In Memoriam:
Stef Postma

South Africa has lost one of her most colourful and eminent computer scientists. Professor Stef Postma passed away peacefully in his sleep on May 5, 2000 after a short illness. He will be remembered for his forthright views and total integrity. Never a man to shy from controversy, he always debated his position with vigour, displaying his extensive vocabulary at every opportunity.

Those who knew him mourn the loss of a very good friend.

Stef was born on August 10, 1938 in Graaff-Reinet and matriculated from Hoërskool Linden in Johannesburg. He majored in geology and mathematics at the University of the Witwatersrand and graduated with honours in mathematics from that university. Stef devoted much of his life to promoting computer science as a science and to this end spent a lot of energy and time defining syllabi for undergraduate and honours courses at our universities. He was the prime mover in creating the South African Institute of Computer Scientists and Information Technologists (SAICSIT) in 1982, providing a professional body to represent the interests of local computer scientists. He was also instrumental in establishing Quaestiones Informaticae (now the South African Computer Journal) which afforded South African computer scientists the opportunity to publish papers locally in a refereed journal.

-Doug Laing
Editorial Page

Links2Go “Computer Science Journals” Award

The SACJ production editing team received the following unsolicited e-mail:


How your page was selected:

Each quarter, Links2Go samples millions of web pages to determine which pages are most heavily cited by web pages authors, such as yourself. The most popular pages are downloaded and automatically categorized by topic. At most 50 of the pages related to a topic are selected as “Key Resources.” Out of 50 pages selected as Key Resources for the Computer Science Journals topic, your page ranked 19th. For topics like Music, where there are a large number of interested authors and related pages, it is harder to achieve selection as a Key Resource than for a special-interest topic, such as Quantum Physics.

The Links2Go Key Resource award differs from other awards in two important ways. First, it is objective. Most awards rely on hand selection by one or more “experts,” many of whom have only looked at tens or hundreds of thousands of pages in bestowing their awards. Selection for these awards means no more than that one person, somewhere, noticed your page and liked it enough to select it. The Key Resource award, on the other hand, is based on an analysis of millions of web pages. Any group or organization who conducts a similar analysis will arrive at similar conclusions. When Links2Go says your page is a Key Resource, we mean that your page is one of the most relevant pages related to a particular topic on the web today, using an objective statistical measure applied to an extremely large data set.

Second, the Key Resource award is exclusive. We get literally hundreds of people requesting that their page be added to one or more topics per week. All of these requests are denied. The only way to get listed as a Key Resource is to achieve enough popularity for our analysis to select your pages automatically. We do not accept fees, offers of link exchanges, free advertising, or bartered livestock as inducements to add new sites to our lists. Fewer than one page in one thousand will ever be selected as a Key Resource.

Once again, congratulations on your award! Links2Go Awards awards@links2go.com
Syntactic Description of Neighbourhood in Quadtree

JR Tapamo

Department of Computer Science, Gaston Berger University, Saint-Louis, Senegal, tapamo@ugb.sn

Abstract

Region representation is very important in Graphics and Image Processing. This paper is concerned with a formalisation of image representation by quadtrees. Indeed an image here is a language over an alphabet of four letters. With rewriting rules we describe the neighbourhood of a node in the representation of image by quadtree.

Keywords: Quadtree, Rewriting rules, Image processing, neighbourhood, q-code


1 Introduction

This article is concerned with a formalisation of region representation by quadtrees which allows us to process pictures. Work has been done on quadtrees representation and manipulation: quadtrees construction, neighbour finding of node, etc. [1, 2, 3, 5, 10, 11, 12, 13]. We propose here a syntactic approach of quadtree manipulation which permits us to establish a link between quadtree representation and formal languages [4].

Image representation by quadtree [5] uses essentially a terminology based on binary images in which Black nodes are those which belong to the scene, White nodes are those which belong to the substratum, and Grey nodes are those belonging to both the substratum and the scene as illustrated in figure 1. Graphics representation by quadtrees needs a lot of memory space. Abel [1] proposed a representation by B-trees, in which a list structure of black node key values can be consulted sequentially by increasing order or by direct access. The advantage here is the fact that we can use quadtree for applications which need a large memory space. Gargantini [5] proposed a linear representation which allows recovery of 66% of the space generally used by quadtrees. It is also proved that algorithms can be executed in logarithmic time. We present here a linear representation of quadtrees that will be manipulated with q-code and quadtree. This quadtree contains addresses of records containing information on maximum and homogeneous blocks. Maximum and homogeneous blocks are those whose fathers are not homogeneous.

2 Coding

We present in figure 1 quadtree representation of a binary image. A linear representation of grey level image by the same structure is then developed from this. In the case of Grey level Images any node contains the address of the block which it represents if this block is homogeneous and

\[ \text{letter } m_i = \begin{cases} a & \rightarrow \text{shifting} \{ 0 \} \\ b & \rightarrow 2^{k-i} \\ c & \rightarrow 2^{k-i} \\ d & \rightarrow 2^{k-i} + 2^{k-i} \end{cases} \]

Figure 3: index shifting

an invalid address otherwise. In the case of figure 2, 16 is an invalid address. Considering that image is composed over a complete quadtree with depth \( k \) [14]. Let us use figure 2 to illustrate a representation of a tree whose leaves are maximum and homogeneous blocks and a structure containing information on leaves.

\( p \)-coding of image with \( 2^k \times 2^k \) pixels is an indexing of sites from 0 to \( 2^{2k} - 1 \), linking with linear storage in which \( p \)-code of site is a binary word with \( 2k \) bits. We decompose \( p \)-code over \( 2k \) bits in \( 2 \) words of \( k \) bits and denote the \( p \)-code of a pixel \( s \) by \( p(s) \).

\( q \)-coding of an image is the indexing relative to the quadtree representation. Q-codes under \( 2^k \times 2^k \) image admit 3 representations:

1) a literal representation corresponds to a word with \( k \) letters over an alphabet \( \Sigma = \{ a, b, c, d \} \), \( m = a_1a_2...a_k \) and \( m \in \Sigma^k \).

2) a numeric representation in base four with the following correspondences:

\( a \rightarrow 0, b \rightarrow 1, c \rightarrow 2, d \rightarrow 3 \) thus \( m = m_1m_2...m_k \) and \( m \in \{ 0,1,2,3 \}^k \).

3) a numeric representation in base two, with the following correspondences:

\( a \rightarrow a_0b \rightarrow 01, c \rightarrow 10, d \rightarrow 11 \) then \( m = m_1m_2...m_{2k} \) which corresponds with one-to-one application of a word with \( 2k \) bits and is an element of \( \{ 0,1 \}^{2k} \).

We consider the literal representation of q-code, with each letter \( \{ \text{from left to right} \} \) corresponding subdivision by four. Then let \( m = m_1m_2...m_k \) be a q-code, each letter \( m_1, m_2, ..., m_k \) corresponding to index shifting as shown in figure 3.
Example: if $k = 4$, we have an image with $16 \times 16$ bits, that is 256 bits

\[
q(s) = abcd \\
p(s) = (0) + (2^{8-2}) + (2^{4-3}) + (2^{4-4} + 2^{5-4}) \\
= 0 + 2^6 + 2^1 + 2^0 + 2^4 \\
= 83 \text{ (base ten)} \\
= 01010011 \text{ (base two)} \\
= 1103 \text{ (base four)}
\]

**Theorem 1.1**

In an image with $2^k \times 2^k$ sites, for all sites $s$ and $t$ such that

\[
p(s) = p_{2k-1} \ldots p_1 p_0 \text{ (p-code on base two)} \\
q(t) = q_{2k-1} \ldots q_1 q_0 \text{ (q-code on base two)}
\]

\[
s = t \iff \begin{cases} 
q_{2i} = p_i \\
q_{2i+1} = p_{i+k}
\end{cases} \quad (\forall i = 0, \ldots, k-1)
\]

**Proof**

If $m_i$ is the $i^{th}$ letter of the q-code we have the decomposition of table 1,

it means that $q_{2k+1-2j} = p_{2k-j}$. Putting $2k - j = i + k$, we have $j = k - i$ and $2k - 2j + 1 = 2k - 2(k-i) + 1 = 2i + 1$ where $q_{2i+1} = p_{i+k}$. $q_{2k-2j} = p_{k-j}$ putting $i = k - j$ we have $j = k - i$ and $2k - 2j = 2i$, hence $q_{2i} = p_i$. \qed

### Table 1: transcoding p-code/q-code

<table>
<thead>
<tr>
<th>$m_i$</th>
<th>$q_{2k+1-2j}$</th>
<th>$q_{2j}$</th>
<th>$p_{2k-j}$</th>
<th>$p_{k-j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 3 Neighbourhood Relations

Every site in an image has a specific position at which it is in touch with other sites, characterised sometimes by geographical relations; North, South, East, West; denoted by N, S, E, W respectively. These relations are applied to image processing in a square network. But if the processing is done in an octagonal network, it is necessary to define supplementary relations NE, NW, SE, SW which correspond respectively to 4-neighbour and 8-neighbour [8, 9, 10, 11].

Given a pixel $p$, we use figures 4.a and 4.b to illustrate respectively neighbourhood relations in square and octagonal networks.

### 3.1 Neighbourhood between two sites.

Let two sites $s$ and $t$ be in $S$ ($S$ is the set of sites of the image; if we represent an image by quadtree we can identify $S$ with $\Sigma^k$, where $\Sigma = \{a, b, c, d\}$ and
L, C and N represent respectively the row of the first pixel of the block, the column of the first pixel of the block, and the level of the block in the quadtree, and the q-code is computed using algorithm described in [5].

Figure 2: Linear representation of grey level image
Neighbourhood relations of the pixel p

\[ p \in R = \{N,S,E,W,NE,NW,SE,SW\} \]

We say that t is neighbour-p of s if t is at p of s and we denote it by \( t = p(s) \). We can then define \( \preceq^p \), such that

\[ \forall s, t \in S, \quad s \preceq^p t \iff t = p(s) \]

### 3.2 Addressing neighbourhood of site by p-code

Given \( s \), a site of an image, the following table gives the p-code of neighbour-\( p \) of \( s \), where \( p \) is an element of the set \( \{N,S,E,W,NE,NW,SE,SW\} \) function of p-code of \( s \). \( t = p(s) \) is the p-code of the site \( s \) and the computation of the neighbours of a site \( s \) is given in table 2:

**Example:** \( s \preceq^E t \iff p(r) = p(s) + 2^0 \)

### 4 Rewriting Rules

With the representation by pointers Samet [10] presents neighbour finding techniques on quadtrees which permit determination of adjacency in horizontal, vertical and diagonal directions, and also studies the complexities of algorithms used. Abel [1] proposes a manipulation of quadtrees by B-trees; Loew and Li [?] present a neighbours finding of nodes based on position and height of node deduced from q-code. We use here the production rules [14] applied on q-codes [5, 4] to determine neighbours, father, sons and others relations of a node in a quadtree.

#### 4.1 Relations Composition

The relations \( NE, NW, SE, SW \) are composed relations of relations \( N, S, E, W \). Thus let \( r_1 \) and \( r_2 \) be elements of \( \{N,S,E,W\} \) for all elements of \( S \), then

\[ r_1 r_2(s) = r_1(r_2(s)) \]

To simplify syntactic study, we instead use a postfixed representation, such that \( s r_2 r_1 \) is equivalent to \( r_1 r_2(s) \).

#### 4.2 Table of Rewriting Rules

Production Rules of neighbourhood relations on the q-codes of sites are given in table 3:

<table>
<thead>
<tr>
<th>Table 3: rewriting Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_E ) rules</td>
</tr>
<tr>
<td>( aE \rightarrow b )</td>
</tr>
<tr>
<td>( bE \rightarrow Ea )</td>
</tr>
<tr>
<td>( cE \rightarrow d )</td>
</tr>
<tr>
<td>( dE \rightarrow Ec )</td>
</tr>
</tbody>
</table>

Putting \( P = P_E \cup P_N \cup P_W \cup P_S \), we define rewriting rules that permit us to generate neighbours of a site.

**Example:** The neighbour - E of \( acbaa \) is given by the derivation \( acbaaE \rightarrow acbab \).

### 5 Determination of the Neighbourhood of a Node

#### 5.1 Block

Doing a recursive decomposition of a region over a quadtree, a block is a quadrant and a node of quadtree.

Previously, we identified a site with its q-code. Q-code of a block is the word described by the labels of the path from the root through the node corresponding to the block in the quadtree. We identify also the block with its q-code.

For a scene \( S \) with \( 2^k \times 2^k \) sites. We consider the language \( \Sigma^k = \Sigma^0 \cup \Sigma^1 \cup \ldots \Sigma^k = \{ x \in \Sigma^* | |x| \leq k \} \).

A block can be considered as an element of \( \Sigma^k \), a subset of \( S \) given by the application \( x \in \Sigma^k \rightarrow B(x) \in P(S) \) (set of subsets of \( S \)) with

\[ B(x) = \{ s \in S | x \sigma q(s) \} \]

where \( \tau \) is the transitive closure of the relation "is the prefix of"

\( B(x) \) will be the block with q-code \( x \):

- \( l(B) = |x| \) is the length of the word \( x \in \Sigma^k \).
- \( n(B) = k - l(B) \) is the level of block \( B \).
- \( d(B) = 2^n(B) \) is the dimension of block \( B \).
- \( r(B) = d^2(B) \) is the size of the block \( B \).
- \( \beta(B) \) is block basis with q-code \( xd^k(x) \).

The block basis is the first site of \( B \) in the lexicographic order given by q-code.

#### 5.2 Neighbours with equal size

**Theorem 4.1**

Let \( A(a_1) \) and \( B(b_1) \) be two blocks with equal size. Then \( A \) and \( B \) are adjacent if and only if there exists a relation \( p \in \{N,S,E,W,NE,NW,SE,SW\} \) such that \( a_1 = b_1p \), where \( l(A) = l(B) \).

**Proof**

We have already used the notation \( a_1 = b_1p \) to describe the neighbourhood of a relation \( p \) of the sites with q-codes \( a_1 \) and \( b_1 \) respectively. We will use it here to describe the neighbourhood of \( B(b_1) \) and \( A(a_1) \) by the relation \( p \).
Without loss of generality, we consider that we have an image with $2^1 \times 2^1 (A)$ sites. We consider also A(a1) and B(b1) pseudo-sites which should be neighbours, hence $a_1 = b_1 p$.

Considering the rewriting rules described in section 4 above, we can compute all the neighbours of a node with equal size. Example: Let $abc$ be the q-code of the block over an image with $2^7 \times 2^7$ pixels. Then $abcN = aba$ and $aha$ is the neighbour $-N$ of $abc$.

Remark: Let $p \in R$ and $x \in V(n)$ where $n$ is the depth of the complete quadtree associated with the scene, $x$ do not have neighbour $-p$ with the same size if and only if $xp = py$ where $y \in V(n)$.

5.3 Neighbours with different sizes

We determine the neighbourhood of a block with size different from the blocks considered. blocks.

Theorem 4.2

Let $B$ be a block with q-code $x \in \Sigma(n)$, if $n = n(B)$, $\beta = xa^n$ and the q-code of the basis $B$ then $B = \{t \in S/q(t) = xa^EiSj \mid 0 \leq i < 2^a, 0 \leq j < 2^b \}$, where $EiSj$ is the postfixed relation of $S(Ei(t))$ and $Ei$ is the composition $i$ times of the operator $E$ (East) and $Sj$ is the composition $j$ times of the operator $S$ (South).

Proof

The property in the theorem is trivial in the following case:
$n = 0$ because $B = \{\beta\} = \{x\}$

$n = 1$ because $B = \{xa, xb, xc, xd\}$

The proof of the theorem relies on the definition of q-coding. Let $q$ be the q-code of the basis $\beta = xy$ then

The operator $Ei$ corresponds to the addition (without overflow)

\[ pk + \ldots + p1p0 + i \]

and the operator $Sj$ corresponds to the addition (without overflow)

\[ p2k + \ldots + p1p0 + j \]

Consequently, when $i$ has maximum of $n$ digits which are $1$ (in the significant bits) in the binary representation, $\beta Ei$ has the form $x_1x_2\ldots x_ny_1y_2\ldots y_n$ which is possible only if: $i \leq 2^n-1$.

\[ \alpha \ldots \ldots \beta \]

\[ \ldots \ldots \ldots \]

\[ \ldots \ldots \ldots \]

\[ \gamma \ldots \ldots \delta \]

Figure 5: Representation of q-codes of the four corners of a block $2^n-1 + 2^{n-2} + \ldots + 2^1 + 2^0 = 2^n - 1$. We have the identical result for $bE^j$. We have then proved that:

\[ \{t \in S/q(t) = \beta Ei; 0 \leq i, j < 2^a\} \subset B \]

Reciprocally, the uniqueness of q-code (or p-code) assume that the application:

\[ (i, j) \in [2^n] \times [2^n] \rightarrow bEiSj \in B \]

is injective, then the inclusion below becomes an equality.

\[ \square \]

Corollary 4.2.1

The block $B$ with level $n$ is a 'square' of $2^n \times 2^n$ sites and therefore connected using neighbourhood relations $\approx p \in R$.

Corollary 4.2.2

The basis $\beta(B)$ of a block is the 'first' (=the smallest) in the following order:

- lexicographic order of p-code $xa \leftrightarrow p(s) \leq p(t)$
- lexicographic order of q-code with convenience that $a < b < c < d$
- numerical order of q-code.

Corollary 4.2.3

\[ \beta E^i 2^j = xa^{n-ib^j} \]

\[ \beta S^j 2^i = xa^{n-ic^i} \]

\[ \beta(SE)^i 2^j = xa^{n-id^j} \]

Remark: Let $B(x)$ be a block and $\alpha, \beta, \gamma, \delta$ represented as shown in figure 5. Then $\alpha = xb^n; \beta = xa^n; \gamma = xe^n; \delta = xd^n$, where $n = l(B)$.

6 Adjacency Relation

6.1 Adjacency of two regions

Two regions $r_1$ and $r_2$ are said to be adjacent if there exists two sites $s_1 \in r_1, s_2 \in r_2$ and $p \in \{N, S, E, W, NE, NW, SE, SW\}$ such that $s_1 \approx p s_2$.

Remark: Let $A = A(a_1)$ and $B = B(b_1)$ such that $n(A) = n(B)$. Then $A$ and $B$ are adjacent if and only if $A \approx B$ for $p \in R$. No 25, 2000
6.2 Adjacency of two blocks with different size.

Suppose \( A = A(a_1) \) and \( B = B(b_1) \) with \( n(A) \neq n(B) \). We say that \( A \) is adjacent to \( B \) in the following cases:

(i) if \( n(A) < n(B) \), \( \exists x \in \Sigma^{n(B)-n(A)} \) such that \( a_1x = b_1p \).
(ii) if \( n(B) < n(A) \), \( \exists x \in \Sigma^{n(A)-n(B)} \) such that \( b_1x = a_1p \).

for \( p \in R \).

7 Applications

7.1 Determination of Neighbourhood of a Block.

Consider the set of q-codes representing quadtree associated with a picture over a grid \( 2^k \times 2^k \), for which we propose an algorithm describing the neighbourhood of node. The father of a block \( B(x) \) is the block \( B(y) \) such that \( y \) is the prefix of \( x \) with size \( |x| - 1 \), denoted by \( \text{father}(x) \).

Every block has eventually four sons and those sons are of types \( NW, NE, SW, SE \). To determine a son, it is necessary to know which type of son we want to find. The function which determines the son of \( B(x) \) is a function of two variables which is denoted \( \text{son}(x, T) \) where \( T \) is the type of the son. \( \text{son}(x, T) \) is then the concatenation of \( x \) and the code of \( T \).

Example:
If \( x = 1023 \) \( \text{father}(x) = 102 \text{ son}(x, NW) = 12030 \)

7.1.1 Algorithm for determination of Neighbourhood.

Let \( LFQ \) be the language of leaves of the quadtree associated with a scene. For a block \( B = B(x) \) do:

**Step1**

Generate the neighbours with equal size if they exist.

Let \( N = B(n), S = B(s), E = B(x), W = B(w), NW = B(r), NE = B(q), SE = B(z), SW = B(t) \) with \( P \) (rewriting rules) as described previously, we can generate the language \( L(x) = \{ (\cup_{p \in \mathbb{R}}) \cap LFQ \} \) where \( p \in R \), which is a set of the neighbours with equal size.

**Step2**

Use the neighbours computed in step 1 for the determination of neighbours with smaller size than the considered block. \( L_p(x) \) is the language of the neighbours of \( B(x) \) with size smaller than \( l(B) \) and at \( p \) of block \( B(x) \) where \( p \in R \). Let us put \( m = n - n(B) \) where \( n \) is the depth of the tree. Hence,

if \( n \in LFQ \) then \( L_N(x) = \{ y \in \{ c, d \}^{(m)}, y \in LFQ \} \) else \( L_N(x) = \emptyset \)

if \( s \in LFQ \) then \( L_S(x) = \{ y \in \{ a, b \}^{(m)}, y \in LFQ \} \) else \( L_S(x) = \emptyset \)

if \( e \in LFQ \) then \( L_E(x) = \{ y \in \{ a, c \}^{(m)}, y \in LFQ \} \) else \( L_E(x) = \emptyset \)

if \( w \in LFQ \) then \( L_W(x) = \{ y \in \{ b, d \}^{(m)}, y \in LFQ \} \) else \( L_W(x) = \emptyset \)

if \( q \in LFQ \) then \( L_NE(x) = \{ y \in \{ b \}^{(m)}, y \in LFQ \} \) else \( L_NE(x) = \emptyset \)

if \( r \in LFQ \) then \( L_NW(x) = \{ y \in \{ a \}^{(m)}, y \in LFQ \} \) else \( L_NW(x) = \emptyset \)

if \( z \in LFQ \) then \( L_SW(x) = \{ y \in \{ d \}^{(m)}, y \in LFQ \} \) else \( L_SW(x) = \emptyset \)

if \( t \in LFQ \) then \( L_SE(x) = \{ y \in \{ c \}^{(m)}, y \in LFQ \} \) else \( L_SE(x) = \emptyset \)

Step3

Use the result of step 1 to determine the neighbours with higher size than \( l(B) \). Let \( l = l(x) \) and let us put:

\( n = n_1 \ldots n_t, s = s_1 \ldots s_t, e = e_1 \ldots e_t, w = w_1 \ldots w_t, q = q_1 \ldots q_t, z = z_1 \ldots z_t, r = r_1 \ldots r_t \).

Using the following algorithm, we compute \( L_p(x) \), the neighbours of \( x \) at \( p \):

Begin

\[ L_p(x) \leftarrow \emptyset \]

\[ Ox \leftarrow \text{father}(xp) \]

\[ Oy \leftarrow \text{father}(x) \]

While \( (Ox \neq Oy) \) do

Begin

If \( Oy \in LFQ \) then \( L_p(x) \leftarrow Oy \)

Else begin

\[ Ox \leftarrow \text{father}(Ox) \]

\[ Oy \leftarrow \text{father}(Oy) \]

end

EndWhile

End

The set of neighbours of a node \( x \) is then

\[ L(x) \cup \left( \cup_{p \in \mathbb{R}} (L_p(x) \cup L_q(x)) \right) \].

7.1.2 Example of the determination of neighbours

We determine the set of neighbours of \( x = 30 \) in figure 6, which is an image with size \( 2^3 \times 2^3 \), using the previous algorithm.

\[ LFQ = \{ 0, 10, 11, 13, 20, 22, 30, 31, 32, 33, 120, 121, 122, 123, 210, 211, 212, 213 \} \].

**Step1**

\[ 30N = 3N2 = 12 \notin LFQ \]

\[ 30NE = 12E = 13 \in LFQ \]

\[ 30S = 32 \in LFQ \]

\[ 30SW = 32W = 3W3 = 23 \notin LFQ \]

\[ 30E = 31 \in LFQ \]

\[ 30SE = 32E = 33 \notin LFQ \]

\[ 30W = 3W1 = 21 \notin LFQ \]

\[ 30NW = 12W = 1W3 = 03 \notin LFQ \]
Figure 6: Representation of node x

12, 21, 23 and 03 are not elements of LFQ then it is possible that there exist neighbours of x which are smaller or greater.

Step2

\[
\begin{align*}
  & x = 12 \{ \\
  & \quad 12\{2,3\}(1) = \{122, 123\} \\
  & \quad L_N(x) = \{122, 123\} \\
  & x = 21 \{ \\
  & \quad 21\{1,3\}(1) = \{211, 213\} \\
  & \quad L_N(x) = \{211, 213\} \\
  & x = 03 \{ \}
  & \quad 03\{0\}(1) = \{030\} \\
  & \quad L_{NW}(x) = \emptyset \text{ because } 030 \notin LFQ
\end{align*}
\]

Step3

Only 03 has neither son nor neighbour with equal size. Then there exists then v, ancestor of 03, which is a neighbour of x. Then applying the algorithm of step 3 we have \( v = 0 \). The set of neighbours of x is then \( \{32, 31, 12, 123, 123, 211, 213, 0\} \)

The algorithm described above can be used as a part of Image segmentation process.

7.2 Pattern Generation

The rewriting rules P can be a generator of patterns. Let x be an element of \( \Sigma^* \). Choosing \( \rho \) to be a finite composition of elements of R we can generate pattern from x by assuming that \( x_0 \) computes the \( \rho \)-neighbour and saves it in G (the set containing blocks which compose the pattern we are to generate). Initially G contains x. We show the following two examples:

a) Square Generator.
Let x be the lower left block of the square with side length equal to 1. Then \( G = xN^0E^1S^1W^1 \).

b) V Generator.
Let x be the upper left block of V and l the length of its branch. Then \( G = x(SE)^l(NE)^l \).

8 Conclusion

We have presented a syntactic tool for determining the neighbours of a node in quadtree with a region of picture described as a language. The present work is limited to the description of the neighbourhood of a node. We are now working on properties of the Grammar \( G = (V_T, V_N, P) \) where \( V_T = \{ a, b, c, d \} \) is the set of terminal symbols, \( V_N = \{ N, S, E, W \} \) is the one of the non-terminal symbols and P the rewriting rules defined above.

Acknowledgements

Thanks to Alain Leduc, Richard Moore, Cecilia Adanma Eberendu and the anonymous referees who made several helpful comments for improving the readability of this paper.

References


