n. Then we have the following properties for multiplication of a matrix with row and column vectors.

(P 1) $\quad x(Ay) = (xA)y$

PROOF

In the following proof a_{ij} is the entry of A in the i^{th} row and j^{th} column.

$$\begin{aligned} x(Ay) &= x(\sum_{j=1}^{n} |A|_j y_j) \\ &= \sum_{i=1}^{m} x_i (\sum_{j=1}^{n} |A|_j y_j)_i \\ &= \sum_{i=1}^{m} x_i (\sum_{j=1}^{n} a_{ij} y_j) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{n} (x_i a_{ij} y_j) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{n}\sum_{i=1}^{m}(x_i a_{ij} y_j) \\
&= \sum_{j=1}^{n} y_j(\sum_{i=1}^{m} x_i a_{ij}) \\
&= \sum_{j=1}^{n} y_j(\sum_{i=1}^{m} x_i \underline{\overline{A}}_i)_j \\
&= \sum_{j=1}^{n}(\sum_{i=1}^{m} x_i \underline{\overline{A}}_i)_j y_j \\
&= (\sum_{i=1}^{m} x_i \underline{\overline{A}}_i) y \\
&= (xA)y
\end{aligned}
$$

■

DEFINITION **3.8** Let A be an $m \times n$ matrix, let $v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$ be a column vector of length n. Then we define Line Multiplication between A and v, denoted as $A \otimes v$, as

$$\underline{\overline{(A \otimes v)}}_i = \underline{\overline{A}}_i \cdot v_i$$

Furthermore we define $v \otimes A$ as

$$v \otimes A = A \otimes v$$

□

3.4 Linear Equations

THEOREM **3.2 (Solvability of Linear Equations)**
Let A be an $m \times n$ matrix, and let b be a row vector of length n. Then exactly one of the following alternatives holds. Either the equation

$$xA = cb \tag{3.1}$$

has a solution for some positive c, or the equations

$$Ay = 0 \quad , \quad by \neq 0 \tag{3.2}$$

have a solution.

PROOF

The proof is by induction on m, the number of rows of A.

$$\boxed{m = 1}$$

Suppose eq.(3.1) and eq.(3.2) both have a solution, then take the scalar product of eq.(3.1) with y, i.e.

$$xAy = cby \tag{3.3}$$

and take the scalar product of equation (3.2) with x, i.e.

$$xAy = 0 \tag{3.4}$$

From eq.(3.2) and because c is positive, we have

$$cby \neq 0 \tag{3.5}$$

But from eq.(3.3) and eq.(3.4) we have that

$$cby = 0$$

This contradicts with equation (3.5). So we have to conclude that not both eq.(3.1) and eq.(3.2) can have a solution.

Now suppose eq.(3.1) has no solution for any positive c. Then $b \neq 0$, for if $b = 0$ then let $x = 0$ which gives $xA = 0 = c \cdot 0 = cb$ for any positive integer c. So x is a solution to equation (3.1), which contradicts our assumption that equation (3.1) has no solution for any positive c.

Because $m = 1$, the row vector x reduces to an integer and the matrix A is really a row vector. So equation (3.1) reduces to

$$x'a = cb \tag{3.6}$$

where x' is an integer and a is a row vector. Equation (3.6) has no solution for any positive c. If $a = 0$ then let $y = b^{\mathsf{T}}$. Then $ay = 0$ and $by = bb^{\mathsf{T}} \neq 0$. Which would prove the theorem. Now, in the following text let $a \neq 0$.

Suppose $(a \star b)^2 = \|a\| \, \|b\|$. Then according to theorem 2.16 there is an integer x' and a positive c such that they are a solution to eq. (3.6). But this contradicts the fact that there is no solution. So we have to conclude that

$$(a \star b)^2 \neq \|a\| \, \|b\|$$

Now, according to theorem 2.17 there is a columnvector y such that $ay = 0$ and $by \neq 0$. With which we proved the theorem for $m = 1$.

$$\boxed{m = m}$$

Now suppose the theorem is true when the number of rows of A is $m = k - 1$. We will now show the theorem is true when the number of rows of A is $m = k$.

Suppose eq.(3.1) and eq.(3.2) both have a solution, then taking the scalar product of eq.(3.1) with y gives

$$xAy = cby \tag{3.7}$$

and taking the scalar product of equation (3.2) with x gives

$$xAy = 0 \tag{3.8}$$

From eq.(3.2) and because c is positive, we have

$$cby \neq 0 \tag{3.9}$$

But from eq.(3.7) and eq.(3.8) we have that

$$cby = 0$$

This contradicts with eq. (3.9). So we have to conclude that not both eq.(3.1) and eq.(3.2) can have a solution.

Assuming now that eq.(3.1) has no solution for any positive c, we shall show that eq. (3.2) has a solution.

Suppose $x \overset{\vdash}{A_k} = cb$ has a solution for some positive c, then construct a new row vector x' where

$$\begin{aligned} x'_i &= x_i \quad \text{for } 1 \leq i < k, \text{ and} \\ x'_k &= 0 \end{aligned}$$

Then

$$\begin{aligned} x'A &= \sum_{i=1}^{k} x'_i a_i \\ &= \sum_{i=1}^{k-1} x'_i a_i + x'_k a_k \\ &= \sum_{i=1}^{k-1} x_i a_i + 0 \cdot a_k \\ &= \sum_{i=1}^{k-1} x_i a_i = x \cdot \overset{\vdash}{A_k} = cb \end{aligned}$$

So A has a solution x' for some positive c. But this contradicts our assumption earlier that A has no solution for any positive c. So we have to conclude that the equation

$$x \overset{\vdash}{A_k} = cb \tag{3.10}$$

does not have a solution for any positive c.

We can now apply the theorem to eq. (3.10) by inductive hypothesis. Remember that the number of rows of $\overset{\vdash}{A}_k$ is $k-1$. So by inductive hypothesis, there exists a y_1, such that

$$\begin{aligned} \overset{\vdash}{A}_k\, y_1 &= 0 && (3.11)\\ \text{and} \quad by_1 &\neq 0 && (3.12) \end{aligned}$$

If also $\underline{\overline{A}}_k y_1 = 0$ then y_1 satisfies eq.(3.2) and the theorem is proved. If, however, $\underline{\overline{A}}_k y_1 \neq 0$, then let

$$\begin{aligned} \overline{a}_i &= (a_i y_1)a_k - (a_k y_1)a_i \quad \text{for } 1 \leq i < k \\ \overline{b} &= (by_1)a_k - (a_k y_1)b \end{aligned} \tag{3.13}$$

Now the equation

$$\sum_{i=1}^{k-1} \overline{\xi}_i \overline{a}_i = \overline{c}\overline{b} \tag{3.14}$$

can have no solution for some positive $\overline{c}$, for if so, substituting eq.(3.13) in eq.(3.14) and rewriting gives

$$\sum_{i=1}^{k-1} \overline{\xi_i} \left\{(a_i y_1)a_k - (a_k y_1)a_i\right\} = \overline{c}\left\{(by_1)a_k - (a_k y_1)b\right\}$$

which is equal to

$$\sum_{i=1}^{k-1} \overline{\xi_i}(a_i y_1)a_k - \overline{c}(by_1)a_k - \sum_{i=1}^{k-1} \overline{\xi_i}(a_k y_1)a_i = -\overline{c}(a_k y_1)b$$

which is equal to

$$\left[\sum_{i=1}^{k-1} \overline{\xi_i}(a_i y_1) - \overline{c}(by_1)\right] a_k - \sum_{i=1}^{k-1} \overline{\xi_i}(a_k y_1)a_i = -\overline{c}(a_k y_1)b \tag{3.15}$$

Now according to eq. (3.11) we have that $a_i y_1 = 0$ for all $1 \leq i \leq k-1$. So this results in

$$\sum_{i=1}^{k-1} \overline{\xi}_i(a_i y_1) = \sum_{i=1}^{k-1} \overline{\xi}_i \cdot 0 = \sum_{i=1}^{k-1} 0 = 0$$

So equation (3.15) reduces to:

$$\left[-\overline{c}(by_1)\right] a_k - \sum_{i=1}^{k-1} \overline{\xi}_i(a_k y_1)a_i = -\overline{c}(a_k y_1)b \tag{3.16}$$

Now according to eq. (3.12) we have

$$(by_1) \neq 0$$

Because $\overline{c} > 0$ the above equation leads to

$$\overline{c}(by_1) \neq 0$$

and finally, by multiplying by -1, we get

$$-\overline{c}(by_1) \neq 0 \tag{3.17}$$

If we let $d = -\overline{c}(by_1)$, then from eq. (3.17) we know that the integer d is not equal to zero, and equation 3.16 becomes

$$da_k - \sum_{i=1}^{k-1} \overline{\xi}_i(a_k y_1)a_i = -\overline{c}(a_k y_1)b \tag{3.18}$$

Now, because $a_k y_1 \neq 0$, which we assumed earlier, we either have

$$a_k y_1 < 0, \quad \text{or} \quad a_k y_1 > 0$$

If $a_k y_1 < 0$ then, because $\overline{c} > 0$, we have

$$\overline{c}(a_k y_1) < 0$$

and multiplying this with -1 gives

$$-\overline{c}(a_k y_1) > 0$$

If we let $c = -\overline{c}(a_k y_1)$ then we know that c is a positive integer. If we further let $x_k = d$ and $x_i = -\overline{\xi}_i(a_k y_1)$ for $1 \leq i \leq k-1$, then we have found a vector x and a positive integer c such that they are a solution to equation (3.1).

If $a_k y_1 > 0$ then, because $\overline{c} > 0$ we have

$$\overline{c}(a_k y_1) > 0 \tag{3.19}$$

If we now multiply eq.(3.18) by -1 we get

$$-da_k + \sum_{i=1}^{k-1} \overline{\xi}_i(a_k y_1)a_i = \overline{c}(a_k y_1)b \tag{3.20}$$

Now, if we let $c = \overline{c}(a_k y_1)$ we know that c is a positive integer by eq. (3.19). If we further let $x_k = -d$ and let $x_i = \overline{\xi}_i(a_k y_1)$ for $1 \leq i \leq k-1$, then we have found a vector x and a positive integer c such that they are a solution to eq.(3.1).

So there is a solution to eq.(3.1), for some positive c. But this contradicts our assumption that there is no solution to eq.(3.1) for any positive c. So we have to conclude that eq.(3.14) has no solution $\overline{\xi}$ for any positive $\overline{c}$.

We can now again apply the inductive hypothesis, but now to eq.(3.14). So there is a vector $\overline{y}$ such that $\overline{a_i}\,\overline{y} = 0$ for $1 \leq i < k$ and $\overline{b}\overline{y} \neq 0$. Now let

$$y = (a_k\overline{y})y_1 - (a_k y_1)\overline{y}$$

we have

$$\begin{aligned} a_i y &= (a_k\overline{y})(a_i y_1) - (a_k y_1)(a_i\overline{y}) \\ &= [(a_i y_1)a_k - (a_k y_1)a_i]\overline{y} \\ &= \overline{a}_i\,\overline{y} = 0 \quad \text{for } 1 \leq i \leq k-1 \end{aligned}$$

and

$$\begin{aligned} by &= (a_k\overline{y})(by_1) - (a_k y_1)(b\overline{y}) \\ &= [(by_1)a_k - (a_k y_1)b]\overline{y} \\ &= \overline{b}\overline{y} \neq 0 \end{aligned}$$

and finally

$$\begin{aligned} a_k y &= (a_k\overline{y})(a_k y_1) - (a_k y_1)(a_k\overline{y}) \\ &= 0 \end{aligned}$$

So that y satisfies eq.(3.2) and the theorem is proved. ■

COROLLARY **3.3** Let A be an $m \times n$ matrix. If the equation

$$xA = cb$$

has no solution for any positive c, then the equations

$$Ay = 0\ , \quad by < 0 \tag{3.21}$$

have a solution.

PROOF

Suppose $xA = cb$ has no solution for any positive c, then we can apply theorem 3.2. There is a y such that

$$Ay = 0 \quad \text{and} \quad by \neq 0$$

Now $by < 0$ or $by > 0$. If $by < 0$ then we have found a suitable y. If $by > 0$ then $-by < 0$. Now take $y' = -y$, then we get

$$Ay' = A(-y) = -Ay = -(0) = 0$$

and

$$by' = b(-y) = -by < 0$$

So there is a y such that it is a solution to eq.(3.21). ■

THEOREM **3.4 (nonnegative solutions of linear equations)**
Let A be an $m \times n$ matrix. Let x and b be rowvectors both of length n, and y is a columnvector of length n and c is a positive integer. Then exactly one of the following alternatives holds. Either the equation

$$xA = cb$$

has a nonnegative solution x for some positive c, or the inequalities

$$Ay \geq 0 \quad , \quad by < 0$$

have a solution.

PROOF

The proof is by induction on m, the number of rows of A.

$$\boxed{m = 1}$$

Suppose eq.(3.4) and eq.(3.4) both have a solution, then taking the scalar product of eq.(3.4) with y gives

$$xAy = cby$$

and taking the scalar product of inequality (3.4) with x gives

$$xAy \geq 0$$

since x is nonnegative. But

$$by < 0$$

and so, since $c > 0$

$$cby < 0$$

With eq.(3.4) it follows that

$$xAy < 0$$

which contradicts with eq(3.4).

Assuming now that (1) has no nonnegative solutions for any positive c, then either eq.(3.4) has no solutions at all, or the only solutions are negative. (Remember that $m = 1$, so ξ is really an integer.) We shall show that (2) has a solution, but first we will show that $b \neq 0$. Suppose that $b = 0$, then $x = 0$ is a nonnegative solution, because $xA = 0 = c \cdot 0 = c \cdot b$, for any positive c. But we assumed earlier that there are no nonnegative solutions for any positive intger c. So we have to conclude that b is indeed not equal to the zero vector.

Now, if eq.(3.4) has no solution at all, no matter what positive c, then by corollary 3.3 there exists a y such that $Ay = 0$ and $by < 0$; hence y satisfies eq.(3.4).

Now, suppose that eq.(3.4) has a solution ξ for a positive integer c that is not nonnegative. Then (1) becomes

$$\xi a_1 = cb$$

where ξ has the property $\xi < 0$ and $c > 0$. Then let $y = -b^{\mathsf{T}}$ and note that $by = b \cdot (-b^{\mathsf{T}}) = -(b \cdot b^{\mathsf{T}}) = -(b \star b) = -\|b\| < 0$, the last because $b \neq 0$. Further we have $\xi a_1 y = \xi a_1(-b^{\mathsf{T}}) = cb(-b^{\mathsf{T}}) = c(-\|b\|) = -c\|b\|$. Because $\xi < 0$ we can write $\xi a_1 y$ as follows

$$\xi a_1 y = -|\xi| a_1 y$$

Together with the result obtained above it follows that

$$-|\xi| a_1 y = -c\|b\|$$

which means that

$$|\xi| a_1 y = c\|b\|$$

Because $|\xi| > 0$ and $c\|b\| > 0$ it follows that

$$a_1 y > 0$$

so that y is a solution of eq.(3.4).

$$\boxed{m = m}$$

Now suppose the theorem is true when the number of rows of A is less than m. We will now show the theorem is true when the number of rows of A is m.

Suppose eq.(3.4) and eq.(3.4) both have a solution, then taking the scalar product of eq.(3.4) with y gives

$$xAy = cby$$

and taking the scalar product of inequality (3.4) with x gives

$$xAy \geq 0$$

since x is nonnegative. But

$$by < 0$$

and so, since $c > 0$,

$$cby < 0$$

With eq.(3.4) it follows that

$$xAy < 0$$

which contradicts with eq.(3.4).

Assuming now that eq.(3.4) has no nonnegative solution for any positive c, we shall show that eq.(3.4) has a solution.

Suppose $x \overset{\vdash}{A}_m = cb$ has a solution where both x is nonnegative and c is positive, then construct a new row vector x' where

$$\begin{aligned} x'_i &= x_i \quad \text{for } 1 \leq i < m, \text{ and} \\ x'_m &= 0 \end{aligned}$$

Then, if we let $a_i = \underline{A}_i$ for $1 \leq i \leq m$, we get

$$\begin{aligned} x'A &= \sum_{i=1}^{m} x'_i a_i \\ &= \sum_{i=1}^{m-1} x'_i a_i + x'_m a_m \\ &= \sum_{i=1}^{m-1} x_i a_i + 0 \cdot a_m \\ &= \sum_{i=1}^{m-1} x_i a_i \\ &= x \overset{\vdash}{A}_m = cb \end{aligned}$$

So A has a nonnegative solution x' for some positive c. But this contradicts our assumption earlier that A has no nonnegative solution x for any positive integer c. So we conclude that the equation

$$x \overset{\vdash}{A}_m = cb$$

does not have a nonnegative solution for any positive integer c. We can now apply the theorem to eq.(3.4) by inductive hypothesis, because remember that the number of rows of $\overset{\vdash}{A}_m$ is less than m.

So, by inductive hypothesis, there exists a y_1 such that

$$\begin{aligned} &\overset{\vdash}{A}_m \, y_1 \geq 0 \\ \text{and} \quad &by_1 < 0 \end{aligned} \tag{3.22}$$

If also $\underline{A}_m y_1 \geq 0$ then y_1 satisfies eq.(3.4) and the theorem is proved. If, however, $\underline{A}_m y_1 < 0$, then let

$$\begin{aligned} \overline{a}_i &= (a_i y_1) a_m - (a_m y_1) a_i \text{ for } 1 \leq i < m \\ \overline{b} &= (by_1) a_m - (a_m y_1) b \end{aligned} \tag{3.23}$$

where again $a_i = \underline{A}_i$ for $1 \leq i \leq m$. Now the equation

$$\sum_{i=1}^{m-1} \overline{\xi}_i \overline{a}_i = \overline{c}\overline{b} \tag{3.24}$$

can have no nonnegative solution ξ for some positive $\overline{c}$, for if so, substituting eq.(3.23) into eq.(**??**) and rewriting gives

$$\sum_{i=1}^{m-1} \overline{\xi}_i \{(a_i y_1)a_m - (a_m y_1)a_i\} = \overline{c}((by_1)a_m - (a_m y_1)b)$$

which is equal to

$$\sum_{i=1}^{m-1} \overline{\xi}_i (a_i y_1)a_m - \overline{c}(by_1)a_m - \sum_{i=1}^{m-1} \overline{\xi}_i (a_m y_1)a_i = -\overline{c}(a_m y_1)b$$

which is equal to

$$\left[\sum_{i=1}^{m-1} \overline{\xi}_i (a_i y_1) - \overline{c}(by_1)\right] a_m - \sum_{i=1}^{m-1} \overline{\xi}_i (a_m y_1)a_i = -\overline{c}(a_m y_1)b$$

We know that $a_i y_1 \geq 0$ for $1 \leq i \leq m$ by eq.(**??**). Furthermore we know that $\overline{\xi}_i \geq 0$ for $1 \leq i \leq< m$, so $\overline{\xi}_i (a_i y_1) \geq 0$ for $1 \leq i < m$.

Further $by_1 < 0$ also by eq.(**??**), and $\overline{c} > 0$, so $-\overline{c} < 0$. Now $-\overline{c}(by_1) > 0$. So it follows that the term

$$\sum_{i=1}^{m-1} \overline{\xi}_i (a_i y_1) - \overline{c}(by_1)$$

is a positive integer.

Next, we know that $a_m y_1 < 0$, because that is what we assumed. Further we already know that $\overline{\xi}_i \geq 0$, so the product $\overline{\xi}_i (a_m y_1) \leq 0$ and so

$$-\overline{\xi}_i (a_m y_1) \geq 0$$

Finally, because $a_m y_1 < 0$ and $\overline{c} > 0$ we have

$$\begin{aligned} \overline{c}(a_m y_1) &< 0, \quad \text{and} \\ -\overline{c}(a_m y_1) &> 0 \end{aligned}$$

Now, if we let

$$x_i = -\overline{c}(by_1) \quad \text{for } 1 \leq i < m$$

and

$$x_m = \left[\sum_{i=1}^{m-1} \overline{\xi}_i (a_i y_1) - \overline{c}(by_1)\right]$$

and finally

$$c = -\overline{c}(a_m y_1)$$

then, together with the results obtained above that all these terms are either positive or zero, we recognize that we have found a nonnegative solution x of equation (3.4)

for a positive integer c. But this contradicts our assumption that there is no nonnegative solution for eq.(3.4) for any positive integer c.

So we must conclude that eq.(**??**) has no nonnegative solution $\overline{\xi}$ for any positive integer $\overline{c}$. We can then again apply the inductive hypothesis, but now to eq.(**??**). So, there is a vector $\overline{y}$ such that $\overline{a}_i\overline{y} \geq 0$ for $1 \leq i < m$ and $\overline{b}\overline{y} < 0$. Now let

$$y = (a_m\overline{y})y_1 - (a_m y_1)\overline{y}$$

we have

$$\begin{aligned} a_i y &= (a_m\overline{y})(a_i y_1) - (a_m y_1)(a_i\overline{y}) \\ &= [(a_i y_1)a_m - (a_m y_1)a_i]\overline{y} \\ &= \overline{a}_i\overline{y} \geq 0 \quad \text{for } 1 \leq i < m \\ by &= (a_m\overline{y})(by_1) - (a_m y_1)(b\overline{y}) \\ &= [(by_1)a_m - (a_m y_1)b]\overline{y} \\ &= \overline{b}\overline{y} < 0 \quad \text{and} \\ a_m y &= (a_m\overline{y})(a_m y_1) - (a_m y_1)(a_m\overline{y}) \\ &= 0 \end{aligned}$$

so that y satisfies eq.(3.4) and the theorem is proved. ■

3.5 Linear Homogeneous Equations

THEOREM **3.5 (semipositive solutions of homogeneous equations)** Let A be an $m \times n$ matrix. Then exactly one of the following alternatives holds.

Either the equation

$$xA = 0$$

has a semipositive solution, or the inequality

$$Ay > 0$$

has a solution.

PROOF

Suppose that both eq.(3.5) and eq.(3.5) have the requested solution, then multiply eq.(3.5) by y on the right, and eq.(3.5) by x on the left, then

$$xAy = 0 \cdot y = 0$$

and

$$xAy > x \cdot 0 = 0$$

by the new theorem in the chapter on vectors.

But this leads to a contradiction, so we have to conclude that not both eq.(3.5) and eq.(3.5) can have a solution.

Now suppose eq.(3.5) has no semipositive solution, then the equations

$$\begin{aligned} \sum_{i=1}^{m} \xi_i a_i &= 0 \\ \sum_{i=1}^{m} \xi_i &= c \end{aligned}$$

where $a_i = \underline{A}_i$ for $1 \leq i \leq m$, have no nonnegative solution for any positive c. For if they do then from $\sum \xi_i = c$, where c is a positive integer, and the fact that $\xi_i \geq 0$ for all $1 \leq i \leq m$ because ξ is nonnegative, we conclude that ξ is in fact semipositive and because $\sum \xi_i a_i = 0$, it follows that ξ is a solution to eq.(3.5) which contradicts the assumption that eq.(3.5) has no semipositive solution x for any positive integer c.

Now we rearrange the above equations a bit. Let the $m \times (n+1)$ matrix A' be defined by

$$A' = (A \ \underline{1})$$

and let the rowvector b' be defined by

$$b' = (\underline{0} \ 1)$$

Then what was stated for the above equations is equivalent to saying that

$$\xi A' = cb'$$

has no nonnegative solution for any positive c.

But from theorem 3.4 it then follows that there is a column vector y and an integer η such that

$$A' \cdot \begin{pmatrix} y \\ \eta \end{pmatrix} \geq 0 \ \text{ and } b' \cdot \begin{pmatrix} y \\ \eta \end{pmatrix} = \underline{0} \cdot y + 1 \cdot \eta = \eta < 0$$

The first equation is equivalent to saying that $a_i y + \eta \geq 0$ for all $1 \leq i \leq m$. From which it follows that

$$a_i y \geq -\eta > 0 \quad \text{for all } 1 \leq i \leq m$$

And so there is a y such that $Ay > 0$. ■

THEOREM **3.6** The following does **not** hold: If there is a column vector y such that $Ay \geq 0$ then there is a column vector x such that $Ax > 0$, for all matrices A.

PROOF

Suppose that the above proposition does hold. We have found earlier in theorem ?? that if there is a column vector x such that $Ax > 0$ then A is not nullable. So we have from this the proposition that if there is a column vector y such that $Ay \geq 0$ then A is not nullable. This is equivalent to the following, if A is nullable then there is not a column vector y such that $Ay \geq 0$.

Now, suppose that we have a matrix A with two rows, where row 1, denoted by a_1, is defined as a mixed vector that has no entries that are zero. The second row of A, denoted by a_2, is defined by $a_2 = -a_1$. Then A is nullable—just add the rows of A. Now, let the column vector y be defined as the all-zero vector, i.e. $y = 0$. Then we have $Ay = A0 = 0 \geq 0$. In other words we have a column vector y such that $Ay \geq 0$. But we also have the proposition derived earlier from our assumption, and repeated here

If A is nullable then there is not a column vector y such that $Ay \geq 0$.

Now we have just derived that our matrix A is nullable, so we can apply the above proposition, i.e. there is a column vector y such that $Ay \geq 0$.

But we found earlier that there is no such column vector, and thus we must conclude that the statement given at the beginning of the theorem does not hold as stated in the theorem. ■

4 Nullability

CHAPTER

4.1 Nullability

DEFINITION **4.1** An $m \times n$ matrix A is called *nullable*, denoted $\mathcal{N}(A)$, if there is a semipositive row vector $v = \begin{pmatrix} v_1 & v_2 & \dots & v_m \end{pmatrix}$ such that $vA = 0$. □

DEFINITION **4.2** Let A be an $m \times n$ matrix. The following operation on the rows of A is called the *elementary row operation*:

1. multiplying any row of A by a positive integer.

□

DEFINITION **4.3** An $n \times n$ elementary matrix is a matrix obtained by performing the elementary row operation on I_n. □

THEOREM **4.1** Let A be an $m \times n$ matrix, and suppose that B is obtained from A by performing the elementary row operation on A. Then there exists an $m \times m$

elementary matrix E such that $B = EA$. In fact E is obtained from I_m by performing the same elementary row operation on I_m as that which was performed on A to obtain B. Conversely, if E is an elementary $m \times m$ matrix, then EA is the matrix obtained from A by performing the same elementary row operation on A as that which produces E from I_m.

PROOF

Let A be the $m \times n$ matrix, and suppose that B is the $m \times n$ matrix obtained from A by multiplying row i of A by c, with $c \in \mathbb{N}$, that is by performing the elementary row operation.
The matrices are

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

and then B becomes

$$B = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ c \cdot a_{i1} & c \cdot a_{i2} & \dots & c \cdot a_{in} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Now, B can be rewritten as

$$\begin{aligned} B &= \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ c \cdot a_{i1} & c \cdot a_{i2} & \dots & c \cdot a_{in} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \\ &= \begin{pmatrix} 0 & & & & & \\ & 0 & & & & \\ & & \ddots & & & \\ & & & c & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} \cdot A + \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} \cdot A \end{aligned}$$

$$
\begin{aligned}
&= \left\{ \begin{pmatrix} 0 & & & & & \\ & 0 & & & & \\ & & \ddots & & & \\ & & & c & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} + \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} \right\} \cdot A \\
&= \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & c & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} \cdot A \\
&= E \cdot A
\end{aligned}
$$

with

$$
E = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & c & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix}.
$$

The matrix E is an elementary matrix, because it is obtainded from I_m by multiplying row i of I_m with c. This is exactly the row operation that was applied to A.

Now, let $B = E \cdot A$, with E an elementary matrix obtained from I_m by multiplying row i of I_m with c. Let us write E as

$$
E = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & c & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix}
$$

Then

$$
B = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & c & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} \cdot A
$$

which can be represented as

$$B = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ c \cdot a_{i1} & c \cdot a_{i2} & \dots & c \cdot a_{in} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

But this matrix can be obtained from A by multiplying row i of A by c, and this is exactly the same row operation as that which was applied to I_m to obtain E. ■

DEFINITION **4.4** Let A be an $m \times n$ matrix. Let the $m \times (n-1)$ matrix, denoted by $\overset{\perp}{A}_q$, be the matrix obtained from A by removing the ith column of A, i.e.

$$\left(\overset{\perp}{A}_q\right)_{ij} = \begin{array}{l} A_{ij} \text{ if } j < q,\ 1 \le i \le m \\ A_{i,j+1} \text{ if } j \ge q,\ 1 \le j \le n-1. \end{array}$$

□

In this thesis we denote the set of non-negative integers by $\mathbb{N}$, i.e. $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, and we call this set the set of *natural numbers*. We often need a subset of the set of natural numbers that consists of a consecutive range of natural numbers. For example $\{4, 5, 6, 7\}$ is such a set and $\{13, 14, 15\}$ is another, but $\{8, 10, 11, 13, 14\}$ is not, because the numbers 9 and 12 are missing from the this set. The following definition makes this concept precise.

DEFINITION **4.5** Let the set of positive integers, denoted by $\mathbb{N}_a^b$, be defined as

$$\mathbb{N}_a^b = \{i \in \mathbb{N} | a \le i \le b\} ,$$

with $a \in \mathbb{N}$ and $b \in \mathbb{N}$, and $a < b$. We call such a set an *interval*. □

DEFINITION **4.6** Let A be an $m \times s$ matrix, and let the kth column vector of A be a paved mixed vector.

1. Let P, N, and Z be sets of natural numbers that are defined as follows:

$$\begin{aligned} P &= \{1 \le j \le m | (\mathbb{A}\mathbb{l}_i)_j > 0\} , \\ N &= \{1 \le j \le m | (\mathbb{A}\mathbb{l}_i)_j < 0\} , \\ \text{and } Z &= \mathbb{N}_1^m \backslash (P \cup N) \end{aligned}$$

2. Define the positive integers p, n, and z as:

$$\begin{aligned} p &= sizeof(P) \ , \\ n &= sizeof(N) \ , \\ z &= sizeof(Z) \end{aligned}$$

3. Let the function $\sigma_P : \mathbb{N}_1^p \to P$ be defined by the following property:

$$\sigma_P(i) < \sigma_P(j) \text{ if } i < j, \text{ for all} i, j \in \mathbb{N}_1^p$$

Let the function $\sigma_N : \mathbb{N}_1^n \to N$ be defined by the following property:

$$\sigma_N(i) < \sigma_N(j) \text{ if } i < j, \text{ for all} i, j \in \mathbb{N}_1^n$$

Let the function $\sigma_Z : \mathbb{N}_1^z \to Z$ be defined by the following property:

$$\sigma_Z(i) < \sigma_Z(j) \text{ if } i < j, \text{ for all} i, j \in \mathbb{N}_1^z$$

Now, the *expansion* of A in column k, denoted by $\mathbf{E}(A)_k$, is the $(n \cdot p + z) \times s$ matrix defined by

1.

$$\underline{\overline{(\mathbf{E}(A)_k)}}_{(i-1)\cdot n+j} = \underline{\overline{A}}_{\sigma_P(i)} + \underline{\overline{A}}_{\sigma_N(j)}, \text{ for } 1 \leq i \leq p \ , \ 1 \leq j \leq n$$

2.

$$\underline{\overline{(\mathbf{E}(A)_k)}}_{p\cdot n+j} = \underline{\overline{A}}_{\sigma_Z(j)}, \text{ for } 1 \leq j \leq z$$

□

THEOREM **4.2** Let A be an $m \times n$ matrix, and let B be the $m \times n$ matrix obtained from A by performing the elementary row operation on A. Then B is nullable if and only if A is nullable.

PROOF

Let row i of A be the row on which the elementary row operation was performed. Then

$$B = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ qa_{k1} & qa_{k2} & \dots & qa_{kn} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

with q a positive natural number, i.e. $p \in \mathbb{N}$, and $p > 0$. With the above representation of B we have

1. $\overline{\underline{A}}_i = \overline{\underline{B}}_i$ for $i = 1, 2, \ldots, m$, and $i \neq k$,
2. $\overline{\underline{B}}_k = q\overline{\underline{A}}_k$

If A is nullable, then there is a semipositive row vector p such that $pA = 0$. Suppose $p_k = 0$, then

$$\begin{aligned}
pA &= \sum_{i=1}^{m} p_i \overline{\underline{A}}_i \\
&= \sum_{i=1, i\neq k}^{m} p_i \overline{\underline{A}}_i + p_k \overline{\underline{A}}_k \\
&= \sum_{i=1, i\neq k}^{m} p_i \overline{\underline{A}}_i + 0\overline{\underline{A}}_k \\
&= \sum_{i=1, i\neq k}^{m} p_i \overline{\underline{A}}_i \\
&= \sum_{i=1, i\neq k}^{m} p_i \overline{\underline{B}}_i \\
&= \sum_{i=1, i\neq k}^{m} p_i \overline{\underline{B}}_i + 0\overline{\underline{B}}_k \\
&= \sum_{i=1, i\neq k}^{m} p_i \overline{\underline{B}}_i + p_k \overline{\underline{B}}_k \\
&= \sum_{i=1}^{m} p_i \overline{\underline{B}}_i \\
&= pB
\end{aligned}$$

Let $t = p$, then we have $tB = pB = pA = 0$, and t is a semipositive row vector. So there is a semipositive rowvector t such that $tB = 0$. Hence B is nullable.

Now, suppose $p_i \neq 0$, then $pA = \sum_{i=1}^{m} p_i \overline{\underline{A}}_i = 0$. If we multiply both sides by q, then $\sum_{i=1}^{m} q \cdot p_i \overline{\underline{A}}_i = 0$. Now,

$$\begin{aligned}
\sum_{i=1}^{m} qp_i \overline{\underline{A}}_i &= \sum_{i=1, i\neq k}^{m} qp_i \overline{\underline{A}}_i + qp_k \overline{\underline{A}}_k \\
&= \sum_{i=1, i\neq k}^{m} qp_i \overline{\underline{B}}_i + p_k \overline{\underline{B}}_k \\
&= \sum_{i=1}^{m} t_i \overline{\underline{B}}_i \\
&= tB
\end{aligned}$$

where we have introduced a new row vector t, defined by

$$t_i = \begin{cases} qp_i & \text{if } i \neq k \\ p_k & \text{if } i = k \end{cases}$$

With $\sum_{i=1}^{m} qp_i\underline{\overline{A}}_i = 0$ it follows from the equality $\sum_{i=1}^{m} qp_i\underline{\overline{A}}_i = tB$ that $tB = 0$.

We now show that t is a semipositive row vector. Let $j \in \mathbb{N}_1^m$, then either $j = k$ or $j \neq k$. If $j = k$ then $tj = pj \geq 0$, and if $j \neq k$ then $t_j = qp_j \geq 0$. So $t_j \geq 0$ for $j = 1, 2, \ldots, m$. We know that p is a semipositive row vector, so there is an $s \in \mathbb{N}_1^m$ such that $p_s > 0$. Now suppose $s \neq k$, then $t_s = qp_s > 0$, and if $s = k$ then $t_s = p_s > 0$. So, there is an $s \in \mathbb{N}_1^m$ such that $t_s > 0$. Hence t is a semipositive row vector, and with the result found earlier we can conclude that there is a semipositive row vector t such that $tB = 0$. From this it follows that B is nullable.

Because either $p_k = 0$ or $p_k \neq 0$, we can conclude from the results obtained above that B is nullable if A is nullable. Conversely, we show next that A is nullable if B is nullable. Suppose B is nullable, then there is a semipositive row vector p such that $pB = 0$.

Suppose $p_i = 0$, then

$$\begin{aligned}
pB &= \sum_{i=1}^{m} p_i\underline{\overline{B}}_i \\
&= \sum_{i=1,i\neq k}^{m} p_i\underline{\overline{B}}_i + p_k\underline{\overline{B}}_k \\
&= \sum_{i=1,i\neq k}^{m} p_i\underline{\overline{B}}_i + 0\underline{\overline{B}}_k \\
&= \sum_{i=1,i\neq k}^{m} p_i\underline{\overline{B}}_i \\
&= \sum_{i=1,i\neq k}^{m} p_i\underline{\overline{A}}_i \\
&= \sum_{i=1,i\neq k}^{m} p_i\underline{\overline{A}}_i + 0\underline{\overline{A}}_k \\
&= \sum_{i=1,i\neq k}^{m} p_i\underline{\overline{A}}_i + p_k\underline{\overline{A}}_k \\
&= \sum_{i=1}^{m} p_i\underline{\overline{A}}_i \\
&= pA
\end{aligned}$$

Let $t = p$, then we have $tA = pA = pB = 0$, and t is a semipositive row vector. So there is a semipositive rowvector t such that $tA = 0$. Hence A is nullable.

Conversely, suppose that $p_k \neq 0$, then

$$\begin{aligned} pB &= \sum_{i=1}^{m} p_i \underline{\overline{B}}_i \\ &= \sum_{i=1, i\neq k}^{m} p_i \underline{\overline{B}}_i + p_k \underline{\overline{B}}_k \\ &= \sum_{i=1, i\neq k}^{m} p_i \underline{\overline{A}}_i + p_k q \underline{\overline{A}}_k \\ &= \sum_{i=1}^{m} t_i \underline{\overline{A}}_i \\ &= tA \end{aligned}$$

where we have introduced a new row vector t, defined by

$$t_i = \begin{cases} p_i & \text{if } i \neq k \\ qp_k & \text{if } i = k \end{cases}$$

With $pB = 0$ it follows from the equality $pB = tA$ that $tA = 0$.

We now show that t is a semipositive row vector. Let $j \in \mathbb{N}_1^m$, then either $j = k$ or $j \neq k$. If $j = k$ then $t_j = qp_j \geq 0$, and if $j \neq k$ then $t_j = p_j \geq 0$. So $t_j \geq 0$ for $j = 1, 2, \ldots, m$. We know that p is a semipositive row vector, so there is an $s \in \mathbb{N}_1^m$ such that $p_s > 0$. Now suppose $s \neq k$, then $t_s = p_s > 0$, and if $s = k$ then $t_s = qp_s > 0$. So, there is an $s \in \mathbb{N}_1^m$ such that $t_s > 0$. Hence t is a semipositive row vector, and with the result found earlier we can conclude that there is a semipositive row vector t such that $tA = 0$. From this it follows that A is nullable.

Because either $p_k = 0$ or $p_k \neq 0$, we can conclude from the results obtained above that A is nullable if B is nullable, and with the results obtained earlier we can conclude that B is nullable if and only if A is nullable. ■

Definition **4.7** An $n \times n$ matrix D is called a *positive diagonal matrix* if all its diagonal elements are positive, and its non-diagonal elements are zero, that is $D_{ii} > 0$, for $i = 1, 2, \ldots, n$, and $D_{ij} = 0$ whenever $i \neq j$. □

Theorem **4.3** Let A be a nullable $m \times n$ matrix, and let D be an $m \times m$ positive diagonal matrix. Then DA is nullable.

Proof

From theorem ?? it follows that we can write DA as

$$\begin{aligned} DA &= (E_m E_{m-1} \ldots E_1 I_m)A \\ &= E_m E_{m-1} \ldots E_1 A \end{aligned}$$

Let $D_i = E_i E_{i-1} \dots E_1 A$, for $i = 1, 2, \dots, m$. Then we prove the result by induction.

$D_1 = E_1 A$ is nullable by theorem ??. Now suppose D_i is nullable, then $D_{i+1} = E_{i+1} D_i$, which is nullable by theorem ??. So, we can conclude that D_i is nullable for $i = 1, 2, \dots, m$. From this it follows that $D_m = E_m E_{m-1} \dots E_1 A = DA$ is nullable. ■

THEOREM **4.4** Let A be an $m \times n$ matrix, and let D be an $m \times m$ matrix. If DA is nullable, then A is nullable.

PROOF

Because DA is nullable, there is a semipositive row vector p, such that $p(DA) = (pD)A = 0$. The product pD is a row vector of length m, and we have $(pD)_i = p_i D_i i$, for $i = 1, 2, \dots, m$.

Because p is a semipositive row vector we have $p_i \geq 0$, for $i = 1, 2, \dots, m$. Now, because $D_{ii} > 0$ for all $i = 1, 2, \dots, m$ we have that $p_i D_{ii} \geq 0$, for $i = 1, 2, \dots, m$. Because p is a semipositive row vector there is an i such that $p_i > 0$, and so because $D_{ii} > 0$ we have that $p_i D_{ii} > 0$. So there is an i such that $p_i D_{ii} > 0$. From this we can conclude that pD is a semipositive row vector. Let $t = pD$, then from the results above we have that t is a semipositive row vector and $tA = (pD)A = 0$. And so A is nullable. ■

COROLLARY **4.5** Let A be an $m \times n$ matrix, and let the $m \times m$ matrix D be a positive diagonal matrix. Then A is nullable if and only if DA is nullable. ■

THEOREM **4.6** Let A be an $m \times n$ matrix such that its i^{th} column vector $|A|_i$ is a paved mixed vector. Then the expansion $\mathbf{E}(A)_i$ of A is nullable if and only if A is nullable.

PROOF

Let the functions σ_P, σ_N, σ_Z, and the integers p, n and z be defined as in the definition for the expansion of A. Let $b = |A|_i$ the paved mixed column of A. If A is nullable then there is a semipositive row vector $s = (s_1, s_2, \dots, s_m)$ such that $s \cdot b = 0$. We have

$$\begin{aligned} sb &= \sum_{i=1}^{m} s_i b_i \\ &= \sum_{i=1}^{p} s_{\sigma_P(i)} b_{\sigma_P(i)} + \sum_{i=1}^{n} s_{\sigma_N(i)} b_{\sigma_N(i)} + \sum_{i=1}^{z} s_{\sigma_Z(i)} b_{\sigma_Z(i)} \\ &= \sum_{i=1}^{p} s_{\sigma_P(i)} b_{\sigma_P(i)} + \sum_{i=1}^{n} s_{\sigma_N(i)} b_{\sigma_N(i)} + \sum_{i=1}^{z} s_{\sigma_Z(i)} 0 \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{p} s_{\sigma_P(i)} b_{\sigma_P(i)} + \sum_{i=1}^{n} s_{\sigma_N(i)} b_{\sigma_N(i)} \\ &= c\left(\sum_{i=1}^{p} s_{\sigma_P(i)} - \sum_{i=1}^{n} s_{\sigma_N(i)}\right) \\ &= 0 \text{ for some } c > 0. \end{aligned}$$

Which means that

$$\sum_{i=1}^{p} s_{\sigma_P(i)} = \sum_{i=1}^{n} s_{\sigma_N(i)} = T.$$

Now,

$$\begin{aligned} sA &= \sum_{i=1}^{m} s_i \underline{A}_i \\ &= \sum_{i=1}^{p} s_{\sigma_P(i)} \underline{A}_{\sigma_P(i)} + \sum_{i=1}^{n} s_{\sigma_N(i)} \underline{A}_{\sigma_N(i)} + \sum_{i=1}^{z} s_{\sigma_Z(i)} \underline{A}_{\sigma_Z(i)} \end{aligned}$$

Now let us define the following functions, the so called pile functions:

$$\pi_P : \mathbb{N}_1^T \to \mathbb{N}_1^m$$

is defined by

$$\pi_P(x) = \sigma_P(j) \text{ for } \sum_{i=1}^{j-1} s_{\sigma_P(i)} < x \leq \sum_{i=1}^{j} s_{\sigma_P(i)}$$

with $j = 1, 2, \ldots, p$ and $\sum_{i=1}^{j-1} s_i = 0$ if $j = 1$.

$$\pi_N : \mathbb{N}_1^T \to \mathbb{N}_1^m$$

is defined by

$$\pi_N(x) = \sigma_N(j) \text{ for } \sum_{i=1}^{j-1} s_{\sigma_N(i)} < x \leq \sum_{i=1}^{j} s_{\sigma_N(i)}$$

with $j = 1, 2, \ldots, n$ and $\sum_{i=1}^{j-1} s_i = 0$ if $j = 1$.

With these functions we can rewrite the above equation as:

$$\begin{aligned} &\sum_{i=1}^{p} s_{\sigma_P(i)} \underline{A}_{\sigma_P(i)} + \sum_{i=1}^{n} s_{\sigma_N(i)} \underline{A}_{\sigma_N(i)} + \sum_{i=1}^{z} s_{\sigma_Z(i)} \underline{A}_{\sigma_Z(i)} \\ &= \sum_{i=1}^{T} \underline{A}_{\pi_P(i)} + \sum_{i=1}^{T} \underline{A}_{\pi_N(i)} + \sum_{i=1}^{z} s_{\sigma_Z(i)} \underline{A}_{\sigma_Z(i)} \\ &= \sum_{i=1}^{T} \underline{A}_{\pi_P(i)} + \underline{A}_{\pi_N(i)} + \sum_{i=1}^{z} s_{\sigma_Z(i)} \underline{A}_{\sigma_Z(i)} \\ &= \sum_{i=1}^{T} \underline{E}_{(\sigma_P^{-1}(\pi_P(i))-1)\cdot n + \sigma_N^{-1}(\pi_N(i))} + \sum_{i=1}^{z} s_{\sigma_Z(i)} \underline{E}_{np+i} \end{aligned}$$

Let $t = (t_1, t_2, \ldots, t_{np+z})$ be defined such that $tE = \sum_{i=1}^{T} \overline{E}_{(\sigma_P^{-1}(\pi_P(i))-1)\cdot n+\sigma_N^{-1}(\pi_N(i))} + \sum_{i=1}^{z} s_{\sigma_Z(i)}\overline{E}_{np+i}$.

Now, $T = 0$, or $T > 0$. If $T = 0$, then $s_{\sigma_P(i)} = 0$ for all $i = 1, 2, \ldots, p$ and $s_{\sigma_N(i)} = 0$ for all $i = 1, 2, \ldots, n$, so there must be an i such that $s_{\sigma_Z(i)} > 0$ for $i = 1, 2, \ldots, z$. This means that $t_{np+i} > 0$ and so there is a j such that $t_j > 0$. This, with $t_j \geq 0$ for all $j = 1, 2, \ldots, np + z$, means that t is a semipositve row vector. So there is a semipositive row vector t such that $tE = 0$.

Now, suppose that $T > 0$. Let $1 \leq i \leq T$, then $1 \leq (\sigma_P^{-1}(\pi(i)) - 1) \cdot n + \sigma_N^{-1}(\pi_N(i)) \leq np$, which together with equation ?? it follows that $t_{\sigma_P^{-1}(\pi(i))-1)\cdot n+\sigma_N^{-1}(\pi_N(i))} > 0$. So there is a j, with $1 \leq j \leq np + z$ such that $t_j > 0$, and because $t_j \geq 0$ for all $1 \leq j \leq np + z$ t is a semipositive row vector. So there is a semipositive row vector t such that $tE = 0$. So E is nullable.

Now suppose that E is nullable. Then there exists a semipositive row vector $t = (1, 2, \ldots, M)$, with $M = np + z$, such that $tE = 0$.

$$\begin{aligned} tE &= \sum_{i=1}^{M} t_i\overline{E}_i \\ &= \sum_{i=1}^{np} t_i\overline{E}_i + \sum_{i=np+1}^{np+z} t_i\overline{E}_i \end{aligned}$$

Now, $z = 0$ or $z > 0$. If $z = 0$ then $\sum_{i=1}^{M} t_i\overline{E}_i = \sum_{i=1}^{np} t_i\overline{E}_i$. Because t is a semipositive row vector, there must be an i with $i = 1, 2, \ldots, np$ such that $t_i > 0$. Now let us rewrite the terms in the above sum as follows.

$$\begin{aligned} tE &= \sum_{i=1}^{np} t_i\overline{E}_i \\ &= \sum_{i=1}^{np} t_i(\overline{A}_{\sigma_P(\lfloor\frac{i-1}{n}\rfloor+1)} + \overline{A}_{\sigma_N(((i-1) \bmod n)+1)}) \\ &= \sum_{i=1}^{np} t_i\overline{A}_{\sigma_P(\lfloor\frac{i-1}{n}\rfloor+1)} + t_i\overline{A}_{\sigma_N(((i-1) \bmod n)+1)} \\ &\equiv \sum_{i=1}^{np} s_i\overline{A}_i \end{aligned}$$

The row vector $s = (1, 2, \ldots, np)$ is a newly defined row vector. The expression given above is an expression involving rows of A and at least one coefficient is positive because $t_i > 0$ for some $i = 1, 2, \ldots, np$ as explained above. So there is a j with $j = 1, 2, \ldots, np$ such that $s_j > 0$. This, together with $s_j \geq 0$ for all $j = 1, 2, \ldots, np$ means that s is a semipositive row vector. So there is a semipositive row vector s such that $sA = 0$.

Now, suppose that $z > 0$. Then

$$\begin{aligned}\sum_{i=1}^{M} t_i \underline{\overline{E}}_i &= \sum_{i=1}^{np} t_i \underline{\overline{E}}_i + \sum_{i=np+1}^{np+z} t_i \underline{\overline{E}}_i \\ &= \sum_{i=1}^{np} t_i (\underline{\overline{A}}_{\sigma_P(\lfloor \frac{i-1}{n} \rfloor + 1)} + \underline{\overline{A}}_{\sigma_N(((i-1) \bmod n)+1)}) + \sum_{i=np+1}^{np+z} t_i \underline{\overline{A}}_{\sigma_Z(i-np)}\end{aligned}$$

This is an expression with only rows of A, so we can introduce a rowvector $s = (1, 2, \ldots, s_m)$ such that

$$\sum_{i=1}^{M} s_i \underline{\overline{A}}_i = \sum_{i=1}^{np} t_i (\underline{\overline{A}}_{\sigma_P(\lfloor \frac{i-1}{n} \rfloor + 1)} + \underline{\overline{A}}_{\sigma_N(((i-1) \bmod n)+1)}) + \sum_{i=np+1}^{np+z} t_i \underline{\overline{A}}_{\sigma_Z(i-np)}$$

Because $t_i > 0$ for some $i = 1, 2, \ldots, M$ we have $s_i > 0$ for some $i = 1, 2, \ldots, m$ and $s_i \geq 0$ for all $i = 1, 2, \ldots, m$. So there is a semipositive row vector s such that $sA = 0$. So A is nullable. ■

THEOREM **4.7** Let A be an $m \times n$ matrix such that its i^{th} column contains only zero elements. Then A is nullable if and only if $\overset{\perp}{A}_i$ is nullable.

PROOF

If A is nullable then there exists a semipositive row vector p such that $pA = 0$, so $p|A|_j = 0$ for all $j = 1, 2, \ldots, n$. This means that $p|A|_j = 0$ for all $j = 1, 2, \ldots, n$, $j \neq i$, and so $p \overset{\perp}{A}_i = 0$. So there is a semipositive row vector p such that $p \overset{\perp}{A}_i = 0$, and this means that $\overset{\perp}{A}_i$ is nullable.

Now suppose that $\overset{\perp}{A}_i$ is nullable. Then there is a semipositive row vector p such that $p \overset{\perp}{A}_i = 0$, which means that $p\left|\overset{\perp}{A}_i\right|_j = 0$ for all $j = 1, 2, \ldots, n-1$, and $p\underline{0} = 0$,with $p\underline{0} = p \overset{\perp}{A}_i$ this means that $p|A|_j = 0$ for all $j = 1, 2, \ldots, n$. So $pA = 0$, and thus there is a semipositive row vector p such that $pA = 0$, which means that A is nullable. ■

DEFINITION **4.8** Let A be an $m \times n$ matrix, and let the k^{th} column vector of A be a semipositive or seminegative vector, with at least one element zero.

Let the column vector v of size m be defined as follows:

$$v = \begin{cases} |A|_k & \text{if } |A|_k \text{ is a semipositive vector} \\ -|A|_k & \text{if } |A|_k \text{ is a seminegative vector} \end{cases}$$

1. Let Z and $\overline{Z}$ be sets of natural numbers that are defined as follows:

$$\begin{aligned} Z &= \{1 \leq j \leq m | v > 0\} \ , \\ \overline{Z} &= \{1 \leq j \leq m | v \neq 0\} \end{aligned}$$

2. Define the positive integers z and p as:

$$\begin{aligned} z &= sizeof(Z) \ , \\ p &= sizeof(\overline{Z}) \ , \end{aligned}$$

3. Let the function $\sigma_Z : \mathbb{N}_1^z \rightarrow Z$ be defined by the following property:

$$\sigma_Z(i) < \sigma_Z(j) \text{ if } i < j, \text{ for all } i, j \in \mathbb{N}_1^z$$

Let the function $\sigma_{\overline{Z}} : \mathbb{N}_1^p \rightarrow \overline{Z}$ be defined by the following property:

$$\sigma_{\overline{Z}}(i) < \sigma_{\overline{Z}}(j) \text{ if } i < j, \text{ for all } i, j \in \mathbb{N}_1^p$$

4. Let B be the $z \times n$ matrix defined as:

$$\underline{\overline{B}}_i = \underline{\overline{A}}_{\sigma_Z(i)} \text{ for } 1 \leq i \leq z$$

Now, the *reduction* of A in column k, denoted by $\mathbf{R}(A)_k$, is the $z \times (n-1)$ matrix defined by:

$$\mathbf{R}(A)_k = \overset{\perp}{B}_k$$

□

THEOREM **4.8** Let A be an $m \times n$ matrix with its k^{th} column vector a semipositive or seminegative vector with at least one of its elements equal zero. Then A is nullable if and only if $\mathbf{R}(A)_k$ is nullable.

PROOF

Let the integers p and z and the functions σ_Z and $\sigma_{\overline{Z}}$ be defined as in the definition for reduction. Let B be the $z \times n$ matrix as defined in the definition for reduction.

If A is nullable then there exists a semipositive row vector t such that $tA = 0$. Let $v = \mathsf{I}A\mathsf{I}_k$ and suppose that v is a semipositive column vector. From the fact that $tA = 0$, we have $tv = 0$, but

$$\begin{aligned} tv &= \sum_{i=1}^{m} t_i v_i \\ &= \sum_{i=1}^{z} t_{\sigma_Z(i)} v_{\sigma_Z(i)} + \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} v_{\sigma_{\overline{Z}}(i)} \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{z} t_{\sigma_Z(i)} 0 + \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} v_{\sigma_{\overline{Z}}(i)} \\ &= \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} v_{\sigma_{\overline{Z}}(i)} \\ &\geq \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \cdot 1 \\ &= \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \end{aligned}$$

So we have

$$\sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \leq tv = 0$$

Because $t_i \geq 0$ for all $1 \leq i \leq m$ it follows that $\sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \geq 0$, and so we have

$$0 \leq \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \leq 0$$

From this it can be concluded that $\sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} = 0$. Because $t_i \geq 0$ for all $1 \leq i \leq m$ it follows that $t_{\sigma_{\overline{Z}}(i)} = 0$ for all $1 \leq i \leq p$.

Now, suppose v is a seminegative column vector, then we have from the fact that $tA = 0$ that $tv = 0$, but

$$\begin{aligned} tv &= \sum_{i=1}^{m} t_i v_i \\ &= \sum_{i=1}^{z} t_{\sigma_Z(i)} v_{\sigma_Z(i)} + \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} v_{\sigma_{\overline{Z}}(i)} \\ &= \sum_{i=1}^{z} t_{\sigma_Z(i)} 0 + \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} v_{\sigma_{\overline{Z}}(i)} \\ &= \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} v_{\sigma_{\overline{Z}}(i)} \\ &\leq \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \cdot -1 \\ &= -\sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \end{aligned}$$

So, from this it follows that $-tv \geq \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)}$. But $tv = 0$, so we have $\sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \leq$

0. Because $t_i \geq 0$ for all $1 \leq i \leq m$ it follows that $\sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \geq 0$, and so we have

$$0 \leq \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \leq 0$$

From this it can be concluded that $\sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} = 0$. Because $t_i \geq 0$ for all $1 \leq i \leq m$ it follows that $t_{\sigma_{\overline{Z}}(i)} = 0$ for all $1 \leq i \leq p$.

But if $t_{\sigma_{\overline{Z}}(i)} = 0$ for all $1 \leq i \leq p$ then it follows that $t_{\sigma_Z(i)} > 0$ for some $1 \leq i \leq z$.

Now, with the result obtained above it follows that

$$\begin{aligned}
tA &= \sum_{i=1}^{m} t_i \underline{\overline{A}}_i \\
&= \sum_{i=1}^{p} t_{\sigma_{\overline{Z}}(i)} \underline{\overline{A}}_{\sigma_{\overline{Z}}(i)} + \sum_{i=1}^{z} t_{\sigma_Z(i)} \underline{\overline{A}}_{\sigma_Z(i)} \\
&= \sum_{i=1}^{p} 0 \cdot \underline{\overline{A}}_{\sigma_{\overline{Z}}(i)} + \sum_{i=1}^{z} t_{\sigma_Z(i)} \underline{\overline{A}}_{\sigma_Z(i)} \\
&= 0 + \sum_{i=1}^{z} t_{\sigma_Z(i)} \underline{\overline{A}}_{\sigma_Z(i)} \\
&= \sum_{i=1}^{z} t_{\sigma_Z(i)} \underline{\overline{A}}_{\sigma_Z(i)} \\
&= \sum_{i=1}^{z} t'_i \underline{\overline{B}}_i \\
&= t'B
\end{aligned}$$

With $t'_i = t_{\sigma_Z(i)}$ for $1 \leq i \leq z$. The row vector t' is a semipositive row vector, so from this and the results obtained above it follows that B is nullable. Because $\mathbf{R}(A)_k = \overset{\perp}{B}_k$ it follows from theorem ?? and the fact that B is nullable that $\mathbf{R}(A)_k$ is nullable.

Now, let $\mathbf{R}(A)_k$ be nullable. Then there is a semipositive row vector s such that $s\mathbf{R}(A)_k = 0$. From this and theorem ?? it follows that $sB = 0$. Let us rewrite sB as follows

$$\begin{aligned}
sB &= \sum_{i=1}^{z} s_i \underline{\overline{B}}_i \\
&= \sum_{i=1}^{z} s_i \underline{\overline{A}}_{\sigma_Z(i)} \\
&= \sum_{i=1}^{z} s_i \underline{\overline{A}}_{\sigma_Z(i)} + \sum_{i=1}^{p} 0 \cdot \underline{\overline{A}}_{\sigma_{\overline{Z}}(i)}
\end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{m} t_i \underline{\overline{A}}_i \\ &= tA \end{aligned}$$

with t defined as follows

$$\begin{aligned} t_{\sigma_Z(i)} &= s_i \text{ for } 1 \leq i \leq z \\ t_{\sigma_{\overline{Z}}(i)} &= 0 \text{ for } 1 \leq i \leq p \end{aligned}$$

Because there is an i with $1 \leq i \leq z$ such that $s_i > 0$ it follows that $t_{\sigma_Z(i)} > 0$ for some $1 \leq i \leq z$, and thus there is a j with $1 \leq j \leq m$ such that $t_j > 0$. Further, form the properties of s and the fact that $t_{\sigma_{\overline{Z}}(i)} = 0 \geq 0$ it follows that $t_j \geq 0$ for all $1 \leq j \leq m$. Thus t is a semipositive row vector and so A is nullable. ■

5 Nullability Reduction

CHAPTER

In the previous chapter we showed that there are operations on matrices that preserve the property of nullability. In this chapter we will introduce the reduction method. Section 5.2 describes the so called reduction rules and section 5.3 describes the method itself. Section 5.4 describes an algorithm that in a structured way produces a reduction for every matrix. Finally section 5.5 discusses some properties of matrices that could be used to improve the reduction method.

5.1 Introduction

In figure 5.1 an example is given of a reduction. As can be seen it consists of a series of matrices, stacked on top of each other, with the original matrix on top. Next to the matrices there are some notes, which are called annotations. Each matrix except for the first is the result of applying a reduction rule to the matrix direct above it. The note on the right of the matrix to which the reduction rule is applied mentions the rule and some optional data that is necessary for applying the reduction rule. In the following sections we will give formal descriptions of all items mentioned above.

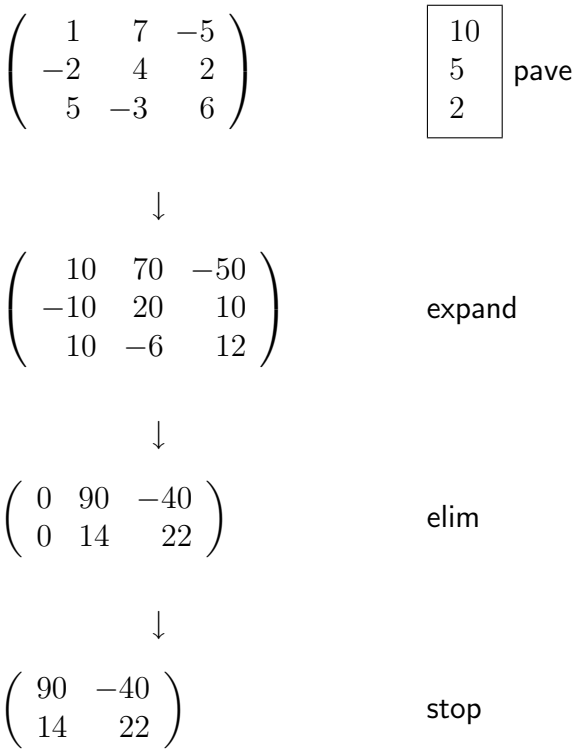

Figure 5.1: An example of a reduction.

We will start the discussion of the reduction method a description of its reduction rules in the next section.

5.2 The Reduction Rules

In this section we will give a formal treatment of the reduction rules. Reduction rules are the tools for transforming matrices into matrices that have a more convenient form. The reduction rules preserve the property of nullability and are equivalent to the corresponding functions we discussed in the previous chapter.

Before we can begin our discussion of the reduction rules, however, we first need to introduce the concept of actions. Actions are functions on matrices. They describe an operation on matrices. These actions will be used to describe the operations of reduction rules later on in this section. We first need to define two concepts that will allow us to formulate the actions more clearly. The first introduces a notation for sets of matrices that have the same number of rows and columns, and the second definition introduces a property of vectors that will be used often in the definitions that will follow.

DEFINITION **5.1** We introduce the following notations for sets of matrices.

1. $M_{m\times n}$ is the set of all matrices that have m rows and n columns.
2. $M_{m^*\times n}$ is the set defined by
$$M_{m^*\times n} = \bigcup_{i=1}^{m} M_{i\times n}$$
3. $M_{m\times n^*}$ is the set defined by
$$M_{m\times n^*} = \bigcup_{i=1}^{n} M_{m\times i}$$

□

Most actions depend on the value of the first entry of a row vector and sometimes only their sign is of importance. The following definition reflects this by defining a convenient property for this purpose.

DEFINITION **5.2** Let v be a row vector of length n. Then v is called

1. a positive endian row vector if v_1 is positive.
2. a negative endian row vector if v_1 is negative.

3. a zero endian row vector if v_1 is zero.

□

We are now ready to give the definition of an action, together with the actions that are actually defined for the reduction method.

DEFINITION **5.3** ***(actions)***
Actions are functions that take a matrix and produce a matrix. The following actions are defined. Let A be the $m \times n$ matrix on which the action is applied, and let B be the $p \times q$ matrix that is the result of applying the action to the matrix A.

A–1 elim The elimininination action is a function $\mathsf{elim} : M_{m\times n} \to M_{m\times(n-1)}$ defined by

$$\mathsf{elim}(A) = \begin{pmatrix} \perp \\ A_1 \end{pmatrix}$$

So the elimination action removes the first column of A.

A–2 basred The basic reduction action is a function $\mathsf{basred} : M_{m\times n} \to M_{m^*\times n}$ defined by $A \mapsto \mathsf{basred}(A)$. The mapping is defined as follows.

Let $z_1, z_2, \ldots, z_s$ be the zero endian rows of A such that z_i is the i^{th} zero row of A counted from above, i.e. from the first row of A. Let's define $B = \mathsf{basred}(A)$, then the result B is defined by

$$\underline{\overline{B}}_i = z_i \quad \text{for all } 1 \leq i \leq s.$$

A–3 reduce The reduction action is a function $\mathsf{reduce} : M_{m\times n} \to M_{m^*\times(n-1)}$ which is the composite function of basic reduction and elimination, that is

$$\mathsf{reduce} = \mathsf{elim} \circ \mathsf{basred} : M_{m\times n} \to M_{m^*\times(n-1)}$$

So first the matrix is reduced. The matrix that results has a first column that is the all-zero vector. Then this column is removed from the result, and the result is the value of reduce at $A \in M_{m\times n}$.

A–4 rowmul The row multiplication action is a function $\mathsf{rowmul} : M_{m\times n} \to M_{m\times n}$ defined by

$$A \mapsto A \otimes v$$

where v is a positive column vector such that there is a k such that $v_k > 1$ and for all $i \neq k$ we have $v_i = 1$. The vector v must be specified externally.

A–5 basexp The basic expansion action is a function basexp : $M_{m\times n} \to M_{q\times n}$.

We will now explain how the mapping of A is defined. Let $u_1, u_2, \ldots, u_k$ be the positive endian rows of A, where u_i is the i^{th} positive endian row in A counted from above. Let $v_1, v_2, \ldots, v_s$ be the negative endian rows of A, where v_i is the i^{th} negative endian row of A counted from above, i.e from row 1 of A. Let $z_1, z_2, \ldots, z_y$ be the zero endian rows of A, where again z_i is the i^{th} zero endian row of A counted from the first row of A. Together the positive endian, negative endian, and zero endian rows account for all rows of A.

Now we will construct a new matrix H as follows. Take u_1 and add to that each time a negative endian row to u_1. Each result is a new row of H. So the first and second row of H become, respectively

$$\underline{\overline{H}}_1 = u_1 + v_1, \quad \underline{\overline{H}}_2 = u_1 + v_2, \text{ etc.}$$

When all negative endian rows are added this way to u_1 we take the next positive endian row, i.e. u_2. Again we add to this row all negative rows. That is

$$\underline{\overline{H}}_{s+1} = u_2 + v_1, \quad \underline{\overline{H}}_{s+2} = u_2 + v_2, \text{ etc.}$$

We repeat this process until there are no positive endian rows left.

Then we copy the zero endian rows of A, in order, to H. That is

$$\underline{\overline{H}}_{k\cdot s+1} = z_1, \ \underline{\overline{H}}_{k\cdot s+2} = z_2, \text{ etc.}$$

This is repeated until all zero rows are added to H this way.

Now H is the mapping of A, i.e.

$$\mathsf{basexp}(A) = H$$

A–6 expand The expansion action is the composite function of basic expansion and elimination, i.e.

$$\mathsf{expand} = \mathsf{elim} \circ \mathsf{basexp} : M_{m\times n} \to M_{q\times(n-1)}$$

So, first the matrix is expanded. The resulting matrix has a first column that is the all-zero vector. Then this column is removed and the result is the value of expand. at $A \in M_{m\times n}$.

A–7 pave The pave action is a function pave : $M_{m\times n} \to M_{m\times n}$ defined by

$$A \mapsto A \otimes v$$

The vector v is a positive column vector of length m that has to be specified externally and has to satisfy the following conditions

1. $A \otimes v$ must satisfy the paved condition.
2. $a_i = b_i$ if a_i is zero endian. For all $1 \leq i \leq m$. Where a_i is the i^{th} row of A and b_i is the i^{th} row of B.

A–8 tight The tight paving action is a function $\mathsf{tight} : M_{m\times n} \to M_{m\times n}$ that is defined by

$$A \mapsto A \otimes v$$

The positive column vector v of length m is defined as follows.

Let a be the first column vector of A, i.e. $a = |A|_1$. Now, let P be the set of prime integers defined by

$$P = \{p \in \mathbb{N} |\ \text{p is prime and there is an entry of } a \text{ that can be divided by } p\}.$$

Let s be the size of P, i.e. $s = |P|$. Let p be the prime vector, i.e. it contains all primes in P ordered from small to big, i.e.

1. $p_i < p_j$ if $i < j$ for all $1 \leq i \leq n$

Now define the matrix K of exponents, i.e. a row i of K represents the exponents of the prime factors of the i^{th} entry of a. If $a_i = 0$ then the corresponding row of K is the all-zero vector, otherwise a_i can be written as

$$a_i = p_1^{e_{i1}} \cdot p_2^{e_{i2}} \cdot \ldots \cdot p_n^{e_{in}}$$

Now let k be the i^{th} row of K, i.e. $k = \underline{K}_i$, then k is defined as follows

$$k_i = e_{ij} \quad \text{for } 1 \leq j \leq n$$

Next, let the row vector $t = (t_1, t_2, \ldots, t_n)$ be defined such that t_1 is maximum entry of the first column of K, and t_i is the maximum entry of the i^{th} column of K in general.

Now we construct a matrix M that is a repetition of row vector t, such that the number of rows equals the number of entries of a. So M and K have the same size, that is the same number of rows and the same number of columns. Each row of M equals t, i.e. $\underline{M}_i = t$ for all $1 \leq i \leq m$.

Let V be defined by $V = M - K$. Now let v be defined as follows. If $a_i = 0$ then $v_i = 1$, otherwise let $w = \underline{V}_i$, then $v_i = p_1^{w_1} \cdot p_2^{w_2} \cdot \ldots \cdot p_n^{w_s}$. And the vector v is defined.

□

elim	A1	removes first column
basred	A2	removes nonzero endian rows
reduce	A3	does basred and then elim
basexp	A4	expands the matrix
expand	A5	does basexp and then elim
rowmul	A6	multiplies a row with a positive integer
pave	A7	paves first column
tight	A8	smallest pave of first column

Table 5.1: The actions

An overview of all actions is given in table 5.2. Every action is mentioned by its corresponding item number in the definition given above.

The tight paving action deserves some explanation. It is a well known fact from number theory that every integer greater than 1 can be written as a product of prime integers. The definition above uses this fact to construct a vector v that makes the first column of the resulting matrix paved, but in such a way that it is the smallest envelope possible. The action is a specialized form of paving, hence **tight** paving.

DEFINITION **5.4** ***(matching pave vector)***
Let A be an $m \times n$ matrix and let v be a positive column vector of length m. Then v is called a matching pave vector if the following conditions are satisfied

1. $B = A \otimes v$ is a matrix for which the first column satisfies the paved condition.
2. Let a_i be the i^{th} row of A and let b_i be the i^{th} row of B. Then $a_i = b_i$ if a_i is a zero endian row vector of A, for all $1 \leq i \leq m$.

□

We are now ready to define the reduction rules, but before we do that we have to introduce some additional properties that a matrix can have. They are called conditions and are defined by the following definition.

DEFINITION **5.5** ***(conditions)***
Let A be an $m \times n$ matrix and let a be the first column vector of A, that is $a = |A|_1$. Let the following conditions be defined.

positive condition If a is positive then it is said that A satisfies the positive condition.

zero condition If $a_1 = a_2 = \ldots = a_n = 0$ then it is said that A satisfies the zero condition.

negative condition If a is negative then it is said that A satisfies the negative condition.

paveable condition If a is mixed and a is not paved then it is said that A satisfies the mixed condition.

expandable condition If a is mixed and a is paved then it is said that A satisfies the expandable condition.

reduceable condition If a is seminegative or semipositive and A has at least one zero endian row vector then it is said that A satisfies the reduceable condition.

□

Next follow the reduction rules. Some reduction rules need some additional requirements or extra data that is used to perform the reduction rule. These will be given in the items that follow.

DEFINITION **5.6** ***(reduction rules)***
A reduction rule is a tuple (N, C, A), where N is the name of the reduction rule, C is the condition that has to be satisfied by the matrix M to which the reduction rule is applied, and A is the action that is applied to M. The reduction that can be used in the reduction method are defined in table 5.2. We will give a short description of each and some additional requirements that are not mentioned in the table.

1. Row Multiplication rule. An extra integer has to be specified. The integer that has to be specified is the positive integer by which the row has to multiplied.
2. Pave rule. We have to specify a positive column vector that must be a matching—see the definition above—for the matrix. This vector is specified in the annotation as a column of positive integers.
3. Tight pave rule. No addtional data has to be specified.
4. Eliminate rule. No additional data has to be specified.
5. Basic Reduce rule. No additional data has to be specified.
6. Reduce rule. No additional data has to be specified.
7. Basic Expand rule. No additional data has to be specified.
8. Expand rule. No additional data has to be specified.

□

Rule No.	Name	Short Description	Action	Condition
R1	**rowmul**	multiplies a row with pos integer	rowmul	None
R2	**pave**	paves A	pave	paveable
R3	**tight pave**	paves A with smallest envelope	tight	paveable
R4	**elim**	removes first column	elim	zero
R5	**basred**	keeps zero endian rows	basred	reduceable
R6	**reduce**	keeps zero endian rows; removes first column	reduce	reduceable
R7	**basexp**	expands A	basexp	expandable
R8	**expand**	expands; removes first column	expand	expandable

Table 5.2: The set of reduction rules that may be used in the reduction ethod.

5.3 The Reduction Method

When we perform a reduction we put some notes to the right of the matrix. Such information is for instance the rule which is applied to the matrix and possibly some optional data that is needed to perform the operation. We call such notes an annotation and it is defined as follows.

DEFINITION **5.7** ***(annotation)***
An annotation is a tuple (S, R, T), where R is a reference to a reduction rule, T is an optional text that gives a short description of what is accomplished by applying the reduction rule. Finally S is specific information that is needed for the application of the reduction rule mentioned by R. There are three types of annotatations that are used in the reduction method, namely

1. Annotations where only the rule is mentioned. So S and T are empty.
2. Annotations where S is filled with either a positive integer for the row multiplication rule, or S is a column of positive integers, for example

$$\begin{bmatrix} 13 \\ 7 \\ \bullet \\ 9 \\ 3 \end{bmatrix}$$

   The dot means we do not multiply this row and we have to think a one in that position. The dot is a clear reminder that we do not multiply the particular row and it is better than a one in that position which would be easily glanced over.
3. Annotations of type 1 or 2, but with the T being set to a short descriptive text that describes the operation.

□

The combination of matrix and annotation is called an annotated matrix and it is the building block for a reduction diagram. It is defined as follows

DEFINITION **5.8** ***(annotated matrix)***
An annotated matrix G is a pair $G = (M, A)$, where M is a matrix and A is an annotation. □

When performing the reduction rules in sequence to obtain a desired matrix, we produce a sequence of annotated matrices. This sequnce is what we call a reduction diagram, and is defined as follows

DEFINITION **5.9** ***(reduction diagram)***
A reduction diagram R is a sequence of annotated matrices. The interval $D = [1, n]$, where $n \in \mathbb{N}$, is the domain of the sequence. The matrices are numbered from 1 to and including n, so that gives the matrices $G_1, G_2, \ldots, G_n$.

The first matrix is called the start matrix, and the last matrix is called the end matrix. The end matrix is possibly an empty matrix, i.e. a matrix that has no columns and no rows. □

We have given a schematic representation of a reduction diagram in figure 5.3.

Note that not every reduction diagram is a proper reduction. We could for example combine a matrix and rule as annotation where that does not make sense, but it would still be a reduction diagram if the annotated matrices are formed properly. The following definition states what makes a reduction diagram a valid reduction.

DEFINITION **5.10** ***(reduction)***
Let A be an $m \times n$ matrix, and let R be a reduction diagram with interval $D = [1, n]$ and annotated matrices $G_1, G_2, \ldots, G_n$. Let M_i be the matrix of G_i and let r_i be the rule mentioned in the annotation belonging to G_i. Then R is a *reduction* if the conditions are satisfied

1. $M_1 = A$
2. M_n is empty or satisfies the positive condition, or it satisfies the negative condition.
3. Matrix M_i satisfies the condition of reduction rule r_i.
4. Every matrix M_i in G_i with $1 < i \leq n$ is the result of applying reduction rule r_{i-1} to M_{i-1} where the information that is needed for applying r_{i-1} is found in the annotation belonging to G_{i-1}.

The length of the reduction, denoted length(R), is the number of annotated matrices, i.e. length(R) $= n$. □

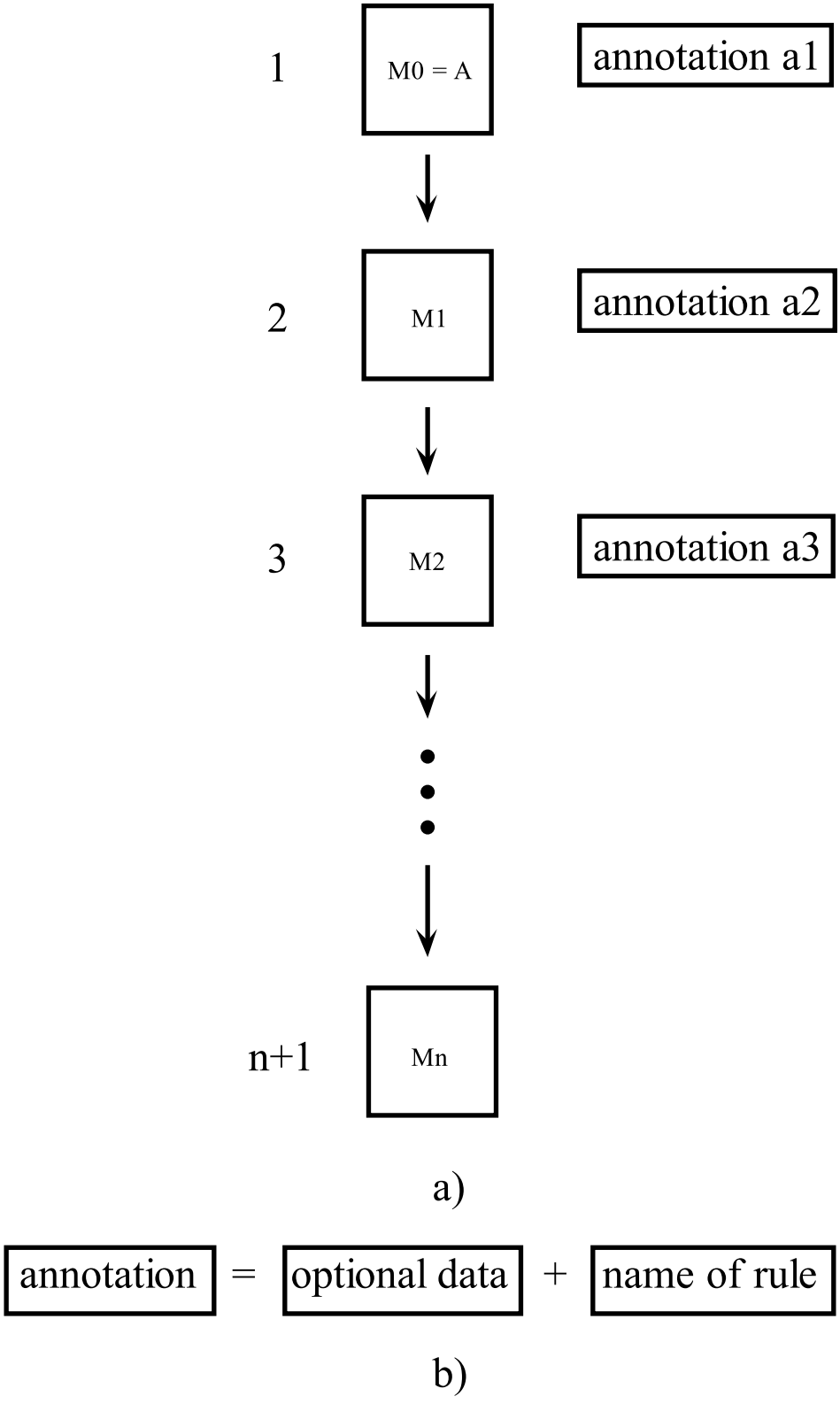

Figure 5.2: A schematic representation of a reduction diagram.

5.4 The Reduction Algorithm

With the reduction method it is possible to derive many different reductions that all lead to the same end result. In this section we will introduce an algorithm that leads to a sequence that we will call the standard reduction.

The algorithm is shown in figure 5.3. It uses a subset of the reduction rules. We still need to show that the algorithm is correct. This means it must not get stuck and it must end in finite number of iterations.

First of all notice that it can never encounter a situation where the first column of A is not satisfied by one of the tests in the algorithm. In other words there is always a test that is satisfied by the first column of A no matter what that column is. So the algorithm can indeed not get stuck by an unforseen condition.

Second notice that as it traverses the flow diagram it encounters a number of tests. Some of these tests stop the algorithm. But it is also possible, as can be seen by inspection of the flow diagram, that we can turn up at the beginning of the loop again. So there seems to be the possibility of an infinite loop. We will now show that this is not the case.

Notice that before we return to the starting point, the number of columns of A is reduced. So if n_i is the number of columns of the resulting matrix in iteration i, and we start with $i = 0$ then $n_0 = n$ which is the number of columns of A, the original matrix. We also have

$$n_0 > n_1 > n_2 > \ldots > n_k$$

So in the end if the algorithm does not stop otherwise, the algorithm will stop by the test that checks if the number of columns equals zero. In other words the algorithm will stop when $n_k = 0$. So, in short, if algorithm does not stop it reduces the matrix and runs from the beginning again, until either a stop condition is performed or the matrix is out of columns.

Finally, as was mentioned at the beginning of this section, the algorithm gives rise to the following definition which defines a standard reduction for each matrix.

Definition **5.11** The reduction for matrix A derived with the flowchart of figure 5.3 is called the standard reduction for A. □

5.5 Efficiency

The reduction method considers only the first column of the matrix it reduces. Consider the following matrix

$$A = \begin{pmatrix} -1 & -2 & 1 \\ 6 & -5 & 2 \\ 4 & 9 & 3 \end{pmatrix}$$

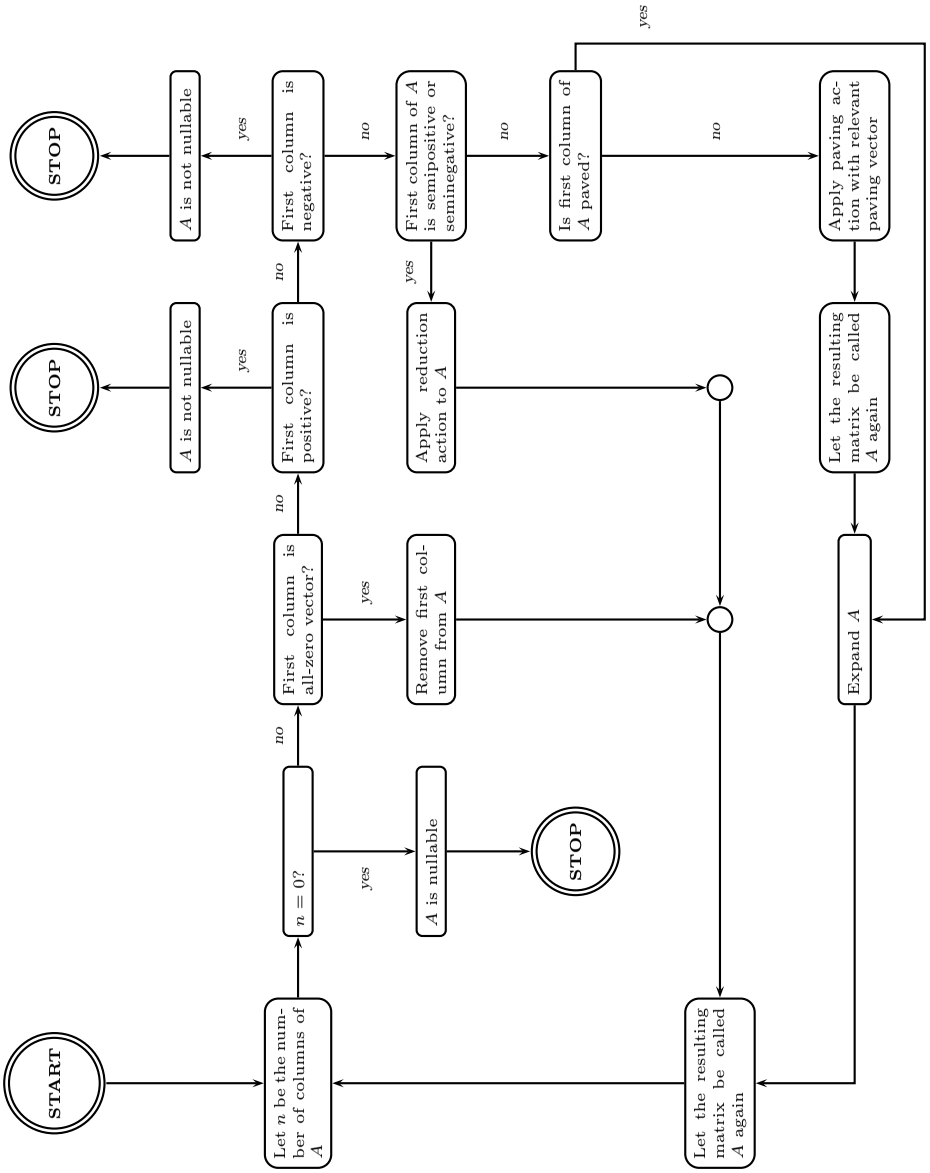

Figure 5.3: The flow chart that can be used to produce a standard reduction of a matrix.

The reduction method has to go all the way to the last column to find out that matrix A is not nullable. But by theorem?? in the chapter on nullability the first and last column can change from position without modifying the nullability of the matrix. The matrix would look like

$$A' = \begin{pmatrix} 1 & -2 & -1 \\ 2 & -5 & 6 \\ 3 & 9 & 4 \end{pmatrix}$$

Now the reduction method would immediately stop on the first column of A'.

So from this example it is clear that by changing the positions of columns we can alter the length of the reduction. By extending the reduction rules of the reduction method with a rule stating that positions of columns can be exchanged we would improve the reduction method considerably.

We could alternatively reorder the columns of A in advance, that before we apply the reduction method. But that would introduce some problems, because it is not always clear in advance which columns should be swapped in order to reduce the length of the reduction. Besides this gives us no opportunity to change the resulting matrices as we see fit while performing the reduction method. So adding additional reduction rules would definitely be the way to improve the reduction methods and it would use in this way use all available information of the matrix.

6 Finding Weight Vectors

CHAPTER

In the first section we will prove for every reduction rule that there is a vector y for A such that $Ay > 0$ when $By' > 0$. Of course we already know that A has such a vector if there is such a vector for B, because the reduction rules preserve the property of nullability.

But what is special about these theorems is that they show that is always possible to construct a vector y for A from the vector y' for B.

This opens up the possibility to introduce a set of rules with which it is possible to construct a vector y, once a reduction is given that ends in positve or negative matrix. These rules are called the reverse reduction rules and are described in the second section.

6.1 Some Theorems

DEFINITION **6.1** Let v be a vector of length n, then the *envelope* m is defined by

1. m is positive and m or $-m$ is an entry of v.

2. $|v_i| \leq m$ for all $1 \leq i \leq n$.

□

THEOREM **6.1** Let A be an $m \times n$ matrix and let B be the $q \times (n-1)$ matrix, such that A satisfies the paved condition, and B is the result of applying the expand action to A. Let y be such that $By > 0$. Then there is a y' such that $Ay' > 0$.

PROOF

We are first going to prove some properties of the rows of A. In order to do that we will define some sets, containing rows of A.

Let P be the set defined by

$$P = \{\overline{A}_i | 1 \leq i \leq m, \ \overline{A}_i \text{ is positive endian}\}$$

So P is the set of rows of A where for each row the first entry of the row is positive. Similarly, define the set N as

$$N = \{\overline{A}_i | 1 \leq i \leq m, \ \overline{A}_i \text{ is negative endian}\}$$

and finally define the set Z as

$$Z = \{\overline{A}_i | 1 \leq i \leq m, \ \overline{Z}_i \text{ is zero endian}\}$$

Now, define u as a column vector such that

$$\begin{aligned} u_1 &= 0 \\ u_{j+1} &= y_j \quad \text{for } 1 \leq j \leq n-1 \end{aligned}$$

Property 1

Now, at least one of the sets P and N must be such that for all rows r of this set we have $ru > 0$, for if not then for both of these sets we have that there are rows r_P and r_N such that

$$r_P u \leq 0 \ \text{ and } \ r_N \leq 0 \tag{6.1}$$

Now the sum $r_{P+N} = r_P + r_N$ is a row of B if we remove the first entry of this sum. For this row r_B we have

$$r_B y > 0$$

which is equivalent to

$$r_{P+N} \cdot u > 0 \tag{6.2}$$

But from equation (6.1) it follows that

$$(r_P + r_N) \cdot u \leq 0$$

which is equivalent to

$$r_{P+N}u \leq 0 \tag{6.3}$$

But equation (6.3) contradicts equation (6.2).

We will call the set that has the property that $ru > 0$ for all rows r of the set $\mathcal{P}$.

The set that satisfies the above described property will be called $\mathcal{P}$, the other set will be called $\mathcal{N}$.

Property 2

The second property we are going to prove is that in absolute value the smallest positive value of ru in $\mathcal{P}$ is greater than the largest (in absolute value) negative value of ru in $mathcalN$.

For if not, let r_P be the row in $\mathcal{P}$ for which $r_P u$ is the smallest positive value , and let r_N be the row in $mathcalN$ for which the value $r_N u$ is the greatest, in absolute value, of the negative values for ru.

Now the sum $r_{P+N} = r_P + r_N$ is a row r_B of B if we remove the first entry of this sum. For this row r_B we have

$$r_B y > 0$$

which is equivalent to

$$r_{P+N} > 0 \tag{6.4}$$

But from the above statements it follows that

$$(r_P + r_N)u \leq 0$$

which is equivalent to

$$r_{P+N}u \leq 0 \tag{6.5}$$

But equation (6.5) contradicts with equation (6.4).

Now, if $\mathcal{N}$ has no negative values for ru then we let $w_A = 0$ and $c = 1$, so

$$\begin{aligned} y'_1 &= o \\ y'_{j+1} &= y_j \quad \text{for } 1 \leq j \leq n-1 \end{aligned}$$

For all rows r of Z we have

$$\begin{aligned} ry' &= 0y'_1 + \sum_{j=2}^{n} r_j y'_j = \sum_{j=2}^{n} r_j y'_j \\ &= \sum_{j=1}^{n-1} b_j y_j \\ &= by > 0 \end{aligned}$$

where b is the corresponding row of r in B.

We already had

$$ry' > 0 \quad \text{for all rows } r \text{ in } P \bigcup N.$$

So we have

$$\overline{A}_i y' > 0 \quad \text{for all } 1 \leq i \leq m$$

So

$$Ay' > 0$$

Let m be the envelope of $|A|_1$.

If $\mathcal{N}$ has a row r such that $ru \leq 0$ then we have to find positive integers w_A and c such that

- $w_A m$ is greater than the absolute value of $r_N uc$, where r_N is the row in $\mathcal{N}$ that has the greatest absolute value of the negative values ru for rows in $\mathcal{N}$.
- $w_A m$ is smaller/less than the value of $r_P uc$, where r_P is the row in $\mathcal{P}$ that has the smallest positive value for ru.

Let V_N be the set defined by

$$V_N = \{ru | r \in \mathcal{N}, ru \leq 0\}$$

Then define the nonnegative integer s by

$$s = -\min(V_N)$$

Also let V_P be defined by

$$V_P = \{ru | r \in \mathcal{P}, ru > 0\}$$

Then define the positive integer t by

$$t = \min(V_P)$$

Now, we proved earlier that

$$t > s$$

which is equivalent to

$$t - s > 0$$

which is euivalent to

$$t - s \geq 1$$

which is equivalent to

$$2(t - s) \geq 2$$

which is equivalent to

$$2t - 2s \geq 2$$

which is equivalent to

$$2t \geq 2 + 2s \tag{6.6}$$

We have

$$1 > 0 \tag{6.7}$$

From which it follows that

$$1 + 1 + 2s > 1 + 2s$$

which is equivalent to

$$2 + 2s > 1 + 2s \tag{6.8}$$

We also have, from equation (6.7)

$$1 + 2s > 2s \tag{6.9}$$

Combining equations (6.6) and (6.8) we have

$$2t > 1 + 2s \tag{6.10}$$

Let the positive integer m be the envelope of $\overline{\underline{A}}_1$. Then from equation (6.9) results

$$m(1 + 2s) > m \cdot 2s \tag{6.11}$$

And from equation (6.10) we have

$$m(2t) > m(1 + 2s) \tag{6.12}$$

Now, if we let

$$\begin{aligned} w_A &= 1 + 2s \text{ , and} \\ c &= 2m \end{aligned}$$

Then from equation (6.11) it follows that

$$mw_A > cs \tag{6.13}$$

From equation (6.12) it follows, with the definitions for w_A and c above,

$$ct > mw_A \tag{6.14}$$

So, from equations (6.13) and (6.14) it follows that c and w_A satisfy the conditions.

If $\mathcal{N} = N$ then we define y' as follows

$$\begin{aligned} y'_1 &= -w_A \\ y'_{j+1} &= cy_j \quad \text{for } 1 \leq j \leq n-1 \end{aligned}$$

Now, we have

$$w_A m < r_P uc$$

from which it follows that

$$r_P uc - w_A m > 0 \tag{6.15}$$

But we also have for every row r of $\mathcal{P}$

$$ruc \geq r_P uc$$

from which it follows that

$$ruc - w_A m \geq r_P uc - w_A m \tag{6.16}$$

So, combining equations (6.15) and (6.16) we get

$$ruc - w_A m > 0$$

which is equivalent to

$$\sum_{j=2}^{n} r_j \cdot y'_j + y'_1 \cdot r_1 > 0$$

which is equivalent to

$$ry' > 0 \quad \text{for all rows } r \text{ of } \mathcal{P} = P \tag{6.17}$$

Also for $\mathcal{N}$ we have

$$w_A m > |r_N u| \cdot c$$

from which it follows that

$$w_A m - |r_N u|c > 0 \tag{6.18}$$

But we also have

$$|ru|c \leq |r_N u|c$$

from which it follows that

$$-|ru|c \geq -|r_N u|c$$

from which it follows that

$$w_A m - |ru|c \geq w_A m - |r_N u|c \tag{6.19}$$

Combining equations (6.18) and (6.19) we get

$$w_A m - |ru|c > 0$$

which is equivalent to

$$(-w_A)(-m) + ruc > 0 \quad \text{(for all } r \text{ for which } ru \leq 0\text{)}.$$

which is equivalent to

$$\sum_{j=2}^{n} r_j y'_j + r_1 y'_1 > 0$$

which is equivalent to

$$ry' > 0 \text{ for all rows } r \text{ in } \mathcal{N} \text{ for which } ru \leq 0. \tag{6.20}$$

For the rows r of $\mathcal{N}$ for which $ru > 0$ we have

$$w_A m + |ru|c > 0$$

which is equivalent to

$$(-w_A)(-m) + \sum_{j=2}^{n} r_j y'_j > 0$$

which is equivalent to

$$y'_1 r_1 + \sum_{j=2}^{n} r_j y'_j > 0$$

which is equivalent to

$$\sum_{j=1}^{n} r_j y'_j > 0$$

which is equivalent to

$$ry' > 0 \text{ for all rows } r \text{ in } \mathcal{N} \text{ for which } ru > 0. \tag{6.21}$$

Combining equations (6.20) and (6.21) we get

$$ry' > 0 \text{ for all rows } r \text{ of } \mathcal{N} = N. \tag{6.22}$$

For all rows r of Z we have

$$\begin{aligned} ry' &= 0 \cdot y'_1 + \sum_{j=2}^{n} r_j y'_j \\ &= \sum_{j=2}^{n} r_j y'_j \\ &= c \cdot \sum_{j=1}^{n-1} b_j y_j \\ &= c(by) > 0 \end{aligned}$$

where b is the corresponding row of r in B.

So, we have

$$ry' > 0 \text{ for all rows } r \text{ of } Z. \tag{6.23}$$

Combining equations (6.17), (6.22), and (6.23) we have

$$\overline{\underline{A}}_i \cdot y' > 0 \text{ for all } 1 \leq i \leq m.$$

which means

$$Ay' > 0$$

Now, if $\mathcal{N} = P$ then we define y' as follows

$$\begin{aligned} y'_1 &= w_A \\ y'_{j+1} &= cy_j \quad \text{for } 1 \leq j \leq n-1 \end{aligned}$$

Now, we have

$$w_A m < r_P uc$$

from which it follows that

$$r_P uc - w_A m > 0 \tag{6.24}$$

But we also have for every row r of $\mathcal{P}$

$$ruc \geq r_P uc$$

from which it follows that

$$ruc - w_A m \geq r_P uc - w_A m \tag{6.25}$$

So, combining equations (6.24) and (6.25) we get

$$ruc - w_A m > 0$$

which is equivalent to

$$ruc + w_A \cdot (-m) > 0$$

which is equivalent to

$$\sum_{j=2}^{n} r_j y'_j + y'_1 r_1 > 0$$

which is equivalent to

$$ry' > 0 \text{ for all rows of } \mathcal{P} = N \tag{6.26}$$

Also for $\mathcal{N}$ we have

$$w_A m > |r_N u| c$$

from which it follows that

$$w_A m - |r_N u| c > 0 \tag{6.27}$$

But we also have

$$|ru|c \leq |r_N u|c$$

from which it follows that

$$-|ru|c \geq -|r_N u|c$$

from which it follows that

$$w_A m - |ru|c \geq w_A m - |r_N u|c \tag{6.28}$$

Combining equations (6.27) and (6.28) we get

$$w_A m - |ru|c > 0$$

which is equivalent to

$$w_A m + ruc > 0 (\text{for all } r \text{ for which } ru \leq 0.)$$

which is equivalent to

$$y'_1 r_1 + \sum_{j=2}^{n} r_j y'_j > 0$$

which is equivalent to

$$ry' > 0 \text{ for all rows } r \text{ in } \mathcal{N} \text{ for which } ru \leq 0. \tag{6.29}$$

For the rows r of $\mathcal{N}$ for which $ru > 0$, we have

$$w_A m + |ru|c > 0$$

which is equivalent to

$$w_A m + \sum_{j=2}^{n} r_j y'_j > 0$$

which is equivalent to

$$y'_1 r_1 + \sum_{j=2}^{n} r_j y'_j > 0$$

which is equivalent to

$$\sum_{j=1}^{n} r_j y'_j > 0$$

which is equivalent to

$$ry' > 0 \text{ for all rows } r \text{ in } \mathcal{N} \text{ for which } ru > 0. \tag{6.30}$$

Combining equations (6.29) and (6.30) we get

$$ry' > 0 \text{ for all rows } r \text{ of } \mathcal{N} = P \tag{6.31}$$

For all rows r of Z we have

$$\begin{aligned} ry' &= 0 \cdot y'_1 + \sum_{j=2}^{n} r_j y'_j \\ &= \sum_{j=2}^{n} r_j y'_j \\ &= c \cdot \sum_{j=1}^{n-1} b_j y_j \\ &= c(by) > 0 \end{aligned}$$

where b is the corresponding row of r in B.

So, we have

$$ry' > 0 \text{ for all rows } r \text{ of } Z \tag{6.32}$$

Combining equations (6.26), (6.31), and eq. (6.32) we have

$$\overline{A}_i y' > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$Ay' > 0$$

as required. ■

THEOREM **6.2** Let A be an $m \times n$ matrix and let B also be an $m \times n$ matrix, such that B is the result of applying the row multiplication action to A. Let y be such that $By > 0$. Then there is a y such that $Ay' > 0$.

PROOF

There is exactly one row of B that is a multiple of the corresponding row of A, that is there is a i such that

$$\overline{B}_i = c\overline{A}_i \tag{6.33}$$

where c is a positive integer. All other rows of B are copied verbatim from A. So, if $\overline{B}_j y > 0$ then also

$$\overline{A}_j y > 0 \text{ for all } j \neq i \text{ , and } 1 \leq j \leq m$$

Now, suppose $\overline{A}_i y \leq 0$, then multiply by the positive c,

$$c\overline{A}_i y \leq 0$$

But this contradicts with equation (6.33), so we have

$$\overline{A}_i y > 0$$

Together with the rows j we have

$$\overline{\underline{A}}_j y > 0 \quad \text{for all } 1 \leq j \leq m$$

So,

$$Ay > 0$$

And we can take $y' = y$. ■

THEOREM **6.3** Let A be an $m \times n$ matrix and let B also be an $m \times n$ matrix, such that A satisfies the mixed condition and B is the result of applying the tight paving action to A. Let y be such that $By > 0$. Then there is a y' such that $Ay' > 0$.

PROOF

For all rows i of B we have

$$\overline{\underline{B}}_i y > 0 \quad \text{for all } 1 \leq i \leq m \tag{6.34}$$

which is equivalent to

$$c_i(\overline{\underline{A}}_i) y > 0 \quad \text{for all } 1 \leq i \leq m \tag{6.35}$$

where c_i are the corresponding entries of the paving vector.

Now, suppose there is an i such that

$$(\overline{\underline{A}}_i) y \leq 0$$

then

$$c_i(\overline{\underline{A}}_i) y \leq 0$$

But from equation (6.35) it follows that

$$c_i(\overline{\underline{A}}_i) y > 0$$

So, this leads to a contradiction and we have to conclude that there is no i such that

$$(\overline{\underline{A}}_i) y \leq 0$$

which is equivalent to

$$(\overline{\underline{A}}_i y \not\leq 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$(\overline{\underline{A}}_i y > 0 \quad \text{for all } 1 \leq i \leq m \tag{6.36}$$

So, we can conclude from equation (6.36) that

$$Ay > 0$$

And we can take $y' = y$. ■

THEOREM **6.4** Let A be an $m \times n$ matrix and let B be a $q \times (n-1)$ matrix, such that A satisfies the reduceable condition and B is the result of applying the reduce action to A. Let y be such that $By > 0$. Then there is a y' such that $Ay' > 0$.

PROOF

Let m_i be defined for every row i of A for which $a_{i1} \neq 0$ as follows

$$m_i = 1 + \left| \sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} \cdot y_j \right|$$

Now, let m be defined as follows

$$m = \begin{cases} \max(m_i) & \text{if } |A|_1 \text{ is semipositive} \\ -\max(m_i) & \text{if } |A|_1 \text{ is seminegative} \end{cases}$$

Now, we have $1 > 0$, from which it follows that

$$1 + \left| \sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j \right| > \left| \sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j \right| \tag{6.37}$$

and we have by definition of modulus

$$|\sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j| \geq -\sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j \tag{6.38}$$

So, combining equations (6.37) and (6.38) we get

$$1 + \left| \sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j \right| > -\sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j \tag{6.39}$$

If $|A|_1$ is semipositive then $a_{i1} \geq 1$ for all i that we consider here. From $a_{i1} \geq 1$ it follows that

$$a_{i1} \cdot \left(1 + |\sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j| \right) \geq 1 + |\sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j| \tag{6.40}$$

Now, if we combine equations (6.39) and (6.40) we get

$$a_{i1} \cdot (1 + |\sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j|) > -\sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j \tag{6.41}$$

Finally we have

$$m \geq 1 + |\sum_{j=1}^{n-1} (\underline{\overline{A}}_i)_{j+1} y_j|$$

from which it follows that

$$a_{i1}m \geq a_{i1}(1 + |\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j|) \tag{6.42}$$

So, combining equations (6.41) and (6.42) we finally get

$$a_{i1}m > -\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j$$

from which it follows that

$$a_{i1}m + \sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j > 0 \tag{6.43}$$

So, if we define y' as

$$\begin{aligned} y'_1 &= m \text{ , and} \\ y'_{j+1} &= y_j \quad \text{for all } 1 \leq j \leq n-1 \end{aligned}$$

then (6.43) becomes

$$a_{i1}m + \sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y'_{j+1} > 0$$

which is equivalent to

$$a_{i1}m + \sum_{j'=2}^{n}(\overline{\underline{A}}_i)_{j'}y'_{j'} > 0$$

which is equivalent to

$$\sum_{j'=1}^{n}(\overline{\underline{A}}_i)_{j'}y'_{j'} > 0$$

which is equivalent to

$$\overline{\underline{A}}_i y' > 0 \tag{6.44}$$

So much for the rows of A where $a_{i1} \neq 0$, now we have to prove that y' also satisfies eq. (6.44) when i refers to a row of A where $a_{i1} = 0$. We have

$$\overline{\underline{A}}_i y' = \sum_{j=1}^{n} a_{ij}y'_j$$

which is equivalent to

$$\sum_{j=1}^{n} a_{ij}y'_j = a_{i1}y'_1 + \sum_{j=2}^{n} a_{ij}y'_j$$

But $a_{i1} = 0$, so

$$a_{i1}y'_1 + \sum_{j=2}^{n} a_{ij}y'_j = \sum_{j=2}^{n} a_{ij}y'_j$$

There is a row i' of B that has the same entries as row i of A for $j = 2$ and higher, so

$$\sum_{j=2}^{n} a_{ij}y'_j = \sum_{j=2}^{n} b_{i',j-1}y'_j$$

but $y_{j-1} = y'_j$ for $2 \leq j \leq n$, so

$$\sum_{j=2}^{n} b_{i',j-1}y'_j = \sum_{j=2}^{n} b_{i',j-1}y_{j-1}$$

rewrite this to

$$\sum_{j=2}^{n} b_{i',j-1}y_{j-1} = \sum_{j'=1}^{n-1} b_{i',j'}y_{j'} = \underline{B}_{i'} \cdot y$$

But $\underline{B}_{i'}y > 0$ for all $1 \leq i' \leq q$. So, we do have

$$\underline{A}_i y' > 0 \quad \text{for all rows } i \text{ for which } a_{i1} = 0.$$

So y' is a column vector such that

$$Ay' > 0$$

Now, if $|A|_1$ is seminegative then $a_{i1} \leq -1$ for all i that we consider here. We know that

$$|\sum_{j=1}^{n-1}(\underline{A}_i)_{j+1}y_j| \geq 0 \tag{6.45}$$

So, from equation (6.37) combined with eq. (6.45) it follows that

$$1 + |\sum_{j=1}^{n-1}(\underline{A}_i)_{j+1}y_j| > 0$$

from which it follows that

$$-(1 + |\sum_{j=1}^{n-1}(\underline{A}_i)_{j+1}y_j < 0 \tag{6.46}$$

Now, from $a_{i1} \leq -1$ and eq. (6.46) it follows that

$$a_{i1}(-(1 + |\sum_{j=1}^{n-1}(\underline{A}_i)_{j+1}y_j|)) \geq 1 + |\sum_{j=1}^{n-1}(\underline{A}_i)_{j+1}y_j| \tag{6.47}$$

Combining equations (6.47) and (6.37) it follows that

$$a_{i1}(-(1+|\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j|)) > |\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j| \tag{6.48}$$

Combining equations (6.48) and (6.38) it follows that

$$a_{i1}(-(1+|\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+}y_j|)) > -\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j \tag{6.49}$$

Now, because we assumed $|A|_1$ is seminegative the value of m is defined to be

$$m = -\max(m_i)$$

From which it follows that

$$m \leq (1+|\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j|)$$

from which it follows that (because $a_{i1} < 0$)

$$a_{i1}m \geq a_{i1}(-(1+|\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j|)) \tag{6.50}$$

So, combining equations (6.49) and (6.50) we get

$$a_{i1}m > -\sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j$$

from which it follows that

$$a_{i1}m + \sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y_j > 0 \tag{6.51}$$

So, if we define y' as

$$\begin{aligned} y'_1 &= m \text{ , and} \\ y'_{j+1} &= y_j \quad \text{for all } 1 \leq j \leq n-1 \end{aligned}$$

then eq. (6.51) becomes

$$a_{i1}m + \sum_{j=1}^{n-1}(\overline{\underline{A}}_i)_{j+1}y'_{j+1} > 0$$

which is equivalent to

$$a_{i1}m + \sum_{j'=2}^{n} (\overline{\underline{A}}_i)_{j'} y'_{j'} > 0$$

which is equivalent to

$$\sum_{j'=1}^{n} (\overline{\underline{A}}_i)_{j'} y'_{j'} > 0$$

which is equivalent to

$$\overline{\underline{A}}_i y' > 0 \tag{6.52}$$

So much for the rows of A where $a_{i1} \neq 0$, now we have to prove that y' also satisfies eq. (6.52) when i refers to a row of A where $a_{i1} = 0$. We have

$$\overline{\underline{A}}_i y' = \sum_{j=1}^{n} a_{ij} y'_j$$

which is equivalent to

$$\sum_{j=1}^{n} a_{ij} y'_j = a_{i1} y'_1 + \sum_{j=2}^{n} a_{ij} y'_j$$

But $a_{i1} = 0$, so

$$a_{i1} y'_1 + \sum_{j=2}^{n} a_{ij} y'_j = \sum_{j=2}^{n} a_{ij} y'_j$$

There is a row i' of B that has the same entries as row i of A for $j = 2$ and higher, so

$$\sum_{j=2}^{n} a_{ij} y'_j = \sum_{j=2}^{n} b_{i',j-1} y'_j$$

but $y_{j-1} = y'_j$ for $2 \leq j \leq n$, so

$$\sum_{j=2}^{n} b_{i',j-1} y'_j = \sum_{j=2}^{n} b_{i',j-1} y_{j-1}$$

rewrite this to

$$\sum_{j=2}^{n} b_{i',j-1} y_{j-1} = \sum_{j'=1}^{n-1} b_{i',j'} y_{j'} = \overline{\underline{B}}_{i'} \cdot y$$

But $\overline{\underline{B}}_{i'} y > 0$ for all $1 \leq i' \leq q$. So, we do have

$$\overline{\underline{A}}_i y' > 0 \quad \text{for all rows } i \text{ for which } a_{i1} = 0.$$

So y' is a column vector such that

$$Ay' > 0$$

So, we have proved that there is a y' such that

$$Ay' > 0$$

Which concludes our proof. ■

THEOREM **6.5** Let A be an $m \times n$ matrix and let B be an $m \times (n-1)$ matrix, such that A satisfies the zero condition, and B is the result of applying the elimination action to A. Let y be such that $By > 0$. Then there is a y' such that $Ay' > 0$.

PROOF

$By > 0$ is equivalent to

$$(\overline{\underline{B}}_i)y > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$\sum_{j=1}^{n-1} b_{ij} y_j > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$0 + \sum_{j=1}^{n-1} b_{ij} y_j > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$0 \cdot 0 + \sum_{j=1}^{n-1} b_{ij} y_j > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$a_{i1} \cdot 0 + \sum_{j=1}^{n-1} b_{ij} y_j > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$a_{i1} \cdot 0 + \sum_{j=1}^{n-1} a_{i,j+1} y_j > 0 \quad \text{for all } 1 \leq i \leq m \tag{6.53}$$

Now, if we introduce a new column vector y', where

$$\begin{aligned} y'_1 &= 0 \text{ , and} \\ y'_{j+1} &= y_j \quad \text{for all } 1 \leq j \leq n-1 \end{aligned}$$

then equation (6.53) becomes

$$a_{i1}y'_1 + \sum_{j=1}^{n-1} a_{i,j+1}y'_{j+1} > 0 \quad \text{for all } 1 \leq i \leq m$$

with $j' = j + 1 \;\Rightarrow\; j = j' - 1$ this is equivalent to

$$a_{i1}y'_1 + \sum_{j'=2}^{n} a_{ij'}y'_{j'} > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$\sum_{j'=1}^{n} a_{ij'}y'_{j'} > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$(\underline{\overline{A}}_i) \cdot y' > 0 \quad \text{for all } 1 \leq i \leq m$$

which is equivalent to

$$A \cdot y' > 0$$

With which the theorem is proved. ■

THEOREM **6.6** Let A be an $m \times n$ matrix and let B be a $q \times n$ matrix, such that A satisfies the reduceable condition and B is the result of applying the basic reduce action to A. Let y be such that $By > 0$. Then there is a y' such that $Ay' > 0$.

PROOF

Voor het bewijs nemen we het bewijs van theorem voor reductie over en passen dit aan. ■

THEOREM **6.7** Let A be an $m \times n$ matrix and let B be a $q \times n$ matrix, such that A satisfies the paved condition and B is the result of applying the basic expand action to A. Let y be such that $By > 0$. Then there is a y' such that $Ay' > 0$.

PROOF

Voor het bewijs nemen we het bewijs over van theorem voor expand en passen dit aan. ■

In this section we have shown that for every reduction rule, given a weight vector for the resulting matrix, we can construct a weight vector for the matrix to which the reduction rule is applied.

In the next section we will present a method and a set of rules. Together they will enable us to finf a weight vector for a matrix, that is not nullable, in a structured way.

6.2 Reverse Reduction Rules

With the results obtained in section 6.1 we can extend the reduction method such that we are able to construct a vector when the reduction ends in a positive or negative matrix.

DEFINITION **6.2** ***(reverse reduction method)***
Let R be a reduction of length n, such that the end matrix is positive or negative. Then the reverse reduction method constructs a sequence of weigth vectors $w_1, w_2, \ldots, w_n$, starting with w_n as follows:

1. If the end matrix is positive then let $w_n = (1)$, else if the end matrix is negative $w_n = (-1)$.

2. The weight vectors $w_1, w_2, \ldots, w_{n-1}$ are determined iteratively starting with w_{n-1} down to w_1, as follows.

   Each w_i is constructed from w_{i+1} by applying one of the reverse reduction rules, as follows. (The reverse reduction rules will be discussed further on in the current section.)

   (a) If M_{i+1} was constructed with the elim rule from M_i, then w_i is constructed with the reverse elim rule.

   (b) If M_{i+1} was constructed with the reduce rule from M_i, then w_i is constructed with the reverse reduce rule.

   (c) If M_{i+1} was constructed with the pave rule from M_i, then w_i is constructed with the reverse pave rule.

   (d) If M_{i+1} was constructed from M_i with the expand rule, then w_i is constructed with the reverse expand rule.

   (e) If M_{i+1} was constructed from M_i with the row multiplication rule, then w_i is constructed with the reverse row multiplication rule.

   (f) If M_{i+1} was constructed from M_i with the basic reduce rule, then w_i is constructed with the reverse basic reduce rule.

   (g) If M_{i+1} was constructed from M_i with the basic expand rule, then w_i is constructed with the reverse basic expand rule.

   (h) If M_{i+1} was constructed from M_i with the tight pave rule, then w_i is constructed with the reverse pave rule.

   The last rule is the same as reverse rule for 'normal' paving. Because of the way the new weight vector is constructed, there is no need for a separate (spelling?) reverse reduction rule for the tight paving rule.

□

We will describe the reverse reduction rules that are used in the reverse reduction method. The following schematic representation of the application of a reduction rule will be used in the description of the reverse reduction rules.

$$\begin{array}{ccc} A & & B \\ \uparrow & & \uparrow \\ M_i & \longrightarrow & M_{i+1} \\ \downarrow & \downarrow & \downarrow \\ w_i & \text{reduction rule } R & w_{i+1} \\ \downarrow & & \downarrow \\ y' & & y \end{array}$$

In the description of the reverse rules that follow let $B = M_{i+1}$ be the result of the application of a reduction rule R, and let $A = M_i$ be the matrix to which the reduction rule R is applied. Further, let $y = w_{i+1}$ be the weight vector for matrix $M_{i+1} = B$, and let $y' = w_i$ be the weight vector for matrix $M_i = A$.

6.2.1 reverse elim rule

Let y' be defined as follows

$$\begin{aligned} y'_1 &= 0 \\ y'_{j+1} &= y_j \quad \text{for } 1 \le j \le k \end{aligned}$$

where k is the length of y.

6.2.2 reverse reduce rule

Let the column vector q of length m be defined by

$$q_i = \begin{cases} 0 & \text{if } (|A|_1)_i = 0 \\ \frac{\sum_{j=1}^{n-1} (\overline{A}_i)_{j+1} \cdot y_j}{|(|A|_1)_i|} & \text{, otherwise} \end{cases}$$

Next, let the column vector q' of length m be defined by

$$q'_i = \begin{cases} 0 & \text{if } (|A|_1)_i = 0 \\ \lfloor |q_i| \rfloor + 1 & \text{if } (|A|_1)_i \neq 0 \text{ and } q_i \le 0 \\ 0 & \text{if } (|A|_1)_i \neq 0 \text{ and } q_i > 0 \end{cases}$$

Let the nonnegative integer d be defined by $d = \max(q'_i)$. Then let the intger w_A be defined by

$$w_A = \begin{cases} d & \text{if } |A|_1 \text{ is semipositive} \\ -d & \text{otherwise} \end{cases}$$

Then finally define y' as

$$\begin{aligned} y'_1 &= w_A \\ y'_{j+1} &= y_j \quad \text{for } 1 \le j \le n-1 \end{aligned}$$

6.2.3 reverse pave rule

Here y' equals y, so

$$y' = y$$

6.2.4 reverse expand rule

We define two sets, N and P, as follows

$$N = \{\underline{\overline{A}}_i | (\underline{\overline{A}}_i)_1 < 0\}$$

and

$$P = \{\underline{\overline{A}}_i | (\underline{\overline{A}}_i)_1 > 0\}$$

Further, let the column vector u with the same length as the rows of A, be defined by

$$\begin{aligned} u_1 &= o \\ u_{j+1} &= y_j \quad \text{for } 1 \leq j \leq n-1 \end{aligned}$$

Now, as was shown in theorem ?? at least one of the sets P and N have the property that $ru > 0$ for rows of that set. Lets call this set with that property $\mathcal{P}$, and let us call the other set $\mathcal{N}$.

Further let us define the following two sets

$$\begin{aligned} V_P &= \{ru | r \in \mathcal{P}\} \\ V_N &= \{ru | r \in \mathcal{N}\} \end{aligned}$$

Now define the two integers s_N and t as follows

$$\begin{aligned} s_N &= \min(V_N) \\ t &= \min(V_P) \end{aligned}$$

If $s_N > 0$ then let $w_A = 0$ and let $c = 1$, with which y' becomes

$$\begin{aligned} y'_1 &= 0 \\ y'_{j+1} &= y_j \quad \text{for all } 1 \leq j \leq n-1 \end{aligned}$$

If $s_N \leq 0$ then let $s = -s_N$ and let m be the envelope of $|A|_1$.
Now, w_A and c are determined in a possibly iterative fashion as follows.
Either one of the following situations applies for m

1. m is less or equal than s, i.e. $m \leq s$
2. m is in between s and t, i.e. $s < m < t$

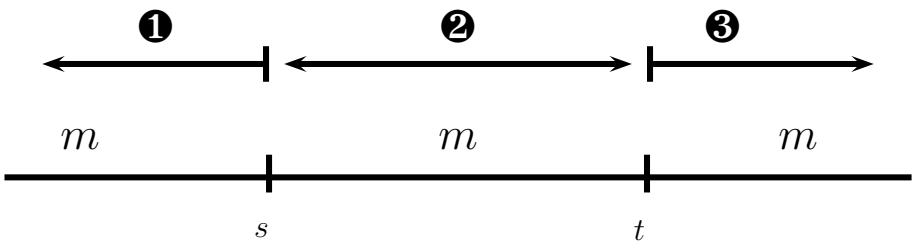

Figure 6.1: The three possible situations for the envelope of the first column.

3. m is greater or equal to t, i.e. $m \geq t$

In figure ? below we have depicted the situation with numbers for reference. Now, set $c' = 1$ and follow the following conditions to determine what the values for w_A and c should be

1. If m is less or equal s, then either one of the following situations applies. Either a multiple of m is in between s and t, or there is no such multiple.

   (a) If there is a multiple of m that is in between s and t then we choose the smallest d that satisfies this condition and let w_A be that value d, and let $c = 1 \cdot c'$.

   (b) If there is no multiple of m in between s and t, then we are going to multiply s and t by the smallest d such that there is a multiple of m between the products of s and the number and the product of t and the number. Let this number be called d, then

   $$d \leq \lfloor \frac{m}{|s-t|} \rfloor + 1$$

   So, we set c by this number, i.e. $c = d \cdot c'$, and we let w_A be the smallest multiplier p, such that that the product is in between s' and t'. In other words p is such that $s' < pm < t$, and there is no positive integer q such that $s' < qm < t'$ and $q < p$.

2. If m is between s and t, then we are already in the situation we want, so we set $w_A = 1$ and $c = 1 \cdot c'$

3. If m is greater or equal to t, then we have to multiply s and t with the smallest positive integer d such that for the products and m one of the situations 1 or 2 applies.

We then continue with construction items 1 and 2, but now with the products s' and t', instead of s and t, remembering that we already multiplied s and t by the multiplier. This is done through the use of the extra value c', which is considered in items 1 and 2 when constructing c. When we are ready to choose a value for c in items 1 or 2, we have to multiply by the old value. This is accomplished with the new value c'. So here, set $c' = d$, where d was the multiplier have just found. So,

$$c' = d$$

and continue at the top of the items list, i.e. with item 1 or 2, which ever satisfies.

Remark for item ii in 2: We want to accomplish that $|s' - t'| > m$, to make sure that a multiple of m always is in between s' and t'. Suppose that $|s - t| = h$, set $d_m = \lfloor \frac{m}{h} \rfloor + 1$, then $d_m h > m$, so $s' = d_m s$ and $t' = d_m t$, which implies $|s' - t'| = d_m s - d_m t| = d_m |s - t| = d_m h > m$. With which we have shown that d_m satisfies the required condition.

If $\mathcal{N} = P$ then let y' be defined by

$$\begin{aligned} y'_1 &= w_A \\ y'_{j+1} &= c \cdot y_j \quad \text{for all } 1 \leq j \leq n-1 \end{aligned}$$

If $\mathcal{N} = N$ then let y' be defined by

$$\begin{aligned} y'_1 &= -w_A \\ y'_{j+1} &= c \cdot y_j \quad \text{for all } 1 \leq j \leq n-1 \end{aligned}$$

6.2.5 reverse row multiplication rule

Here y' equals y, so

$$y' = y$$

6.2.6 reverse basic reduce rule

To obtain a value for y', use the reverse reduce rule with y taken to be $y[1:]$.

6.2.7 reverse basic expand rule

To obtain a value for y', use the reverse expand rule with y set to $y[1:]$.

Table ?? shows the available (spelling?) reverse reduction rules.

reverse reduction rule	apply if:	new weight vector
reverse expand	A $\downarrow$ expand rule B	$A \to y'$ $\uparrow$ $B \to y$
reverse basic expand	A $\downarrow$ basic expand rule B	$A \to y'$ $\uparrow$ $B \to y$
reverse reduce	A $\downarrow$ reduce rule B	$A \to y'$ $\uparrow$ $B \to y$
reverse basic reduce	A $\downarrow$ basic reduce rule B	$A \to y'$ $\uparrow$ $B \to y$
reverse paving	A $\downarrow$ any paving rule B	$A \to y'$ $\uparrow$ $B \to y$
reverse row multiplication	A $\downarrow$ row multiplication rule B	$A \to y'$ $\uparrow$ $B \to y$

Table 6.1: The reverse reduction rules.

7 Threshold Logic

CHAPTER

In the next two chapters we will introduce threshold logic and the threshold logic unit or device, starting in this chapter with a description of the threshold unit. We will apply the results we obtained in previous chapters to threshold logic. As will be shown we will obtain new insights.

7.1 Threshold Logic Device

A threshold logic device or threshold logic unit is a mathematical construct that functionally resembles Boolean logic gates. It has inputs that take as input binary valued signals, here represented as 0 and 1, just as we do for Boolean logic gates. Its output is also a binary valued signal and it too is represented by 0 and 1. Its output depends on its inputs and this dependence is described by a linear equation. In the following definition the threshold logic unit is described in a more precise way.

Definition **7.1** ***(TLU)***

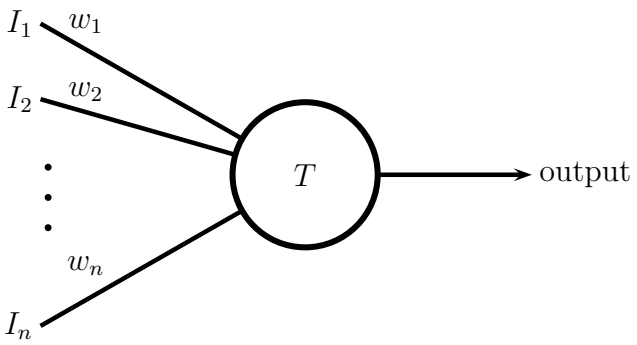

Figure 7.1: The Threshold Logic Unit, or TLU.

A Threshold Logic Unit (TLU) is a binary gate that takes binary valued inputs and produces a binary valued output. Outputs and inputs are elements of the binary set $\mathbb{B}$, where $\mathbb{B} = \{0, 1\}$.

Mathematically, the output of the TLU is given by

$$output = \text{sign}(\sum_{i=1}^{n} I_i \cdot w_i - T)$$

where

- n is the number of inputs of the TLU,
- I_i is the i^{th} input of the TLU,
- w_i is the i^{th} weight of the TLU, belonging to the i^{th} input. A weight is an integer.
- T is the threshold value of the TLU. The threshold value T is also an integer.

□

In figure 7.1 a schematic representation is shown of the threshold logic unit. Here on the left the inputs with weights are shown, and inside the circle is the threshold value of the TLU. On the right is the output.

There are two possible versions for the sign function which will be given shortly. Depending on the specific definition used we have two distinct definitions for the TLU.

DEFINITION **7.2** ***(Sign function)***

1. The function $\text{sign}^+ : \mathbb{Z} \to \mathbb{B}$ defined by

$$\text{sign}^+(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

2. The function $\text{sign}^- : \mathbb{Z}\backslash\{0\} \to \mathbb{B}$ defined by

$$\text{sign}^-(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}$$

□

Notice that we did not define a value for $\text{sign}^-(0)$, that is because the domain of that function doesn't include zero.

Now, depending on which function for sign we use we get two different TLUs

1. TLU^+, uses the sign^+ function

2. TLU^-, uses the sign^- function

We will for now use both definitions in what follows, but as will be shown later, for all practical purposes these two threshold logic units are the same.

Now, as we have seen, for every combination of input values the TLU defines an output, and effectively it defines a function on the set of tuples from $\mathbb{B}^n$. We will give an exact definition stating this, but first we have to define what exactly a Boolean function is.

DEFINITION **7.3** A function $f : \mathbb{B}^n \to \mathbb{B}$ is called a Boolean function. □

DEFINITION **7.4** Let t be a TLU with inputs $I_1, I_2, \ldots, I_n$, where the numbering of the inputs is as shown in figure 7.1. Further let $g : \mathbb{B}^n \to \mathbb{B}$ be a Boolean function.

If $g(i_1, i_2, \ldots, i_n)$ equals the output of the TLU t for all possible values of $i_1, i_2, \ldots, i_n$, where $I_1 = i_1$, $I_2 = i_2, \ldots$, and $I_n = i_n$, then g is called the representation of t. It is also said that g is implemented by t. □

We will now show that the two definitions for TLUs are, for all practical purposes, the same. We do that by showing that every Boolean function implemented by one TLU can also be implemented by the other TLU and vice versa.

It was found in the chapter on matrices and inequalities that it is not true in general that if we have found a column vector y such that the product Ay is a semipositive column vector that we then we can find a column vector x such that the product Ax is a positive vector.

Take for example

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{and } y = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

then $Ay = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, which is semipositive. Now, if it where true then there would be an x such that $Ax > 0$ so for the second row we would have $0x > 0$, which is euivalent to $0 > 0$ which is a contradiction. Hence the result stated.

Although this is true in general, we can however find such a relation when the matrix is a threshold matrix, for which we will give a definition later in this chapter. But the following theorem gives an equivalent result.

THEOREM **7.1** Let $g : \mathbb{B}^n \to \mathbb{B}$ be implemented by a TLU$^+$, then g can also be implemented by a TLU$^-$.

PROOF

Let $w_1, w_2, \ldots, w_n$ and T be the values of the weights and threshold respectively for the TLU$^+$ t, then let

$$\begin{aligned} w'_1 &= 2w_1, w'_2 = 2w_2, \ldots, w'_n = 2w_n, \text{ and} \\ T' &= 2T \end{aligned}$$

Then we have for every row of inputs either

- The sum of products of weigths w'_i and corresponding inputs and threshold T' is positive and greater than or equal to 2, or
- The sum of products of weights w'_i and corresponding inputs and threshold T' is negative and less than or equal to -2, or
- The sum of products of weigths w'_i and corresponding inputs and threshold T' is zero.

Now change the threshold value T' to $T' = 2T - 1$, then the sums all shift to the right by a value of one and we have

- The sum is positive and greater than or equal to 1, or
- It is less than or equal to -1

Note that a sum that was zero or positive and would produce a 1 with the old TLU$^+$, now has a positive value and still produces the required value of 1 with the new TLU$^-$. And the same is true when the value is negative, it will still be negative

after modification of the value of T', and so the old and the new TLU will produce the same, required, value at their output.

So, with the values for $w'_1, w'_2, \ldots, w'_n$, and T' we have found values for the TLU$^-$ such that it will produce the same output as the TLU$^+$. ■

THEOREM **7.2** Let the Boolean function $g : \mathbb{B}^n \to \mathbb{B}$ be implemented by a TLU$^-$, then there is a TLU$^+$ that implements g.

PROOF

This follows immediately from the definitions. ■

The theorems show that the TLU$^+$ definition does not define a more powerful version of a threshold logic unit in that it would implement more Boolean functions than a TLU$^-$. Both implement the same set of Boolean functions.

So, from now on we will use the definition of a TLU$^-$ as our definition of a TLU. So when we speak of a TLU, the definition of the TLU$^-$ is implied as the definition for this TLU.

Now that we have settled on one definition for a TLU, we need to know how we can find out if a Boolean function can be implemented by a TLU and what its weights and threshold value should be if it can implement the Boolean function. These issues will be addressed in the next section.

7.2 Threshold Matrix

If we want a TLU to implement a Boolean function we need to find the weights for the TLU such that for every input of the Boolean function its output will have the value of the Boolean function at that input.

The output of the TLU is determined by the sign$^-$ function. If the output has to be 1 then the sum of the products of the weights has to be greater than zero. If the output has to be 0 then the sum of the products of the weights with the inputs has to be less than zero.

So, we get a system of linear inequalities

$$\begin{aligned}
b_{11}w_1 + b_{12}w_2 + \ldots + b_{1n}w_n - T &> 0 \text{ (if } f(b_{11}, \ldots, b_{1n}) = 1 \text{)} \\
b_{21}w_1 + b_{22}w_2 + \ldots + b_{2n}w_n - T &< 0 \text{ (if } f(b_{21}, \ldots, b_{2n}) = 0 \text{)} \\
&\vdots \\
b_{m1}w_1 + b_{m2}w_2 + \ldots + b_{mn}w_n - T &< 0 \text{ (if } f(b_{m1}, \ldots, b_{mn}) = 0 \text{)}
\end{aligned}$$

With $T' = -T$, the system of linear inequalities becomes

$$\begin{aligned}
b_{11}w_1 + b_{12}w_2 + \ldots + b_{1n}w_n + T' &> 0 \text{ (if } f(b_{11}, \ldots, b_{1n}) = 1 \text{)} \\
b_{21}w_1 + b_{22}w_2 + \ldots + b_{2n}w_n + T' &< 0 \text{ (if } f(b_{21}, \ldots, b_{2n}) = 0 \text{)}
\end{aligned}$$

$$\vdots$$
$$b_{m1}w_1 + b_{m2}w_2 + \ldots + b_{mn}w_n + T' \;<\; 0 \text{ (if } f(b_{m1}, \ldots, b_{mn}) = 0 \text{)}$$

Now, this system of inequalities is equivalent to the system of inequalities that is obtained by multiplying each row by -1 if f is zero there. We obtain a system that looks like

$$\begin{aligned} b_{11}w_1 + b_{12}w_2 + \ldots + b_{1n}w_n + T' &> 0 \\ -b_{21}w_1 - b_{22}w_2 - \ldots - b_{2n}w_n - T' &> 0 \\ &\vdots \\ -b_{m1}w_1 - b_{m2}w_2 - \ldots - b_{mn}w_n - T' &> 0 \end{aligned}$$

We can write this system of inequalities as an equivalent multiplication between a matrix B and a vector w, where B is defined by

$$B = \begin{pmatrix} b_{11} & b_{12} & \ldots & b_{1n} & 1 \\ -b_{21} & -b_{22} & \ldots & -b_{2n} & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -b_{m1} & -b_{m2} & \ldots & -b_{mn} & -1 \end{pmatrix}$$

and w is a column vector

$$w = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ T' \end{pmatrix}$$

Now the system of inequalities in matrix notation becomes

$$Bw > 0 \tag{7.1}$$

So, finding weigths such that the TLU implements the Boolean function turns out to be equivalent to finding a solution w for inequality (7.1).

But here we recognize a familiar form that we have come accross before in previous chapters. We can use the results obtained in previous chapters to

- Show that the Boolean function can be implemented by a TLU or not.
- Find the weight vector if the Boolean function can be implemented by a TLU.

The matrix B is called the modulated threshold matrix. This matrix can be easily obtained from the function table as follows. First we add an extra column as the last column to the function table. This column is the column of the function values of the multiplier function of f for which the definition is given below.

DEFINITION **7.5** Let the Boolean function $f : \mathbb{B}^n \to \mathbb{B}$ be given, then the multiplication function of this function, denoted by $f^* : \mathbb{B}^n \to \mathbb{Z}$, is defined by

$$f^*(x_1, x_2, \ldots, x_n) = 2 \cdot f(x_1, x_2, \ldots, x_n) - 1$$

□

We then immediately remove this column again and we turn it into a colunm vector that looks exactly like the column in the table. That is, the i^{th} entry in this column in the table becomes the i^{th} entry of this new vector v, which is called the modulation vector.

Next we replace the colunm of the function values by a column of which all entries are 1. Now we view this table as a matrix Q, that is the i^{th} entry in column j becomes the i^{th} entry of column j in the matrix Q. This matrix Q is called the threshold matrix. Finally, the modulated threshold matrix B is obtained by row multiplication of Q and v, i.e. $B = B \otimes v$.

The process is illustrated below

I_1	I_2	F
t_{11}	t_{12}	f_1
t_{21}	t_{22}	f_2
t_{31}	t_{32}	f_3

I_1	I_2	F	F^*
t_{11}	t_{12}	f_1	$2f_1 - 1$
t_{21}	t_{22}	f_2	$2f_2 - 1$
t_{31}	t_{32}	f_3	$2f_3 - 1$

Next we obtain the modulation vector v,

$$v = \begin{pmatrix} 2f_1 - 1 \\ 2f_2 - 1 \\ 2f_3 - 1 \end{pmatrix}$$

and the function table becomes

I_1	I_2	
t_{11}	t_{12}	1
t_{21}	t_{22}	1
t_{31}	t_{32}	1

from which we derive the threshold matrix Q

$$Q = \begin{pmatrix} t_{11} & t_{12} & 1 \\ t_{21} & t_{22} & 1 \\ t_{31} & t_{32} & 1 \end{pmatrix}$$

Most of the time we have to add the output of other TLUs as an extra input to the TLU that generates the output. By realizing that this added TLU implements a Boolean function, we can add its values to the function table. We place them immediately before the values of our Boolean function F, so the column of F remains the last column of the table.

The threshold matrix for this case can be obtained in exactly the same way as we did when no extra columns were added to the function table. When no extra columns are added to the function table we call the threshold matrix that is obtained , the standard threshold matrix. When we do, however, add extra columns to the function table, the resulting threshold matrix is called the extended threshold matrix. When we add TLUs as extra input to a TLU we say we extend the threshold matrix.

Before multiplying with the modulation vector all our matrices have entries that are either 0 or 1. Formally they are called Boolean matrices, as defined in the following definition.

DEFINITION **7.6** An $m \times n$ matrix B, where all entries are from the Boolean set $\mathbb{B}$, that is $b_{ij} \in \mathbb{B}$, for all $1 \leq i \leq m$, and $1 \leq j \leq n$, is called a Boolean matrix. □

Every Boolean matrix that has a column where each entry is a 1, can serve as matrix B, after multiplication with the modulation vector, in inequality (7.1). We would, however, probably need to change the mapping of the weights and threshold value. Hence the following definition.

DEFINITION **7.7** A Boolean $m \times n$ matrix T is called a threshold matrix, if

1. One of the columns of T is the all-one column vector, and
2. $n \geq 2$.

□

Note that for all practical purposes we will only work with the threshold matrices that are the result of the procedure described earlier.

When we write the function table in the usual way and do not use additional functions in our table, that is there are no extensions, then the the threshold matrix will have a familiar form which is described in the following definition.

DEFINITION **7.8** The normal threshold matrix is a $2^n \times (n+1)$ threshold matrix T, representing the inputs of an n-input Boolean function, such that

1. The right-most column, i.e. column $n+1$, is the all-one vector.
2. Row i of $(\overset{\perp}{T})_{n+1}$ is the binary representation of the integer $i-1$, for all $1 \leq i \leq 2^n$.

□

EXAMPLE **4** The $2^3 \times (3+1)$ matrix T, given by

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

is the normal threshold matrix for a 3-input Boolean function when we build the function table in the usual way. □

Now, let us apply what we have discussed so far to two familiar Boolean functions, the AND-function and the NOT-function.

EXAMPLE **5** Let us examine if the NOT-function can be implemented by a TLU, and if so what the weights need to be. The function table is as shown below.

I_1	F
0	1
1	0

The modulated threshold matrix can be derived with the method described above and is given by

$$M = \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix}$$

Now we can apply the reduction method to the modulated threshold matrix M. We first apply the reduce rule. Then we are left with a single 1. Which is a positive 'column' and so the matrix closes. The function can be implemented by a TLU.

We will now apply the reverse reduction method. The column is positive so T' becomes $T' = 1$. Because we have applied a reduce rule to column 1, and the column is seminegative we have to multiply column 1 by a negative integer, which is $w_1 = -2$. So the TLU has $w_1 = -2$ and $T = -1$. The resulting TLU is given in figure 7.2. □

EXAMPLE **6** In this example we will examine if the AND-function can be implemented by a TLU. The function table is given below.

I_1	I_2	F
0	0	0
0	1	0
1	0	0
1	1	1

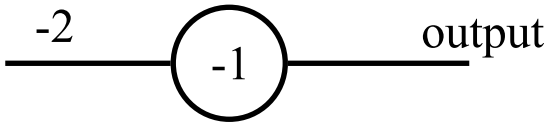

Figure 7.2: The NOT-function implemented by a TLU, with weights $w_1 = -2$ and threshold value $T = -1$.

From this table we can again derive the modulated threshold matrix with the procedure described earlier. The modulated threshold matrix M becomes

$$M = \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & -1 \\ -1 & 0 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

We can now apply the reduction method. We first apply the expand rule.

$$M' = \begin{pmatrix} 0 & -1 \\ -1 & -1 \\ 1 & 0 \end{pmatrix}$$

We can then again apply the expand rule to M', which becomes

$$M'' = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

Which is a negative column, so the the matrix closes and the Boolean function can be implemented by a TLU.

Next we apply the reverse reduction method to M''. The column is negative so T' becomes $T' = -1$. Before that we applied an expand rule, so we multiply T' by 2, which becomes $T' = -2$ and w_2 becomes $w_2 = 1$. Before that we also applied an expand rule so we have to multiply the existing values by 2, then T' becomes $T' = -4$ and w_2 becomes $w_2 = 2$. Then w_1 becomes $w_1 = 3$. So $w_1 = 3$, $w_2 = 2$, and $T = 4$. The resulting TLU is shown in figure 7.3. Note that we have only used the reverse expand rule, not its improvements. □

The next two theorems show that when we apply the reduction method, we obtain results that are consistent with results obtained through other methods. The first theorem shows that when we extend the threshold matrix with the Boolean function itself that then the function can be implemented by a TLU. Which is of course obvious because one of the inputs is the Boolean function itself. The second theorem shows that when we offer the complement of a Boolean function as input to a TLU then the original function can be implemented by the TLU, which is somewhat more of a surprise because the OR-gate is not capable of this without the help of other Boolean gates.

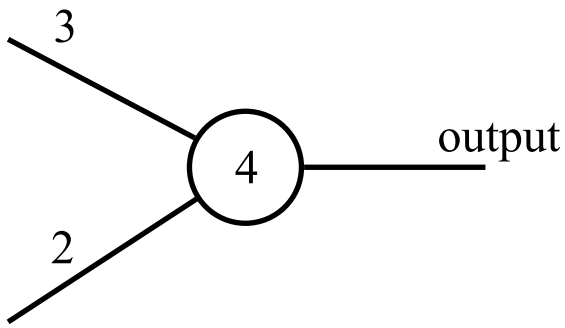

Figure 7.3: The AND-function implemented by a TLU, with weights $w_1 = 3$, $w_2 = 2$ and threshold value $T = 4$.

THEOREM **7.3** Let T be the $2^n \times (n+1)$ normal threshold matrix for the Boolean function $f : \mathbb{B}^n \to \mathbb{B}$, and let T' be the matrix T that is extended with the Boolean function itself. Then the modulated threshold matrix M of T' will close in the reduction method.

PROOF

Remember that the matrix M is constructed as follows. For each entry of f that is zero, the corresponding row of T' is multiplied by -1. Notice that the added column, that is f itself will not change, because the rows that were multiplied by -1 are exactly those rows that have a zero entry in the column of f.

Second, notice that the column vector in M that corresponds to the all-one vector in T' has entries equal to -1 in exactly those row positions where the function f has a corresponding zero entry. So this column vector can be obtained by replacing all zero entries of f by -1.

Third, the added column, that is the function itself, is not changed in the modulated threshold matrix M, as we mentioned earlier. That is, it is the same column vector as its corresponding column vector in T'.

According to the results obtained in the chapter on nullability we can exchange the positions of the added column vector and the first vector, without modifying the results. We can also exchange the positions of the column vector corresponding to the all-one vector and the second column vector in M.

Now, because the first vector, which is the column vector corresponding to the function f, is a semipositive column vector, we can apply the reduce rule to this vector. Then all rows where the function was zero will remain in the new reduction matrix. But as we stated earlier, the second column vector contained exactly entries that were 1 in the row positions that were removed, and it has entries that are -1 in the row positions that remained. So effectively the resulting column vector, which is now the first column vector, contains only entries that are -1. So, the resulting

first column is a negative column vector, and this means the resulting matrix is closed, and thus the Boolean function can be implemented by a TLU when we add the function itself as an extra input to this TLU. ■

For the following theorem we need to define a new function, which is given in the following definition.

DEFINITION **7.9** Let the Boolean function $f : \mathbb{B}^n \to \mathbb{B}$ be given, then the complement of this function, denoted by $f^{\mathsf{c}} : \mathbb{B}^n \to \mathbb{B}$, is defined by

$$f^{\mathsf{c}}(x_1, x_2, \ldots, x_n) = \begin{cases} 0 & \text{if } f(x_1, x_2, \ldots, x_n) = 1 \\ 1 & \text{if } f(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

□

THEOREM **7.4** Let T be the $2^n \times (n+1)$ normal threshold matrix for the Boolean function $f : \mathbb{B}^n \to \mathbb{B}$, and let T' be the matrix T that is extended with the complement of the Boolean function itself. Then the modulated threshold matrix M of T' will close in the reduction method.

PROOF

Remember that the matrix M is constructed as follows. For each entry of f that is zero, the corresponding row of T' is multiplied by -1. Notice that the added column, that is f^{c}, will change as follows. Because it is the complement of f, the entries that are multiplied by -1 are exactly the entries that are 1 because that is where f is zero. So this column vector becomes a seminegative column vector.

Second, notice that the column vector in M that corresponds to the all-one vector in T' has entries equal to -1 in exactly those row positions where the function f has a corresponding zero entry. So this column vector can be obtained by replacing all zero entries of f by -1.

According to the results obtained in the chapter on nullability we can exchange the positions of the added column vector and the first vector. We can also exchange the positions of the column vector corresponding to the all-one vector and the second column vector in M.

Now, because the first vector, which is the column vector corresponding to f^{c} is a seminegative column vector, we can apply the reduce rule to this vector. Then all rows where the function f^{c} was zero will remain in the new reduction matrix. But as we stated earlier the second column vector contains exactly entries that were -1 in the row positions that were removed, and it has entries that are 1 in the row positions that remained. So effectively the resulting column vector, which is now the first column vector, contains only entries that are 1. But this means the first column is a positive column vector and that the resulting matrix is closed, and thus the Boolean function can be implemented by a TLU when we add the complement as an extra input to this TLU. ■

When a Boolean function cannot be implemented by a single TLU we need to design a circuit of TLUs such that this will implement the desired Boolean function. This will be the topic of the next chapter. But first we will explain what a circuit is and we will give its formal definition.

7.3 TLU Circuits

Most Boolean functions cannot be implemented by a single TLU. But it is possible to implement a Boolean function if we add the output of other TLUs to the TLU. For instance the function given in table **??** is not implementable by a single TLU. This function is called the parity function because the function assumes the value one if and only if the number of inputs that are one is odd.

w	x	y	z	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Table 7.1: The odd-parity function with four inputs.

By adding certain functions that can themselves be implemented by a single TLU we can implement the parity function. These functions have the property that the value becomes one when the sum of the inputs that are one exceeds a certain predefined value. Obviously these functions are excellent candidates to be implemented by a single TLU.

In figure 7.4 we have shown the modulated threshold matrix of the extended threshold matrix, where the extra functions were added to the standard threshold matrix for the parity function. In the figure we have also given the reduction for this matrix. The values in the gray boxes are the values for the weights. They are

found by using the reverse reduction method, starting at the last matrix in the lower right corner of the figure and working backwards to the first matrix in the upper left corner of the figure.

As can be seen, it turns out that the value for w_6 is zero, i.e. $w_6 = 0$, and so the corresponding added function is not necessary in order for the matrix to close. So effectively we only needed to add two extra functions in order for the TLU to be able to implement the function.

Close examination of the example given above reveals a pattern. It seems that we only need the extra functions that become one when the sum exeeeds an *even* value. It turns out this is true for every odd parity function as was proved in Sorin Cotofana's Phd. dissertation.

By adding extra functions to the TLU we have actually constructed a circuit. Notice that constructing a circuit leads in natural way to the extended threshold matrix, which can then be reduced after we have modulated it with the Boolean function. In the next definition a formal definition for the circuit is given.

DEFINITION **7.10** A set of TLUs is called a circuit if:

1. It consists of more than one TLU.
2. Outputs of TLUs are connected to the inputs of other TLUs.
3. There is one TLU that does not have its output connected to any of the other TLU's in the circuit.
4. There are some TLUs that do not have their inputs connected to outputs of other TLUs in the circuit.

□

Sometimes we do not have TLUs with sufficient inputs. The following theorem shows how we can construct a circuit of TLUs that have a maximum number of inputs that is one less the number of inputs of the original TLU, and that implements the same function as the original TLU did.

THEOREM **7.5** Suppose we have an n-input TLU t. Then the function f that is implemented by t can be implemented by a circuit of $(n-1)$-input TLUs, where $n > 3$.

PROOF

Let's write the Boolean function f in the usual way as a table. In this proof we denote the most significant bit by x_n instead of the more usual x_{n-1}. So the least significant bit is denoted by x_1, and we number up to the most significant bit.

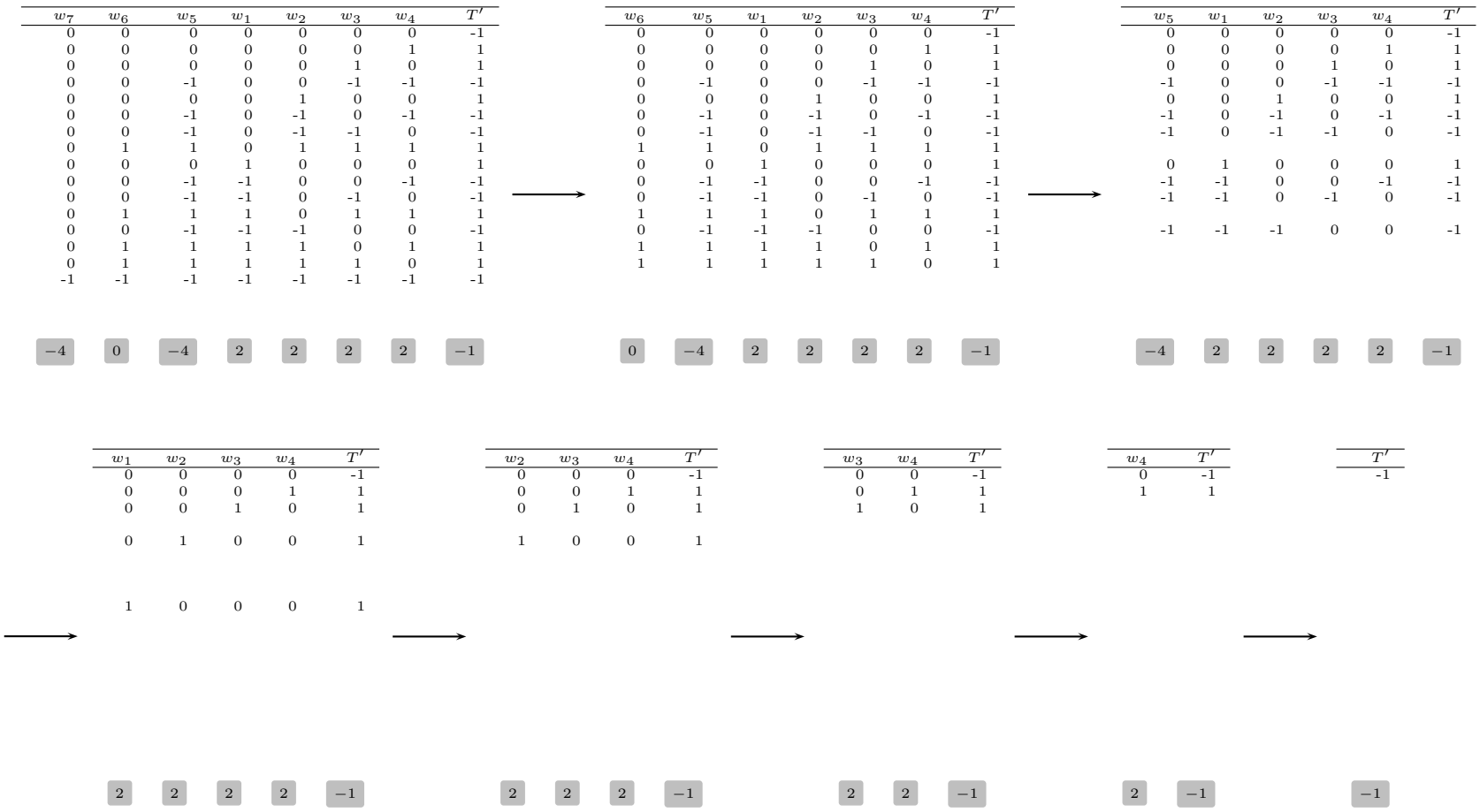

Figure 7.4: The modulated threshold matrix to which we apply the reduction method and the reverse reduction method. The values in gray are the values for the weights computed with the reverse reduction method.

Suppose that the upper part of the table, that is the part for which $x_n = 0$, can be implemented by a TLU t_a that has $n-1$ inputs. Similar assume that the lower part of the table, that is the part for which $x_n = 1$, can also be implemented by a TLU t_b, that also has $n-1$ inputs.

Now, examine the circuit depicted in figure 7.5. Note that if $w_b = 0$ and $x_n = 0$ then the circuit implements the upper part of the table, because the sum in the output TLU t_o is

$$\begin{aligned} \text{sum} &= 2t_a + w_b t_M - 2x_n - 1 \\ &= 2t_a - 1 \end{aligned}$$

Now, if $t_a = 1$ then the sum is equal to 1, and when $t_a = 0$ then the sum is equal to -1, so the sign function in t_o will give the desired result:

$$\begin{aligned} t_o &= 1 \quad \text{if } t_a = 1 \\ t_o &= 0 \quad \text{if } t_a = 0 \end{aligned}$$

Note that if $x_n = 0$ then $\overline{x}_n = 1$ and this will cause the sum in TLU M always to be negative, and so the output of M will be zero. Thus this is effectively the same as the assumption we made at the beginning by assuming that $w_b = 0$. In other words we can relax the condition $w_b = 0$ we made earlier, it is not necessary for the results above to be valid.

Now suppose that $x_n = 1$, then the circuit should give the lower part of the table at its output. Note that in this case t_a is still active. The result of t_a needs to be overruled by the result of t_b, i.e. it needs to be suppressed. We can do this by making the value of w_b big enough so that the result of t_a and its weight value at the output will not influence the final result. We will set the value of w_b to $w_b = 4$. Now the sum will be

$$\begin{aligned} \text{sum}_{\text{out}} &= w_b t_M - 2x_n + 2t_a - 1 \\ &= 4t_M - 2 \cdot 1 + 2t_a - 1 \end{aligned}$$

So, we have the following possible combinations and resulting sums

t_M	t_a	sum	t_o
0	0	-3	0
0	1	-1	0
1	0	1	1
1	1	3	1

Note that the output of t_M is just the output of t_b when $x_n = 1$, because the suppressing input $\overline{x}_n$ is not doing its work and the sum value will not be made extra negative in this case. So it follows that the sign function of TLU t_o again gives the desired results:

$$\begin{aligned} t_o &= 1 \quad \text{if } t_b = 1 \\ t_o &= 0 \quad \text{if } t_b = 0 \end{aligned}$$

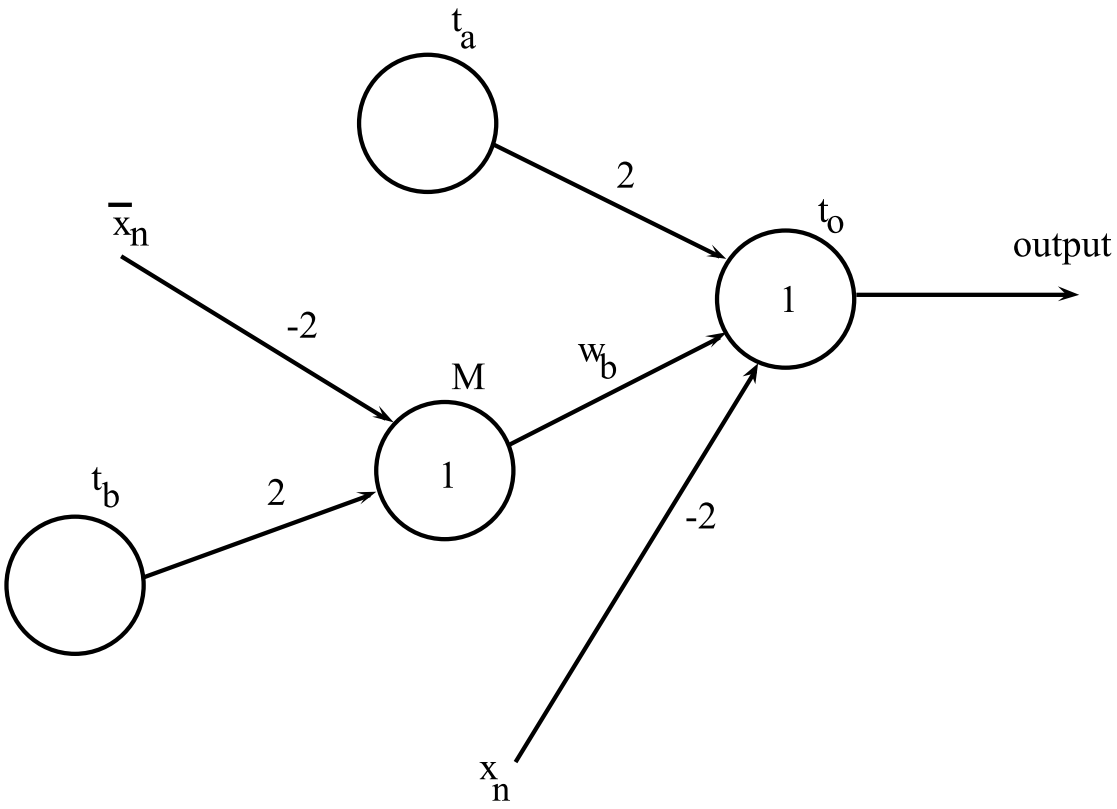

Figure 7.5: The circuit implementing the function of a TLU that has one more input than the TLUs in the circuit. So, one TLU is replaced by five new TLUs that have a maximum number of inputs one less that of the original TLU. The fifth TLU comes from the invertor producing $\overline{x}_n$. The weight w_b has value $w_b = 4$.

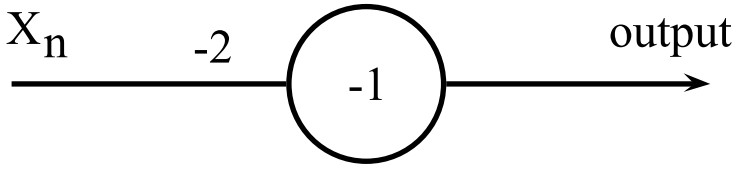

Figure 7.6: A TLU implementing a invertor.

Because x_n will always be one in this case, its weight value -2 assumes the role of an extra threshold value for TLU t_o. This extra threshold value is necessary to force the sum of t_o to a negative value when t_M is zero.

Now, remember that we assumed that there are TLUs with the number of inputs equal to $(n-1)$ that would implement the upper and lower parts of the table.

The TLU t_a should implement the upper part of the table. Notice that in that case $x_n = 0$, which means that the n^{th} weight of the original TLU does not participate in the determination of the sum part in that original TLU. Thus the upper part of the table is formed without the value of the n^{th} weight. So, we can just copy the values of the other weights and also copy the value of the threshold value of the original TLU, and use it in the construction of a new TLU with one less input.

Now, the other TLU, the one that has to implement the lower part of the table gets it values as follows. Note now that always $x_n = 1$ in this case. So, in effect the value of the n^{th} weight in the original TLU works as an extra threshold value. By combining this value with the value of the threshold into a new threshold value as follows

$$T' = T - w_n$$

we can implement the same function into a TLU that has only $(n-1)$ inputs. We can again take the values for the other weight vectors from the original TLU.

Note that an inversion can be implemented as shown in figure 7.6. The output will be as depicted in the table below.

x_n	sum	sign
0	$0+1=1$	1
1	$-2+1=-1$	0

■

So we can first construct a circuit for a Boolean function without paying attention to any restrictions that might exist on the number of inputs of a TLU. By using theorem 7.5 we can then map this circuit to a circuit of TLUs that have the proper number of inputs.

Note that we need TLUs that can handle at least three inputs. If that is a problem we can use the more conventional circuit shown in figure 7.7 which uses an extra TLU compared to the circuit in figure 7.5.

In the next chapter we will discuss several methods for constructing circuits in a structured manner.

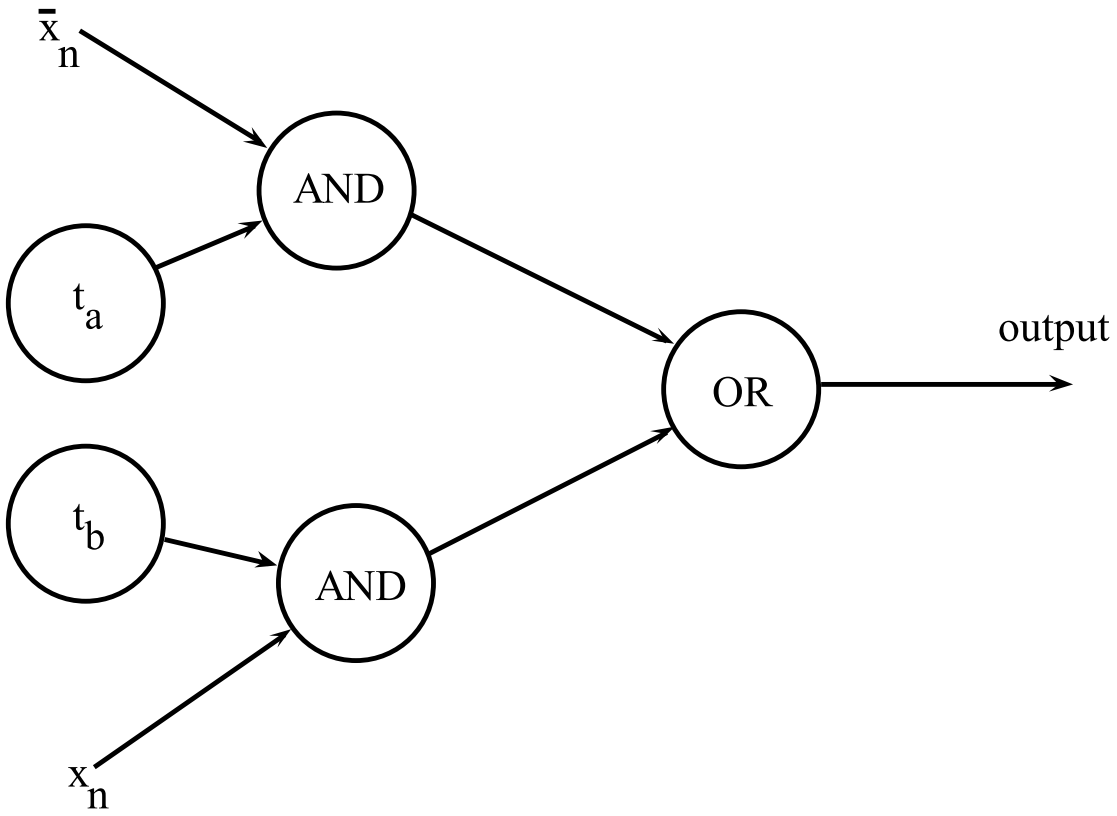

Figure 7.7: A more conventional TLU circuit replacing a TLU with more inputs. This circuit can be used in case there are no TLUs that have three inputs.

8 Circuit Synthesis

CHAPTER

As was shown in the previous chapter, if a Boolean function can be implemented by a TLU then the corresponding modulated threshold matrix will close. But only a few Boolean functions can be implemented by a single TLU. The majority of Boolean functions needs to be implemented by a circuit.

In conventional digital logic design we use the minterm approach and we would like to have an equivalent method for threshold logic.

In the first section we will discuss what would be necessary for such a method to be succesful. In the next three sections an approach is suggested for finding a set of input functions to a TLU such that the TLU can implement the required function. In the next section another approach for finding such an input set is presented which is more computational efficient but which lacks the exactness of the former approach mentioned earlier. Finally, the last section discusses minimization of the number of functions in the input sets.

8.1 The Minterm Approach

When implementing a Boolean function in digitial logic with Boolean gates we often use the minterm approach. In the minterm approach we bring a Boolean function in complete disjunctive normal form (CDNF). A function is in CDNF if it is described as a disjunction of minterms. A minterm is a conjunction of literals such that each variable of the function is represented exactly once. The number of terms is minimized by using the Karnaugh map method or the Quine-McCluskey method. Finally the result is mapped to a circuit of Boolean gates. This means that we are OR-ing a specific set S of minterms, where some minterms are combined into terms.

What makes this method so succeful is that we can easily determine which minterms to include in S and that we have methods that aid us in combining these minterms, resulting in less input to the OR-gate. The two most important methods are the Karnaugh mapping method and the Quine-McCluskey algorithm.

To summarize, we have

1. An output gate for which the output gives the desired Boolean function.

2. An 'easy' way to determine which minterms to include in the input set, out of the set of all possible minterms.

3. Methods that combine these minterms into other easily implementable terms.

It was already known that we could use the same approach when using TLUs. We would use the same set of minterms we would start with if we would be designing a conventional digital circuit. We could also use the same minimization methods. Using the minimized set of terms as input to a TLU, the TLU would perfectly be able to implement the required Boolean function.

But there is not much to be gained from using the TLU in this way, because essentially we are using the TLU as an OR-gate. To really use the extra potential of a TLU we need to know which sets of primary functions make that the corresponding modulated threshold matrix will close.

To summarize, when we use threshold logic gates we would like to have a comparable way to implement functions when compared to conventional digital logic design, where the problem of finding a suitable set of input functions is resolved. We need

1. A TLU as output.

2. A set of basic functions from which we can easily choose the right set S of basic functions such that the output TLU will close on this set.

3. Methods to combine these basic functions into other easily implementable functions so as to minimize the number of input functions.

In the next sections we show that there are methods to choose a set as input for a TLU. The last section of this chapter discusses minimization of the set of input functions.

8.2 The Characteristic Matrix

As we have shown in the preceding chapter, if a Boolean function can not be implemented by a single TLU then the reduction method will end in the empty matrix, that is the matrix will not close. We want to know for which Boolean function, when appended, the matrix will close. We could try all Boolean functions one by one but that would be impractical.

Instead we append an extra column vector to the threshold matrix, where instead of each entry being a zero or one, each entry is a variable x_i. These variables can take on the values 0 or 1. So the vector x of length m to be appended looks like this

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}$$

The resulting threshold matrix T looks like this

$$T = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} & 1 & x_1 \\ b_{21} & b_{22} & \cdots & b_{2n} & 1 & x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} & 1 & x_m \end{pmatrix}$$

Where as usual the values b_{ij} are determined by the Boolean function that is to be implemented.

Now, if we produce the modulated threshold matrix from T and perform the reduction method again up to, but not including, the new column vector then we are left with a matrix that has a single column. Each entry is a linear expression of the entries of x.

We can rewrite this result as a multiplication of a matrix K and the vector x, that is

$$\text{rest column vector} = K \cdot x \tag{8.1}$$

We call the matrix K the characteristic matrix for the Boolean function f.

EXAMPLE **7** Let us examine the function $F : \mathbb{B}^3 \to \mathbb{B}$ given by the following table.

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

The modulated threshold matrix with the extra column x is constructed from this table and is given below.

$$T = \begin{pmatrix} 0 & 0 & 0 & 1 & x_1 \\ 0 & 0 & -1 & -1 & -x_2 \\ 0 & -1 & 0 & -1 & -x_3 \\ 0 & -1 & -1 & -1 & -x_4 \\ -1 & 0 & 0 & -1 & -x_5 \\ 1 & 0 & 1 & 1 & x_6 \\ 1 & 1 & 0 & 1 & x_7 \\ -1 & -1 & -1 & -1 & -x_8 \end{pmatrix}$$

We will compute the characteristic matrix from this modulated threshold matrix in a number of steps. We start by applying the expand rule to the modulated threshold matrix, which gives

$$T' = \begin{pmatrix} 0 & 0 & 1 & x_1 \\ 0 & -1 & -1 & -x_2 \\ -1 & 0 & -1 & -x_3 \\ -1 & -1 & -1 & -x_4 \\ 0 & 1 & 0 & -x_5 + x_6 \\ -1 & 0 & 0 & x_6 - x_8 \\ 1 & 0 & 0 & x_7 - x_5 \\ 0 & -1 & 0 & x_7 - x_8 \end{pmatrix}$$

Again, we apply the expand rule to T'.

$$T'' = \begin{pmatrix} 0 & 1 & x_1 \\ -1 & -1 & -x_2 \\ 1 & 0 & -x_5 + x_6 \\ -1 & 0 & x_7 - x_8 \\ 0 & -1 & x_7 - x_5 - x_3 \\ -1 & -1 & x_7 - x_5 - x_4 \\ 0 & 0 & x_7 - x_5 + x_6 - x_8 \end{pmatrix}$$

Again, we apply the expand rule, but this time to T''.

$$T''' = \begin{pmatrix} 1 & x_1 \\ -1 & x_7 - x_5 - x_3 \\ 0 & x_7 - x_5 + x_6 - x_8 \\ -1 & -x_2 - x_5 + x_6 \\ 0 & -x_5 + x_6 + x_7 - x_8 \\ -1 & -x_5 + x_6 + x_7 - x_5 - x_4 \end{pmatrix}$$

Finally, we once more apply the expand rule, but this time to T'''.

$$T'''' = \begin{pmatrix} x_7 - x_5 + x_6 - x_8 \\ -x_5 + x_6 + x_7 - x_8 \\ x_1 + x_7 - x_5 - x_3 \\ x_1 - x_2 - x_5 + x_6 \\ x_1 - x_5 + x_6 + x_7 - x_5 - x_4 \end{pmatrix}$$

Which can be written as

$$T'''' = \begin{pmatrix} & & & & -1 & 1 & 1 & -1 \\ & & & & -1 & 1 & 1 & -1 \\ 1 & & -1 & & -1 & & 1 & \\ 1 & -1 & & & -1 & 1 & & \\ 1 & & & -1 & -2 & 1 & 1 & \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix}$$

Where we have omitted the zeros for clarity. From this it follows that the characteristic matrix K is given by

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & -2 & 1 & 1 & 0 \end{pmatrix} \tag{8.2}$$

□

What does this characteristic matrix mean? Well, suppose that instead of the column vector x we appended a Boolean function g to the threshold matrix T. If we would perform the same sequence of reduction rules to this extended matrix, after modulation, then we would obtain the same rest column vector as when we would have performed the multiplication with K and g. In other words we substitute the values of g for the entries of x in equation (8.1).

Notice that this also works if we were to append not one but several Boolean functions to the threshold matrix. The resulting columns after applying the same sequence of reduction rules can be obtained by applying the multiplication with the characteristic matrix K and the original appended Boolean functions. Which in other words is a multiplication between matrices K and F, where F is the matrix formed of the appended Boolean functions.

So, by choosing values for the entries of x, we are implicitly choosing a Boolean function that is appended to the threshold matrix. We do not have to perform the reduction method each time we choose another Boolean function, but instead only have to perform the multiplication given by equation (8.1). By choosing the values for x_i carefully we can produce a positive or negative column. In the next section we will investigate how we can choose these values for x_i. Let us clarify this in the following example.

EXAMPLE **8** This example shows that instead of every time performing the reduction method after we have chosen new values for the entries of x, we only need to perform a matrix multiplication of K_F and x.

Consider the Boolean function given in the table below n the left, which is of course the familiar XOR function. On the right is its modulated threshold matrix where the extra vector is already appended.

		F
0	0	0
0	1	1
1	0	1
1	1	0

$$T = \begin{pmatrix} 0 & 0 & -1 & -x_1 \\ 0 & 1 & 1 & x_2 \\ 1 & 0 & 1 & x_3 \\ -1 & -1 & -1 & -x_4 \end{pmatrix}$$

Apply the reduction method to T. This gives the following sequence of reductions.

$$T_1 = \begin{pmatrix} 0 & -1 & -x_1 \\ 1 & 1 & x_2 \\ -1 & 0 & x_3 - x_4 \end{pmatrix} \quad \rightarrow \quad \begin{pmatrix} -1 & -x_1 \\ 1 & x_2 + x_3 - x_4 \end{pmatrix} \quad \rightarrow$$

$$T_3 = \begin{pmatrix} -x_1 + x_2 + x_3 - x_4 \end{pmatrix}$$

So, the characteristic matrix K_F is

$$K_F = \begin{pmatrix} -1 & 1 & 1 - 1 \end{pmatrix}$$

Letting a single entry of x be 1 is sufficient to let T close. Let us set $x_2 = 1$ for example and let the other values be zero, i.e. $x_1 = x_3 = x_4 = 0$. Now let us again perform the reduction method but this time we will use the chosen values for the entries of x.

$$T' = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 \end{pmatrix} \quad \rightarrow T_1' \begin{pmatrix} 0 & -1 & 0 \\ 1 & 1 & 1 \\ -1 & 0 & 0 \end{pmatrix}$$

$$\rightarrow T_2' = \begin{pmatrix} -1 & 0 \\ 1 & 1 \end{pmatrix} \quad \rightarrow T_3' = \begin{pmatrix} 1 \end{pmatrix}$$

which means it is closed. □

8.3 Properties of the Characteristic Matrix

In the previous section we showed how we can use the characteristic matrix of a Boolean function to produce the result of applying the reduction method. In this section we will investigate the characteristic matrix itself. We will start by showing that the columns have some very interesting properties.

Remember that in the reduction rules of the reduction method we never multiply by a negative number, i.e. we always multiply rows by a positive number. Furthermore, we always add rows, we never subtract rows. After modulation, the variables x_i of x, that is the entries of the appended vector x, have either a minus sign or not.

Suppose that the j^{th} entry of x gets a minus sign, then after we have applied all reduction rules the entries in the single column matrix that contain x_j have terms containing x_j with negative coefficients. So this leads to the following theorem.

THEOREM **8.1** Let K_f be the characteristic matrix belonging to the Boolean function $f : \mathbb{B}^q \rightarrow \mathbb{B}$ that can not be implemented by a single TLU. Then for all columns of K_f we have that all nonzero entries that belong to the same column have the same sign.

PROOF

This is because of the way the rules of the reduction method were designed. Multiplication of rows is always with positive integers, and rows are always added, never subtracted from eachother. ■

DEFINITION **8.1** Let K be a characteristic matrix, and let v be a column vector of K. Then v is called a

1. **poscol** if the nonzero entries of v are positive.
2. **negcol** if the nonzero entries of v are negative.

□

So, if we have a Boolean function $f : \mathbb{B}^q \rightarrow \mathbb{B}$ and K_f is its characteristic matrix then if the i^{th} entry of f is equal to 1, i.e. $f(x) = 1$, then column i of K_f is a poscol. If the i^{th} entry of f is equal to 0, i.e. $f(x) = 0$, then column i of K_f is a negcol.

Until now we have said we could find Boolean functions that close the modulated threshold matrix by carefully choosing the right values for x_i. But from the previous

chapter we already know some functions that will close the modulated threshold matrix, namely the Boolean function itself and its complement. This is used to obtain the following result.

THEOREM **8.2** Let K_f be the characteristic matrix of a Boolean function $f : \mathbb{B}^q \to \mathbb{B}$ that is not implementable by a single TLU. Then the set of columns of K_f that are poscol, when added together is a positive column vector.

PROOF

As we saw in chapter 7 the Boolean function f itself will close the matrix. So if we substitute this function f for x and multiply with K_f we would obtain a positive or negative column. But lets look more closely to the function f represented as a vector x. When multiplied with K_f we can write this as

$$\begin{aligned} \text{sum} &= \sum_i x_i |K_f|_i \\ &= \sum_{x_i=1} x_i |K_f|_i \end{aligned}$$

But this is exactly the sum of all the poscol columns of K_f. Because we have positive or semipositive columns this means we get entries of the resulting sum vector that are zero or positive. But we noticed earlier that we get a negative or a positive sum vector. So together this means we get a positive sum vector. And this proves the result that was required. ■

A similar result can be obtained for all negcols if we use the fact that the complement of the Boolean function to be implemented also closes the modulated threshold matrix when appended. The next theorem states this result.

THEOREM **8.3** Let K_f be the characteristic matrix of a Boolean function $f : \mathbb{B}^q \to \mathbb{B}$ that is not implementable by a single TLU. Then the set of columns of K_f that are negcol, when added together is a negative column vector.

PROOF

As we saw in chapter 7 the complement $\overline{f}$ of the Boolean function f will close the matrix. So if we substitute this function $\overline{f}$ for x and multiply with K_f we would obtain a positive or negative column. But lets look more closely to the function $\overline{f}$ represented as a vector x. When multiplied with K_f we can write this as

$$\begin{aligned} \text{sum} &= \sum_i x_i |K_f|_i \\ &= \sum_{x_i=1} x_i |K_f|_i \end{aligned}$$

But this is exactly the sum of all the negcol columns of K_f, because if $x_i = 1$ then the corresponding entry of f is zero, which means that the corresponding column of K_f is a negative column. Because we have negative or seminegative columns this means we get entries of the resulting sum vector that are zero or negative. But we noticed earlier that we get a negative or a positive sum vector. So together this means we get a negative sum vector. And this proves the result that was required. ■

8.4 Finding a Cover

As was shown in the previous section the sum of certain combinations of columns from the characteristic matrix is positive or negative. In this section we will investigate which other combinations of columns of the characteristic matrix have this property as well. We will start with a definition that will make this notion exact.

DEFINITION **8.2** ***(Cover)***
Let $f : \mathbb{B}^q \to \mathbb{B}$ be a Boolean function and let the $m \times n$ matrix K_f be its characteristic matrix. Let the set $V = \{v_1, v_2, \ldots, v_p\}$ be a subset of the columns of K_f. Then V is called a *cover* if for every i, with $1 \leq i \leq m$, there is a vector $v \in V$ such that the i^{th} entry of v is a nonzero entry.

The cover V is called homogeneous if one of the following conditions holds.

- All entries of the columns in V are either zero or positive, or
- All entries of the columns in V are either zero or negative

A cover that is not homogeneous is called heterogeneous. □

EXAMPLE **9** Consider the function $F : \mathbb{B}^3 \to \mathbb{B}$ again that was given in example 7. Its characteristic matrix was given by K in equation 8.2. Now a cover would be for example $V_1 = \{v1, v6\}$, but also $V_2 = \{v_6, v_7\}$, yet another would be $V_3 = \{v_1, v_8\}$, yet still another would be $V_4 = \{v_2, v_3, v_4, v_8\}$. A cover containing only one column is $V_5 = \{v_5\}$. □

THEOREM **8.4** Let $f : \mathbb{B}^q \to \mathbb{B}$ be a Boolean function and let the $m \times n$ matrix K be its characteristic matrix. Let the set $V = \{v_1, v_2, \ldots, v_k\}$ be a subset from the set of columns of K. Let the $m \times k$ matrix $M = (v_1, v_2, \ldots, v_k)$ be formed from the columns of V, i.e. $|M|_i = v_i$, for $1 \leq i \leq k$. Then M is not nullable if and only if V is a cover.

PROOF

Suppose M is not nullable, then M will close when we apply the reduction method to M. After applying zero or more reduction rules a positive or negative column will

result. Because the columns of M are columns of the characteristic matrix K, they can be positive, semipositive, negative, or seminegative. So either we have found the closing column, or we can apply the reduce rule.

Every time the reduce rule is applied, one or more rows will be removed from the matrix. The rows that were removed had entries unequal zero. So, if we tag every row with its row number and start with a set of numbers containing every row number, then we can remove the corresponding row numbers from this set when the rows are removed from the matrix by the reduce rule. So, if a row number is not present in the set anymore it means that row has a nonzero entry. When we encounter a positive or negative column the matrix closes and exactly the remaining rows were not yet removed. Because all entries are nonzero in the last step all row numbers can be removed from the set and the set becomes the empty set and so all rows have a nonzero entry.

Now suppose M is a cover, then every row of M contains a nonzero entry. Because the columns of M are either positive, semipositive, negative, or seminegative we can apply the reduce rule each time. Suppose the matrix will not close, then this means that there are rows for which all entries are zero. Because note that only rows will remain for which the entries were zero when removing the column as a result of applying the reduce rule. This means that we have not yet encountered a nonzero entry for these rows. But this contradicts the fact that all rows have at least one nonzero entry. ■

DEFINITION **8.3** A Boolean function $f : \mathbb{B}^q \rightarrow \mathbb{B}$ is called a primary Boolean function if it has exactly one function value that is equal to one and the rest of the function values is zero. In other words there is a tuple $(b_{q-1}, b_{q-2}, \ldots, b_0)$ such that $f(b_{q-1}, b_{q-2}, \ldots, b_0) = 1$ and $f(x_{q-1}, x_{q-2}, \ldots, q_0) = 0$ if $(x_{q-1}, x_{q-2}, \ldots, q_0) \neq (b_{q-1}, b_{q-2}, \ldots, b_0)$.

The primary Boolean function for which $f(b_{q-1}, b_{q-2}, \ldots, b_0) = 1$ is designated by $\underline{B}_n$, where $n = b_{q-1} \cdot 2^{q-1} + b_{q-2} \cdot 2^{q-2} + \ldots + b_0 \cdot 2^0 + 1$. □

We often will call a primary Boolean function just a primary function, thereby leaving out the word Boolean.

EXAMPLE **10** The primary function $f : \mathbb{B}^3 \rightarrow \mathbb{B}$, where $f(0, 0, 0) = 1$ is denoted by $\underline{B}_1$. □

EXAMPLE **11** The primary function $q : \mathbb{B}^4 \rightarrow \mathbb{B}$, where $q(1, 0, 0, 1) = 1$ is denoted by $\underline{B}_{10}$, because

$$\begin{aligned} 10 &= 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \\ &= 8 + 1 + 1 \end{aligned}$$

□

THEOREM **8.5** Let $f : \mathbb{B}^p \to \mathbb{B}$ be a Boolean function and let the $m \times n$ matrix K be its characteristic matrix. Let the set $V = \{v_1, v_2, \ldots, v_q\}$ be a subset of the columns of K. Let $B = \{b_1, b_2, \ldots, b_q\}$ be the corresponding primary functions for the columns in V, i.e. b_i is the primary function such that the multiplication $Kb_i = v_i$ holds if we consider the primary function as a column vector. Let T be the threshold matrix extended with the functions of B. Then the modulated threshold matrix T' that is formed from T is not nullable if and only if V is a cover.

PROOF

Suppose the matrix T' is not nullable, then the matrix T' will close when the reduction method is applied to T'. Perform the first reduction rules until we are about to perform a reduction rule to the first of the appended primary functions. The matrix M that results is a matrix where all columns are columns of K, i.e. $|M|_i = v_i$. Because this matrix will close eventually, the matrix is not nullable. We can therefor apply theorem 8.4, resulting in V being a cover.

Now suppose that V is a cover, then by theorem 8.4 the matrix $M = (v_1, v_2, \ldots, v_k)$ is not nullable. But this matrix M is obtained by performing the reduction method to T', so T' is also not nullable. ■

DEFINITION **8.4** ***(Minimum Cover)***
Let $f : \mathbb{B}^q \to \mathbb{B}$ be a Boolean function and let the $m \times n$ matrix K_f be its characteristic matrix. Let the set $V = \{v_1, v_2, \ldots, v_n\}$ be a cover for K_f. We call V a minimum cover if for all covers V' the number of columns in V' is greater or equal to n. Here n is the number of columns in V. We call the number of columns in V the *designation* of f. □

EXAMPLE **12** See the function in example 9. The cover given by V_5 is a minimum cover for function F. So, its designation is $n = 1$. □

COROLLARY **8.6** Let $f : \mathbb{B}^p \to \mathbb{B}$ be a Boolean function and let the integer n be the designation of f. Let $b_1, b_2, \ldots, b_k$ be primary functions that are appended to the threshold matrix T of f. If the modulated threshold matrix T' formed from T is not nullable then $k \geq n$.

PROOF

Suppose the modulated threshold matrix T' is not nullable. Let $V = \{v_1, v_2, \ldots, v_k\}$ be the column vectors of the characteristic matrix K of f, where v_i correspond to the primary functions b_i. Suppose $k < n$, from theorem 8.5 it follows that V is a cover. Because n is the designation of f, we have that all covers have n or more columns. So $k \geq n$. But this contradicts with $k < n$. So we must conclude that $k \geq n$. ■

Theorem 8.5 and its corollary are very important results because they show that for each Boolean function there is a minimum number of primary Boolean functions with which the function can be implemented.

This is the result we are after as was noted in the first section. By finding a minimum cover for the characteristic matrix belonging to the Boolean function to be implemented we obtain a set of corresponding primary functions with which we can implement the Boolean function as a depth-two circuit.

In a later section of this chapter we will discuss minimization of the set of primary functions.

The following theorem shows that the results obtained are independent of the way we obtained the characteristic matrix belonging to the Boolean function.

THEOREM **8.7** Let $f : \mathbb{B}^q \rightarrow \mathbb{B}$ be a Boolean function. Let Q_1 and Q_2 be characteristic matrices of f, obtained through different reductions. Then the columns $v_{k_1}, v_{k_2}, \ldots, v_{k_n}$ of Q_1 form a cover if and only if the corresponding columns of Q_2, i.e. $v'_{k_1}, v'_{k_2}, \ldots, v'_{k_n}$ in Q_2 form a cover.

PROOF

Let $b_{k_1}, b_{k_2}, \ldots, b_{k_n}$ be the primary functions that correspond to the columns $v_{k_1}, v_{k_2}, \ldots, v_{k_n}$ of Q_1. Let $V = \{v_{k_1}, v_{k_2}, \ldots, v_{k_n}\}$ be a cover, then the threshold matrix extended with the primary functions will close after being modulated according to theorem 8.5. According to theorem 8.5 this also means that $V = \{v'_{k_1}, v'_{k_2}, \ldots, v'_{k_n}\}$ of Q_2 will be a cover.

A similar argument proves the reverse proces. ■

8.5 The Mask Method

Although computation of a suitable set by means of the characteristic matrix leads to exact results, it can be too elaborate in computational terms. We would like a method to determine an input set that is less elaborate and that gives satisfactory results.

In this section we will propose a method called the mask method that will give a satisfactory input set that will close the TLU too, but it will not necessarily lead to a minimum set of input functions, i.e. it can lead to a set that contains more primary functions than strictly necessary.

Primary functions have an interesting property. When a primary function is appended to a threshold matrix and the modulated threshold matrix is determined, the entry of the primary function that is one becomes in the modulated threshold matrix either a -1 or a 1. Because primary functions have exactly one entry that is one, this means that primary functions become semipositive or seminegative columns in modulated threshold matrices.

We know that the reduce rule operates on semipositive or seminegative columns. So we can apply the reduce rule to the appended primary functions, thereby effectively removing the row designated by the primary function.

Now, by choosing the right primary functions we can, in a number of steps, transform a mixed column into a seminegative or semipositive column. We can then apply the reduce rule to this resulting column as well.

By repeating this process several times we can transform every mixed column into a semipositive or seminegative column and remove it with the reduce rule, until we have found a positive or negative column which is what we require. The total set of primary functions that are appended form the cover.

Let us explain this with the following example.

EXAMPLE **13** Examine the following threshold matrix where the Boolean function is given as an extra columm. Note that this is the same function that was used in the examples of the previous chapter, where it was used to compute its characteristic matrix.

	x_3	x_2	x_1	T'	F
1	0	0	0	1	1
2	0	0	-1	-1	0
3	0	-1	0	-1	0
4	0	-1	-1	-1	0
5	-1	0	0	-1	0
6	1	0	1	1	1
7	1	1	0	1	1
8	-1	-1	-1	-1	0

Look at column x_1. It has one 1 in row 6, the rest is either 0 or -1. So, if that row wouldn't be there we could apply the reduce rule to column x_1. So, we add the primary function B_6, which is a column of all zeros except for row 6, where a 1 is placed. In the threshold matrix above it will turn up as the same column, that is, a 1 in row 6 and the rest all zeros. We apply the reduce rule to this newly added column. Now column x_1 has only zero or one entries, so we can apply the reduce rule to this column as well. We do not actually change the matrix as we are accustomed to in the reduction method, instead we make a mental note that we have removed certain rows. So for our purposes rows 2, 4, 6, and 8 are removed from this threshold matrix, leaving rows 1, 3, 5, and 7 still to be considered.

Next, let us add a 1 to row 7, that is, add primary function B_7. Then we can apply the reduce rule to this added column, effectively removing row 7. Now we can apply the reduce rule to columm x_3, thereby removing row 5, leaving us only with rows 1 and 3. Without adding any more primary functions we can now apply the reduce rule to column x_2, removing row 3. Now we are left with row 1 which has a positive 'column' as the last column, and we can stop adding primary functions, as the resulting matrix now closes. □

Note that we do not actually remove rows and columns. We do this to preserve the structure of the original modulated threshold matrix. It is easier then to determine which primary functions to append.

It can become a bit tedious to, every time we add a primary function, mention the primary function by name when all we need to know is which row we want to remove from the matrix. We therefor introduce an alternative notation, the so called dot-notation, that enables us to quickly add new primary functions.

When we want to remove a row we place a dot in front of the row to its left. The dots that are meant for transforming the same column are placed under each other, i.e. in the same column. If we want to remove another column we add dots in a new column. See figure 8.1.

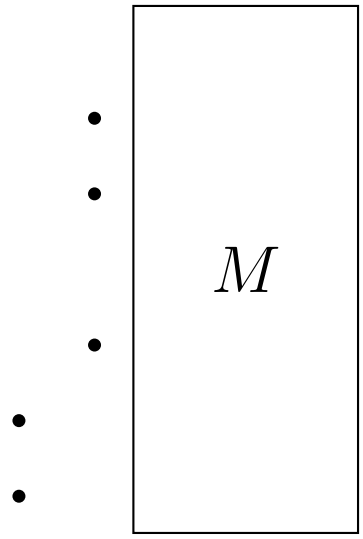

Figure 8.1: Adding primary functions to the threshold matrix is denoted by placing dots in front of the modulated threshold matrix. The dot marks the position of the one-entry for the primary function.

Now, lets see how this turns out for our function in the example 13 given earlier.

Example **14** The added primary functions in example 13 are shown below by the

dots in rows 6 and 7.

	0	0	0	1
	0	0	−1	−1
	0	−1	0	−1
	0	−1	−1	−1
	−1	0	0	−1
•	1	0	1	1
•	1	1	0	1
	−1	−1	−1	−1

□

The following rules can be used to determine which primary functions to add in order to be able to remove a column with the reduce rule.

1. For all columns count the number of ones and zeros. The minimum of these forms the indicator i, the maximum value p. The smallest value for i/p for these columns is the next candidate for removal. If there are more of these columns than we choose one with the following criteria in mind.

2. If the number of zeros and ones are equal in a column then we choose that one that eliminates as much of minority ones in other columns as possible. Minority ones are the entries that are 1 or -1 and when counted in the column are 1 or -1 and of opposite sign.

As was mentioned earlier this mask method probably adds more primary functions than strictly necessary. For instance take the following function

	x_2	x_1	T'	B_4
1	0	0	-1	0
2	0	-1	-1	0
3	-1	0	-1	0
4	1	1	1	1

Which is of course the AND-function and can immediately be implemented by a TLU. But the mask method would add a 1 to row 4, and then come to a stop on the last column which would become negative.

8.6 Minimizing Primary Functions

In this section we will focus on the third point of the conditions we stated in the first section for a succesful method for threshold logic. The aim is to reduce the number of functions in the cover we have found. We will first show that it is possible to replace a set of primary functions with a single function. We will then show that we

can just use the methods we are acustomed to and we will discuss a new element in minimizing which is new when compared to conventional logic design. But first we have to introduce some new concepts introduced by the following definitions.

DEFINITION **8.5** Let $f : \mathbb{B}^q \to \mathbb{B}$ be a Boolean function and let the set $P = \{b_1, b_2, \ldots, b_n\}$ be a set of primary functions that each have q inputs . Let the tuple $t = (t_1, t_2, \ldots, t_q)$ be such that $b_1(t_1, t_2, \ldots, t_q) = 1$, then the primary functions in P are said to be monosigned if for all possible tuples r we have

$$\text{if } b_i(r) = 1 \text{ then } f(r) = f(t) \text{ for all } b_i \in P$$

□

This definition expresses the idea that after modulation the primary functions can be classified into functions that have a -1 entry and functions that have a 1 entry. If a set of primary functions is monosigned it means they all have the same sign, i.e. they are all poscols or all negcols.

Sometimes we need a single function that has the same entries as a set of primary functions. The following definition makes this exact.

DEFINITION **8.6** Let the set $P = \{b_1, b_2, \ldots, b_n\}$ be a set of primary functions that each have q inputs and let $f : \mathbb{B}^q \to \mathbb{B}$ be a Boolean function that is defined by

$$\begin{aligned} f(t_1, t_2, \ldots, t_q) \;=\; & b_1(t_1, t_2, \ldots, t_q) \\ & \wedge\, b_2(t_1, t_2, \ldots, t_q) \\ & \vdots \\ & \wedge\, b_n(t_1, t_2, \ldots, t_q) \\ & \text{for all } t_1, t_2, \ldots, t_q \end{aligned}$$

The resulting function f is called the compound function. □

With these definitions we can now state the following theorem.

THEOREM **8.8** Let $P = \{b_1, b_2, \ldots, b_q\}$ be a set of primary functions that have each n inputs and that are monosigned with respect to the Boolean function $f : \mathbb{B}^q \to \mathbb{B}$. Let $g : \mathbb{B}^n \to \mathbb{B}$ be the compound function for P. Let T_1 be the modulated threshold matrix for f to which the functions in P where appended (before modulation), and let T_2 be the modulated threshold matrix for f to which the function g is appended (before modulation). Then T_1 is not nullable if and only if T_2 is not nullable.

PROOF

We assume that the primary functions of P and the compound function g are the first columns of T_1 and T_2 respectively.

Suppose T_1 is not nullable. If $q = 2^n$ then g is a positive or negative column in T_2 and so T_2 is not nullable. So, lets assume that $q < 2^n$. Then perform q times the reduce rule to T_1 and call the resulting matrix M. Because T_1 is not nullable the same holds for M. Now, perform the reduce rule to T_2. This leads to the same matrix M. Because M is not nullable, so is T_2.

Now suppose T_2 is not nullable. Again, if $q = 2^n$ then perform $q - 1$ times the reduce rule to T_1. The last primary function is now reduced to a positive or negative—single item—column, so T_1 is not nullable. Suppose that $q < 2^n$ then perform the reduce rule to T_2 and call the resulting matrix M. So, M is not nullable too. Now, perform q times the reduce rule to T_1 this leaves us with the same matrix M. So from this it follows that T_1 is not nullable too. With which we have obtained the required result. ■

This theorem states that it is indeed possible to replace some primary functions in a matrix that is not nullable, by a single Boolean function such that the matrix stays not nullable.

Contrary to what we are used to in conventional digital logic design we cannot combine all primary functions in a cover because covers are most of the time heterogeneous. The primary functions that are to be combined need to have the sam sign.

So, we have to divide a cover into two camps. One is all the primary functions in the cover that are all negcols in the modulated threshold matrix. We call thisset the negcamp. The other is comprises of all the primary functions in the cover that are poscols in the modulated threshold matrix. We call this set the poscamp. See figure 8.2. We need to minimize these two sets separately.

We still need to know how we should combine the primary functions. For this we need to realise that the primary functions are exactly the minterms we are familiar with from conventional logic design.

This gives us the opportunity to use the familiar methods for reducing the number of minterms. The two methods mostly used are the Karnaugh map method and the Quine-McCluskey algorithm.

When using TLUs to implement our Boolean function gives us an extra design dimension when compared to conventional digital logic design.

Observe the situation depicted by the Karnaugh map in figure 8.3. We can reduce the minterms to three terms. But when minterm $\overline{w}\overline{x}yz$ where one instead of zero as depicted in the Karnaugh map in figure 8.3 we could reduce the minterms to one term.

The following theorem gives us the possibility to add minterms so that we can create better conditions for minimization of the minterms.

THEOREM **8.9** Let T be the threshold matrix belonging to a function g such that T is nullable. Further let the Boolean function $f : \mathbb{B}^q \to \mathbb{B}$ be appended to the threshold matrix T such that the modulated threshold matrix for T is not nullable

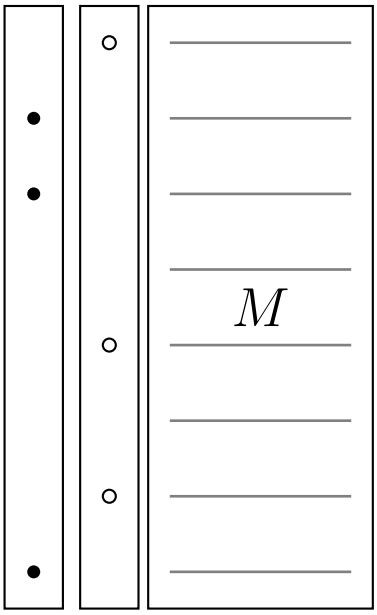

Figure 8.2: The situation where we are finished applying the mask method or finding a cover through the characteristic matrix, and we have collected all poscols in the poscamp and the negcols in the cover in the negcamp. The column on the left with the solid dots represents the set of poscols, i.e. the poscamp, and the column to its right, with the open dots is the negcamp. Together they form the cover. We minimize these two sets seperately.

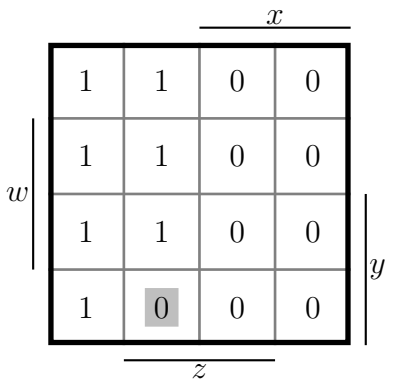

Figure 8.3: If the minterm $\overline{wx}yz$ were one instead of zero we could reduce the minterms to one term instead of three.

and call this matrix T_1. Let the Boolean function $f' : \mathbb{B}^q \to \mathbb{B}$ be the same as f except that it differs in row positions $r_1, r_2, \ldots, r_n$ and let it be appended to T. Let also the primary functions that have a one in row positions $r_1, r_2, \ldots, r_n$ be appended to this matrix. Call this matrix T_2. Then T_2 is also not nullable.

Proof

Perform the reduce rule to the primary functions in T_2, then the resulting matrix has rows that are a subset of the rows of T_1. Suppose there is a combination of rows in T_2 that are zero then this combination is also zero for T_1. But in that matrix no combination can make the rows become zero, because we have assumed that T_1 is not nullable. So, we have a contradiction and we have to conclude that there is no combination for T_2 such that that rows become zero. This means that T_2 is also not nullable. ■

So, this theorem shows we can indeed add extra primary function such that we can perform a better minimization, i.e. reduce the number of terms even further which would not be possible without the addition of th extra primary functions.

When the minterm has the same sign we can just add the minterm to the monosigned set we are minimizing, because we can apply the above theorem. Then we can split up the formed function into all primary functions, which is possible by theorem 8.8 stated in this section. Then we have twice the primary function we added as an extra primary function and we can leave out one. Then, finally, we can recombine all the primary functions into one function by theorem 8.8 again, which results in the desired situation.

If the added minterm is of opposite sign then we just add it to the camp and form the combined function from the existing minterms and the added minterm of opposite sign. We are then left with an added primary function and this new term, which together will be equivalent in the resulting new matrix.

So, why does this work? Because, by introducing a primary function of opposite sign and combining this into one function we have introduced the possibility of a mixed column in the product of the characteristic matrix and the Boolean functions obtained so far. A mixed column might mean it is possible that the matrix becomes nullable again. By adding the primary function as an extra column this is, however, prevented. We can immediately apply the reduce rule to this extra column and by doing this all rows are eliminated for which the entries of the mixed column might be of opposite sign. Lets examine this more closely in terms of the columns of the characteristic matrix in the following example given below.

$$
\begin{array}{cccc}
\overbrace{}^{❶} & & \overbrace{}^{❷} & \overbrace{}^{❸} \quad \overbrace{}^{❹} \\
0\ 0\ 0\ -1 & & -1 & -1 \\
1\ 0\ 0\ -3 & & -2 & -3 \\
0\ 0\ 0\ \ 0 & & 0 & 0 \quad 0 \\
0\ 2\ 0\ \ 0 & & 2 & 0 \quad 2 \\
0\ 0\ 0\ \ 0 & \longrightarrow \quad \sum = & 0 & 0 \quad \longrightarrow \quad 0 \\
4\ 1\ 0\ \ 0 & & 5 & 0 \quad 5 \\
5\ 0\ 2\ \ 0 & & 7 & 0 \quad 7 \\
6\ 0\ 1\ -5 & & 2 & -5 \\
0\ 0\ 0\ \ 0 & & 0 & 0 \quad 0
\end{array}
$$

The four columns on the left are the primary functions, including the extra appended primary of opposite sign which is shown here as the fourth column. By introducing a function that is the combination of these primary function we introduce a mixed column which is denoted here by the sum sign. The column to its right is the newly appended primary function which remains added to the matrix. By performing a reduce rule to this column we obtain the column on the right. So, to summarise we have

❶ The first three columns are the columns of the characteristic matrix corresponding to the minterms we want to combine that are already in the cover. The fourth column is the column in the characteristic matrix corresponding to the newly appended primary function.

❷ Here the column is the sum of the columns in 1. This column is corresponding to the newly obtained combined function. Observe that this column is indeed a mixed column.

❸ This is the fourth column in 1, which remains in the matrix.

❹ This column is the result of applying the reduce rule to column 3 in the matrix that contains the columns depicted by 2 and 3.

Remember that we had a cover. By introduction of the mixed column we no longer have a cover, because mixed columns cannot belong to a cover. But the combined function and the added primary function together still remove the rows that would have been removed by the original set of minterms. To see this let us first apply the reduce rule to column 3, then some rows that would have been covered by the set of minterms in the original setting are now covered by column 3. Those not covered by column 3 are still covered by column 2. Note that it is possible that column 3 covers rows that are not covered by the original set, but this is not a problem since it is a cover. By performing the reduce rule to column 3 first, we see that the mixed column becomes a simipositive column again to which we can apply the reduce rule. Together they remove at least the same set of rows as the replaced set of minterms would.