RLE-based Algorithm for Testing Biorders

Oliver Lanzerath

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

28. September 2015

Content

Biorder

Definition

Let $R \subseteq X \times X$ be a homogeneous binary relation. R is called **biorder**, iff

$$aRb \wedge cRd \wedge \neg aRd \rightarrow cRb$$

holds $\forall\ a, b, c, d \in X$.

R	a	b	c	d
a	0	1	0	0
b	1	1	0	0
c	1	1	0	1
d	0	0	0	0

Biorder

Definition

Let $R \subseteq X \times X$ be a homogeneous binary relation. R is called **biorder**, iff

$$aRb \wedge cRd \wedge \neg aRd \rightarrow cRb$$

holds $\forall\ a, b, c, d \in X$.

R	a	b	c	d
a	0	$\boxed{1}$	0	$\boxed{0}$
b	1	1	0	0
c	1	1	0	$\boxed{1}$
d	0	0	0	0

Biorder

Definition

Let $R \subseteq X \times X$ be a homogeneous binary relation. R is called **biorder**, iff

$$aRb \wedge cRd \wedge \neg aRd \rightarrow cRb$$

holds $\forall\ a, b, c, d \in X$.

R	a	b	c	d
a	0	1	0	0
b	1	1	0	0
c	1	1	0	1
d	0	0	0	0

Echelon Block Form

Definition

Let $M \subseteq A \times B, A = \{a_1, ..., a_n\}, B = \{b_1, ..., b_m\}$ be a binary matrix. The linear arrangement of the elements corresponds with their indices. The matrix M is in **echelon block form**, iff $\exists k_i, 0 \leq k_i \leq m$ for each row $\vec{a_i}$ with

- $(\{a_i\}, \{b_1, ..., b_{k_i}\})$ built a 1-block
- $(\{a_i\}, \{b_{k_i+1}, ..., b_m\})$ built a 0-block

and $k_i \geq k_{i+1}$.

Echelon Block Form - Example

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_1	1	1	1	1	1	1	0	0
a_2	1	1	1	1	0	0	0	0
a_3	1	1	1	1	0	0	0	0
a_4	1	0	0	0	0	0	0	0
a_5	0	0	0	0	0	0	0	0

Biorder $\Leftrightarrow$ Echelon Block Form

Lemma

A binary relation is called biorder, iff the corresponding binary matrix can be represented in echelon block form by rearranging rows and columns independently.

R	ESP	CHL	NLD	AUS	$\lvert\vec{x}\rvert$
ESP	0	0	0	1	1
CHL	1	0	0	1	2
NLD	1	1	0	1	3
AUS	0	0	0	0	0
$\lvert\vec{y}\rvert$	2	1	0	3	

Biorder $\Leftrightarrow$ Echelon Block Form

Lemma

A binary relation is called biorder, iff the corresponding binary matrix can be represented in echelon block form by rearranging rows and columns independently.

R	ESP	CHL	NLD	AUS	$\lvert\vec{x}\rvert$
NLD	1	1	0	1	3
CHL	1	0	0	1	2
ESP	0	0	0	1	1
AUS	0	0	0	0	0
$\lvert\vec{y}\rvert$	2	1	0	3	

Biorder $\Leftrightarrow$ Echelon Block Form

Lemma

A binary relation is called biorder, iff the corresponding binary matrix can be represented in echelon block form by rearranging rows and columns independently.

R	AUS	ESP	CHL	NLD	$\|\vec{x}\|$
NLD	1	1	1	0	3
CHL	1	1	0	0	2
ESP	1	0	0	0	1
AUS	0	0	0	0	0
$\|\vec{y}\|$	3	2	1	0	

Biorder ⇔ Echelon Block Form

Lemma

A binary relation is called biorder, iff the corresponding binary matrix can be represented in echelon block form by rearranging rows and columns independently.

R	AUS	ESP	CHL	NLD	$\lvert\vec{x}\rvert$
NLD	1	1	1	0	3
CHL	1	1	0	0	2
ESP	1	0	0	0	1
AUS	0	0	0	0	0
$\lvert\vec{y}\rvert$	3	2	1	0	

Run Length Encoding

Definition

Let $seq_i \in \{\mathbf{0}^j | j \in \mathbb{N}\} \cup \{\mathbf{1}^j | j \in \mathbb{N}\}$, $i \in \mathbb{N}$, be a sequence with

$$value(seq_i) = \begin{cases} 0 & seq_i \in \{\mathbf{0}^j | j \in \mathbb{N}\} \\ 1 & seq_i \in \{\mathbf{1}^j | j \in \mathbb{N}\} \end{cases}.$$

Then, a bitvector $\vec{x} = x_0 \ldots x_{n-1} \in \{0,1\}^n$ can be represented as $\vec{x} = seq_1 \ldots seq_k$, $1 \leq k \leq n$, $value(seq_i) \neq value(seq_{i+1})$, $\sum_{i=1}^{k} |seq_i| = n$. The RLE-coding of a vector $\vec{x}$ is given by the vector

$$\vec{x}^{rle} = x_0 \, [|seq_1|, \ldots, |seq_k|]$$

RLE-Coding - Example

$$\underbrace{\underbrace{{\color{red}1}1111}_{5}\ \underbrace{000}_{3}\ \underbrace{11111111111}_{11}}_{{\color{red}1}[5,3,11]}$$

M	a	b	c	d
a	1	0	0	1
b	1	0	1	0
c	1	1	1	1
d	0	0	0	1

M	
$\vec{a}^{rle}$	1[1,2,1]
$\vec{b}^{rle}$	1[1,1,1,1]
$\vec{c}^{rle}$	1[4]
$\vec{d}^{rle}$	0[3,1]

Test by Rearranging Rows and Columns

If a given relation is a biorder, the echelon block form can be achieved in two steps:

1. Sort the rows by their Hamming weight in descending order.
2. Sort the columns by their Hamming weight in descending order.

With respect to biorder tests the second step is not needed because...

Test by Rearranging Rows and Columns

...after sorting the rows only three types of column vectors can occur if the relation is a biorder.

Test by Rearranging Rows and Columns

...after sorting the rows only three types of column vectors can occur if the relation is a biorder.

$$\begin{array}{r|l} 0 & 0 \\ & 0 \\ & 0 \\ & 0 \\ & 0 \\ & \vdots \\ & 0 \\ & 0 \\ & 0 \\ & 0 \\ n & 0 \end{array} \qquad \begin{array}{r|l} 0 & 1 \\ & 1 \\ & 1 \\ & 1 \\ & 1 \\ & \vdots \\ & 1 \\ & 1 \\ & 1 \\ & 1 \\ n & 1 \end{array} \qquad \begin{array}{r|l} 0 & 1 \\ & 1 \\ & \vdots \\ & 1 \\ & 1 \\ \mathrm{j} & 1 \\ & 0 \\ & 0 \\ & \vdots \\ & 0 \\ n & 0 \end{array}$$

Test by Rearranging Rows and Columns

...after sorting the rows only three types of column vectors can occur if the relation is a biorder.

$$\begin{array}{c|c} 0 & 0 \\ & 0 \\ & 0 \\ & 0 \\ & 0 \\ & \vdots \\ & 0 \\ & 0 \\ & 0 \\ & 0 \\ n & 0 \end{array} \quad \text{0[n]} \qquad \begin{array}{c|c} 0 & 1 \\ & 1 \\ & 1 \\ & 1 \\ & 1 \\ & \vdots \\ & 1 \\ & 1 \\ & 1 \\ & 1 \\ n & 1 \end{array} \quad \text{1[n]} \qquad \begin{array}{c|c} 0 & 1 \\ & 1 \\ & \vdots \\ & 1 \\ & 1 \\ \text{j} & 1 \\ & 0 \\ & 0 \\ & \vdots \\ & 0 \\ n & 0 \end{array} \quad \text{1[j,n-j]}$$

Analysis

- Row vectors **and** column vectors must be RLE-coded.
- Sorting: $\mathcal{O}(n \log n)$
- Checking against the three types
  - $\rightarrow$ binary: $\mathcal{O}(n^2)$
  - $\rightarrow$ RLE-coded: $\mathcal{O}(n)$

R	a	b	c	d	$\vec{x}^{rle}$
a	0	0	0	0	0 [4]
b	1	0	1	1	1 [1, 1, 2]
c	0	1	0	1	0 [1, 1, 1, 1]
d	1	1	1	1	1 [4]
$\vec{y}^{rle}$	0 [1, 1, 1, 1]	0 [2, 2]	0 [1, 1, 1, 1]	0 [1, 3]	

Analysis

- Row vectors **and** column vectors must be RLE-coded.
- Sorting: $\mathcal{O}(n \log n)$
- Checking against the three types
  - $\rightarrow$ binary: $\mathcal{O}(n^2)$
  - $\rightarrow$ RLE-coded: $\mathcal{O}(n)$
- **BUT:** each change operation for rows can have effects on all RLE-coded column vectors.

R	a	b	c	d	$\vec{x}^{rle}$
d	1	1	1	1	1 [4]
b	1	0	1	1	1 [1, 1, 2]
c	0	1	0	1	0 [1, 1, 1, 1]
a	0	0	0	0	0 [4]
$\vec{y}^{rle}$	1 [2, 2]	1 [1, 1, 1, 1]	1 [2, 2]	1 [3, 1]	

$\rightarrow$ RLE-coded: $\mathcal{O}(n^3)$

Test by Rearranging only Rows

R	a	b	c	d	$\lvert\vec{x}\rvert$	$\vec{x}^{rle}$
a	0	0	0	0	0	$0\,[4]$
b	1	0	1	1	3	$1\,[1,1,2]$
c	0	1	0	1	2	$0\,[1,1,1,1]$
d	1	1	1	1	4	$1\,[4]$

Test by Rearranging only Rows

- Sorting the rows.

R	a	b	c	d	$\lvert\vec{x}\rvert$	$\vec{x}^{rle}$
d	1	1	1	1	4	$1\,[4]$
b	1	0	1	1	3	$1\,[1,1,2]$
c	0	1	0	1	2	$0\,[1,1,1,1]$
a	0	0	0	0	0	$0\,[4]$

Test by Rearranging only Rows

- Sorting the rows.
- $\vec{d} \vee \vec{b} = \vec{d}$?

R	a	b	c	d	$\lvert\vec{x}\rvert$	$\vec{x}^{rle}$
d	1	1	1	1	4	1 [4]
b	1	0	1	1	3	1 [1, 1, 2]
c	0	1	0	1	2	0 [1, 1, 1, 1]
a	0	0	0	0	0	0 [4]

Test by Rearranging only Rows

- Sorting the rows.
- $\vec{d} \vee \vec{b} = \vec{d}$?
  $\rightarrow \surd$
- $\vec{b} \vee \vec{c} = \vec{b}$?

R	a	b	c	d	$\lvert\vec{x}\rvert$	$\vec{x}^{rle}$
d	1	1	1	1	4	$1\,[4]$
b	1	0	1	1	3	$1\,[1,1,2]$
c	0	1	0	1	2	$0\,[1,1,1,1]$
a	0	0	0	0	0	$0\,[4]$

Test by Rearranging only Rows

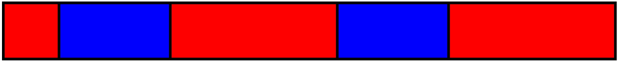
- Sorting the rows.
- $\vec{d} \vee \vec{b} = \vec{d}$?
  $\rightarrow \surd$
- $\vec{b} \vee \vec{c} = \vec{b}$?
  $\rightarrow \lightning$

$\rightarrow$ The relation is no biorder.

R	a	b	c	d	$\lvert\vec{x}\rvert$	$\vec{x}^{rle}$
d	1	1	1	1	4	$1\,[4]$
b	1	0	1	1	3	$1\,[1,1,2]$
c	0	1	0	1	2	$0\,[1,1,1,1]$
a	0	0	0	0	0	$0\,[4]$

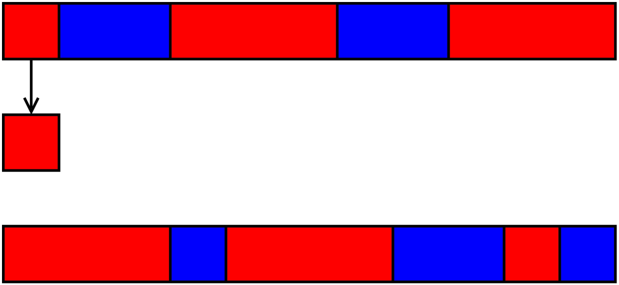
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



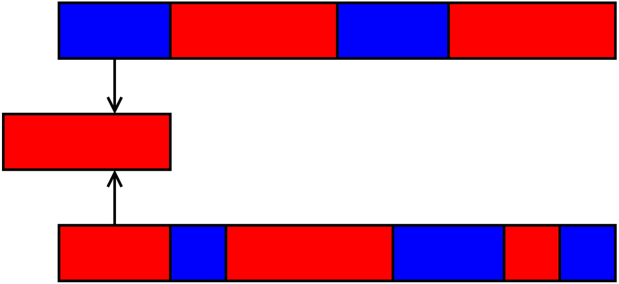
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



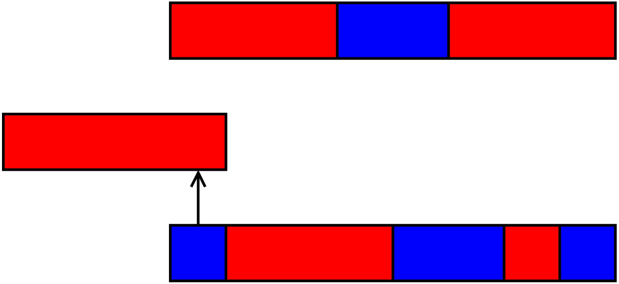
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



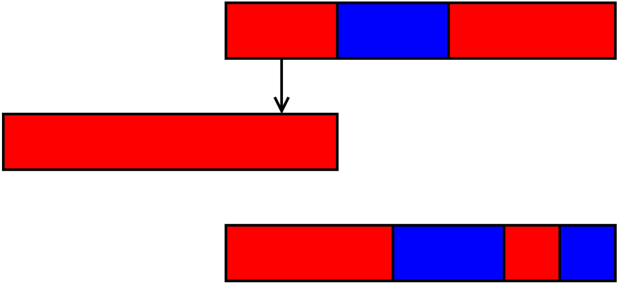
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



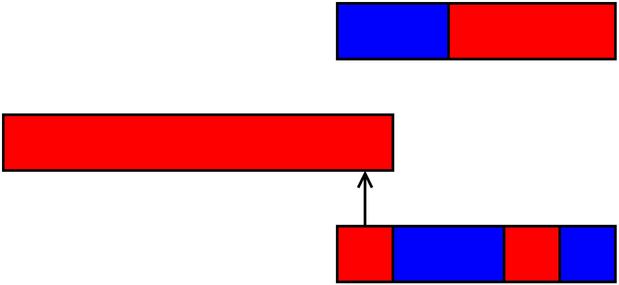
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



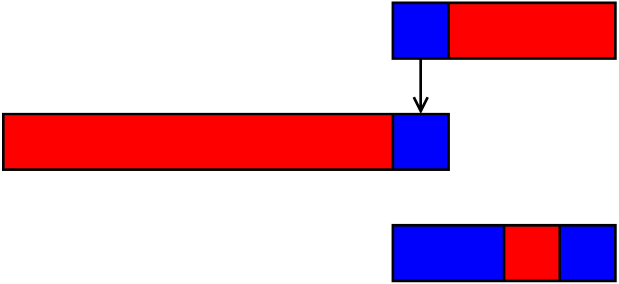
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



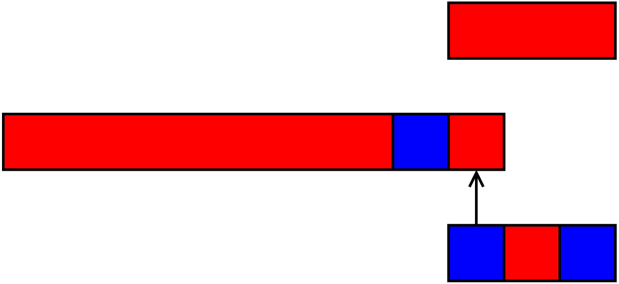
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



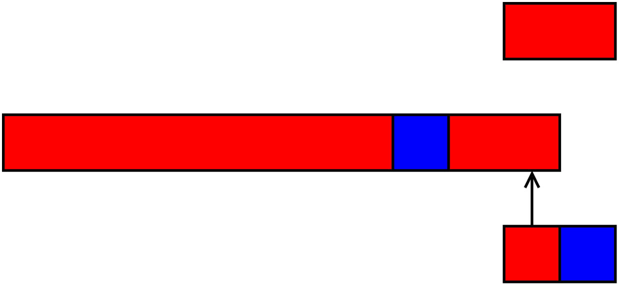
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



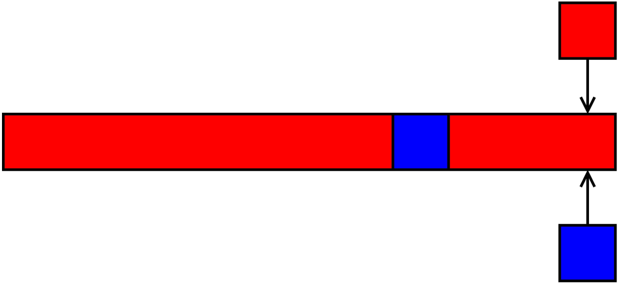
Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!

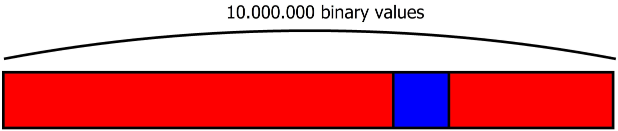


Analysis

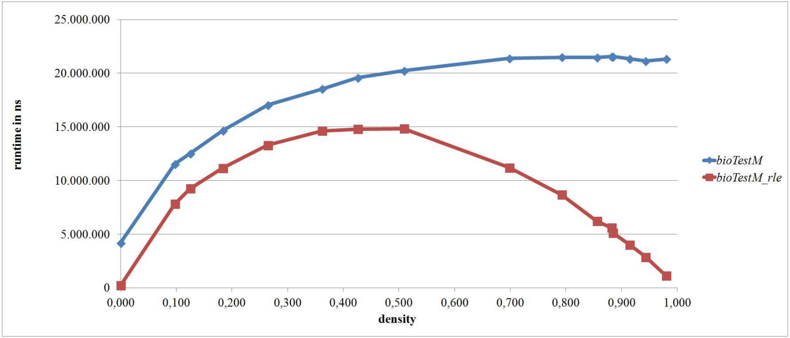
- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!

Analysis

- Only a line-by-line RLE-coding of the matrix is required.
- Sorting: $\mathcal{O}(n \log n)$
- A logical OR-operation for RLE-coded vectors is needed.
  $\rightarrow$ Runtime depends on the size of the array!



Time Measurement



References

R. M. Haralick, *The Diclique Representation and Decomposition of Binary Relations*, Journal of the ACM, vol. 21, pp. 356-366, 1974.

D. Salomon, *Data compression - The Complete Reference*, 4th Edition, London, UK: Springer, 2007.

G. Schmidt, *Relational Mathematics, Encyclopedia of Mathematics and its Applications*, vol. 132, Cambridge, UK: Cambridge University Press, 2011.

Thank you for your attention!