Fishburn trees

Giulio Cerbai
Joint work with Anders Claesson

University of Iceland
giulio@hi.is

The Fishburn numbers

$$\sum_{n\geq 0}\prod_{k=1}^{n}\left(1-(1-x)^k\right) = 1 + x + 2x^2 + 5x^3 + 15x^4 + 53x^5 + 217x^6 + \dots$$

The Fishburn numbers

$$\sum_{n\geq 0}\prod_{k=1}^{n}\left(1-(1-x)^k\right) = 1 + x + 2x^2 + 5x^3 + 15x^4 + 53x^5 + 217x^6 + \dots$$

Milestone papers

(I) *(2+2)-free posets, ascent sequences and pattern avoiding permutations*,
M. Bousquet-Melou, A. Claesson, M. Dukes, S. Kitaev, 2010.

-Ascent sequences (and modified ascent sequences);

-Fishburn permutations;

-Unlabeled (**2**+**2**)-free posets;

-Stoimenow matchings.

The Fishburn numbers

$$\sum_{n\geq 0}\prod_{k=1}^{n}\left(1-(1-x)^k\right) = 1 + x + 2x^2 + 5x^3 + 15x^4 + 53x^5 + 217x^6 + \dots$$

Milestone papers

(II) *Ascent sequences and upper triangular matrices containing non-negative integers*,
M. Dukes, R. Parviainen, 2010.

-Fishburn matrices and ascent sequences.

The Fishburn numbers

$$\sum_{n\geq 0}\prod_{k=1}^{n}\left(1-(1-x)^k\right) = 1 + x + 2x^2 + 5x^3 + 15x^4 + 53x^5 + 217x^6 + \dots$$

Milestone papers

(iii) *Composition matrices, (2+2)-free posets and their specializations*, M. Dukes, V. Jelínek, M. Kubitzke, 2011.

-Fishburn matrices and (**2**+**2**)-free posets.

The Fishburn numbers

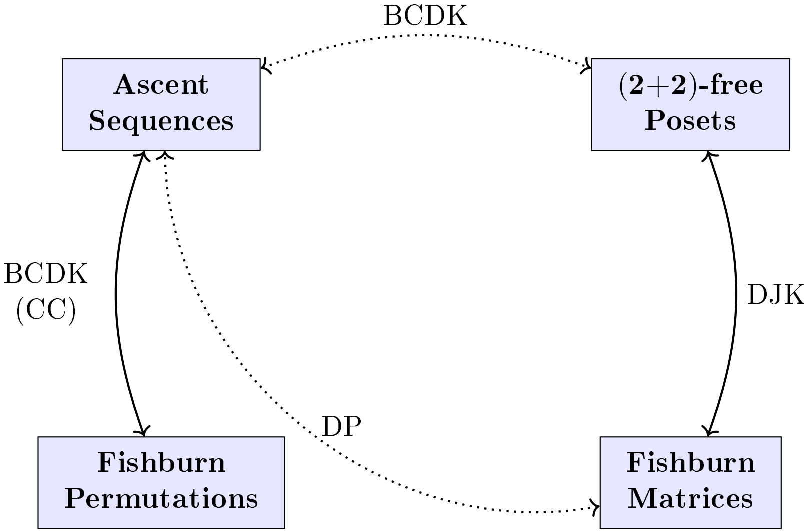
$$\sum_{n\geq 0}\prod_{k=1}^{n}\left(1-(1-x)^k\right) = 1 + x + 2x^2 + 5x^3 + 15x^4 + 53x^5 + 217x^6 + \dots$$

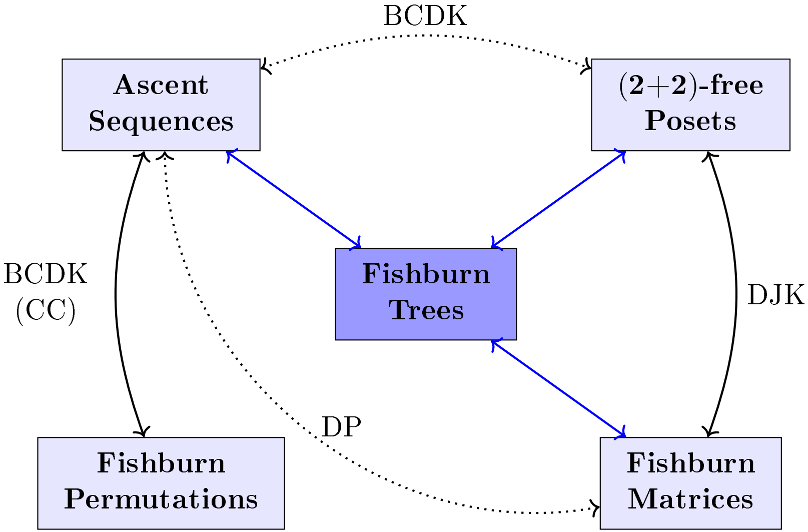
One more reference

(IV) *Transport of patterns by Burge transpose*,
C., Claesson, PP2022.

-Modified ascent sequences and Fishburn permutations;

-Transport of patterns based on the Burge transpose.


BCDK
Ascent Sequences
(2+2)-free Posets
BCDK
(CC)
DJK
DP
Fishburn Permutations
Fishburn Matrices


BCDK
Ascent Sequences
(2+2)-free Posets
BCDK (CC)
Fishburn Trees
DJK
DP
Fishburn Permutations
Fishburn Matrices

Modified Ascent Sequences

Combinatorial definition (C., Claesson, 2020)

A sequence $x = x_1 \cdots x_n$ is a **modified (ascent) sequence** if:

1. $x_1 = 1$;
2. $\{x_1, \ldots, x_n\}$ is an interval;
3. $x_i < x_{i+1}$ if and only if x_{i+1} is the leftmost copy of x_{i+1} in x.

Modified Ascent Sequences

Combinatorial definition (C., Claesson, 2020)

A sequence $x = x_1 \cdots x_n$ is a **modified (ascent) sequence** if:

1. $x_1 = 1$;
2. $\{x_1, \ldots, x_n\}$ is an interval;
3. $x_i < x_{i+1}$ if and only if x_{i+1} is the leftmost copy of x_{i+1} in x.

$$\{x_1\} \cup \{\text{Ascent tops}\} = \{\text{Leftmost copies}\}$$

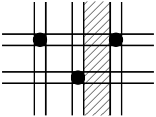
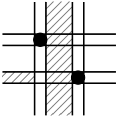
Modified Ascent Sequences

Combinatorial definition (C., Claesson, 2020)

A sequence $x = x_1 \cdots x_n$ is a **modified (ascent) sequence** if:

1. $x_1 = 1$;
2. $\{x_1, \ldots, x_n\}$ is an interval;
3. $x_i < x_{i+1}$ if and only if x_{i+1} is the leftmost copy of x_{i+1} in x.

Cayley permutations avoiding the Cayley-mesh patterns

 and 

Modified Ascent Sequences

Combinatorial definition (C., Claesson, 2020)

A sequence $x = x_1 \cdots x_n$ is a **modified (ascent) sequence** if:

1. $x_1 = 1$;
2. $\{x_1, \ldots, x_n\}$ is an interval;
3. $x_i < x_{i+1}$ if and only if x_{i+1} is the leftmost copy of x_{i+1} in x.

Example

$$\mathrm{x} = \underline{1}\ 1\ \underline{4}\ 1\ 1\ 1\ \underline{2}\ 2\ \underline{5}\ 2\ 2\ \underline{3}$$

Modified Ascent Sequences

Combinatorial definition (C., Claesson, 2020)

A sequence $x = x_1 \cdots x_n$ is a **modified (ascent) sequence** if:

1. $x_1 = 1$;
2. $\{x_1, \ldots, x_n\}$ is an interval;
3. $x_i < x_{i+1}$ if and only if x_{i+1} is the leftmost copy of x_{i+1} in x.

Non-example

$$\mathrm{x} = 1\ 2\ 1\ \boxed{2}$$

Modified Ascent Sequences

Combinatorial definition (C., Claesson, 2020)

A sequence $x = x_1 \cdots x_n$ is a **modified (ascent) sequence** if:

1. $x_1 = 1$;
2. $\{x_1, \ldots, x_n\}$ is an interval;
3. $x_i < x_{i+1}$ if and only if x_{i+1} is the leftmost copy of x_{i+1} in x.

Another non-example: why?

$$\mathrm{x} = 1\ 2\ 1\ 4\ 3$$

FROM MODIFIED SEQUENCES TO FISHBURN TREES

- Let x_m be the leftmost copy of $\max(x)$ in x;
- The **MAX-DECOMPOSITION** of $x = x_1 \dots x_n$ is

$$\boxed{x \;=\; \mathrm{pref}(x)\, x_m\, \mathrm{suff}(x)}$$

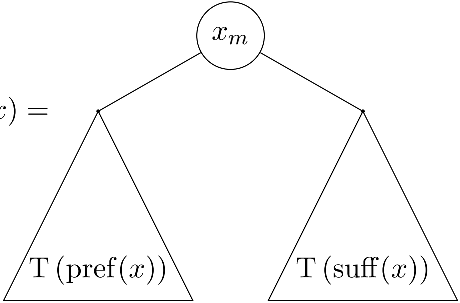
From modified sequences to Fishburn trees

- Let x_m be the leftmost copy of $\max(x)$ in x;
- The **max-decomposition** of $x = x_1 \ldots x_n$ is

$$\boxed{x \;=\; \mathrm{pref}(x)\, x_m\, \mathrm{suff}(x)}$$

The **rooted**, **labeled**, **binary** tree $\mathrm{T}(x)$ is defined by $\mathrm{T}(\emptyset) = \emptyset$ and:

$\mathrm{T}(x) =$



$$x = \quad 1 \quad 1 \quad 4 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 5 \quad 2 \quad 2 \quad 3$$

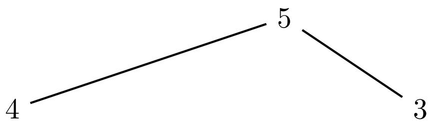
$$\mathrm{T}(x) =$$

$x =$ 1 1 4 1 1 1 2 2 5 2 2 3

5

$\mathrm{T}(x) =$

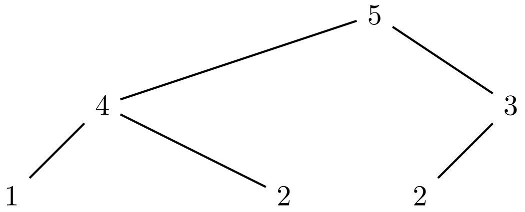
$x =$ 1 1 4 1 1 1 2 2 5 2 2 3

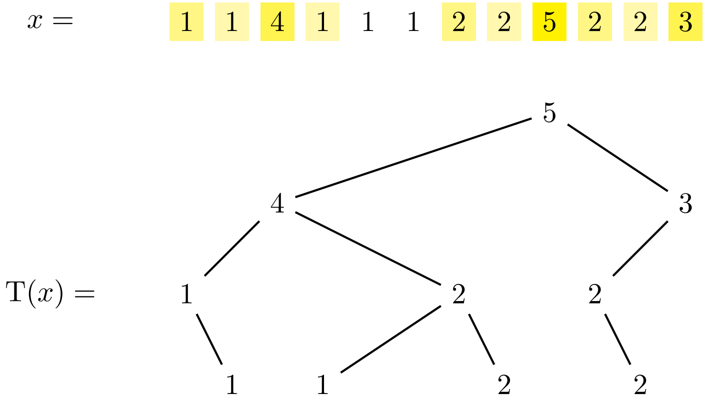


$\mathrm{T}(x) =$

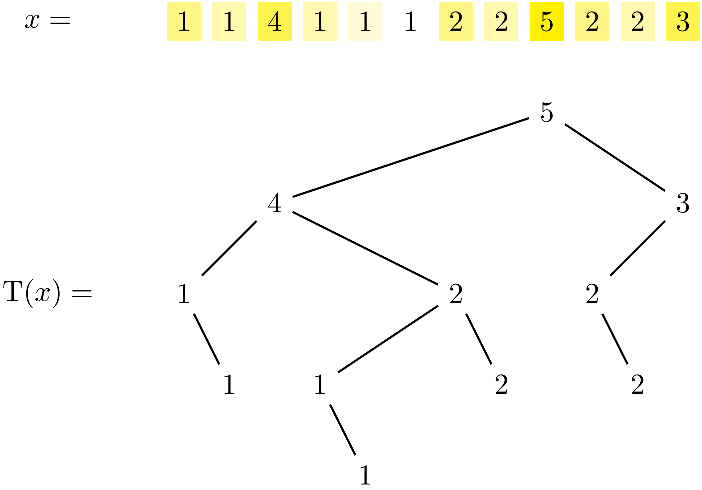
$x = 1\ 1\ 4\ 1\ 1\ 1\ 2\ 2\ 5\ 2\ 2\ 3$

$\mathrm{T}(x) =$

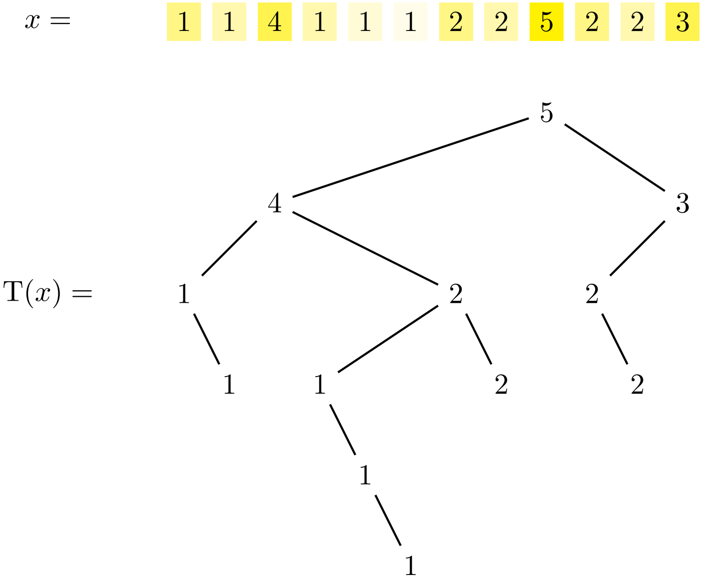




x = 1 1 4 1 1 1 2 2 5 2 2 3
T(x) =
5
4 3
1 2 2
1 1 2 2



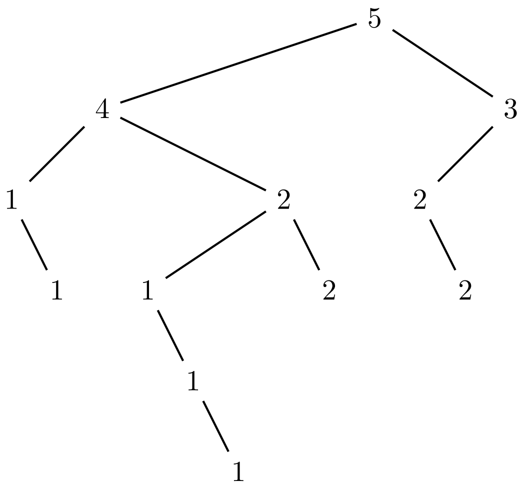
x =
1 1 4 1 1 1 2 2 5 2 2 3
T(x) =
5
4 3
1 2 2
1 1 2 2
1



x =
1 1 4 1 1 1 2 2 5 2 2 3
T(x) =
5
4 3
1 2 2
1 1 2 2
1
1

$x =$ 1 1 4 1 1 1 2 2 5 2 2 3

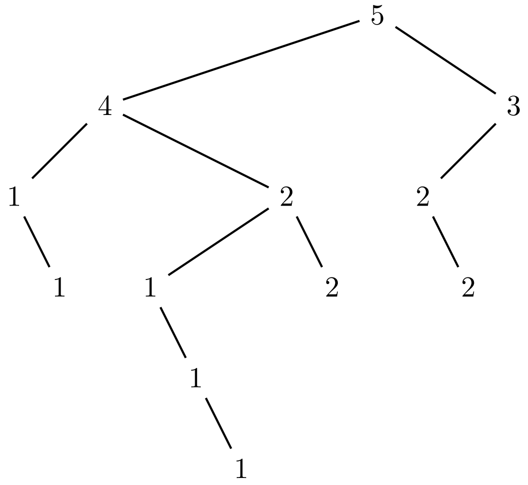
$\mathrm{T}(x) =$



Can you guess how to determine x from the tree $\mathrm{T}(x)$?

$x = \; 1 \; 1 \; 4 \; 1 \; 1 \; 1 \; 2 \; 2 \; 5 \; 2 \; 2 \; 3$

$\mathrm{T}(x) =$



Can you guess how to determine x from the tree $\mathrm{T}(x)$?

We use the in-order traversal!

Let $T = (L, r, R)$ be the rooted, labeled, binary tree with:

- Left subtree L;
- Root r with label $\mathfrak{l}(r)$;
- Right subtree R

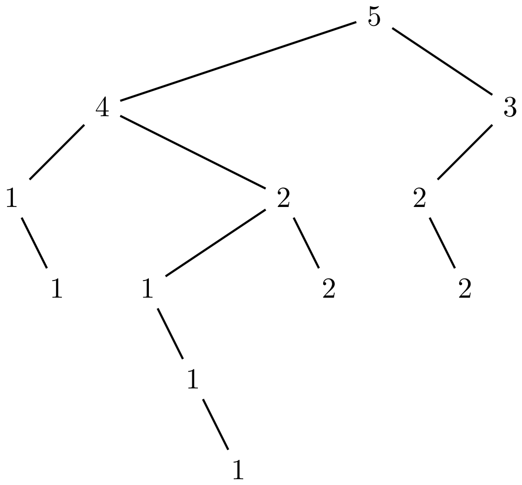
In-order sequence

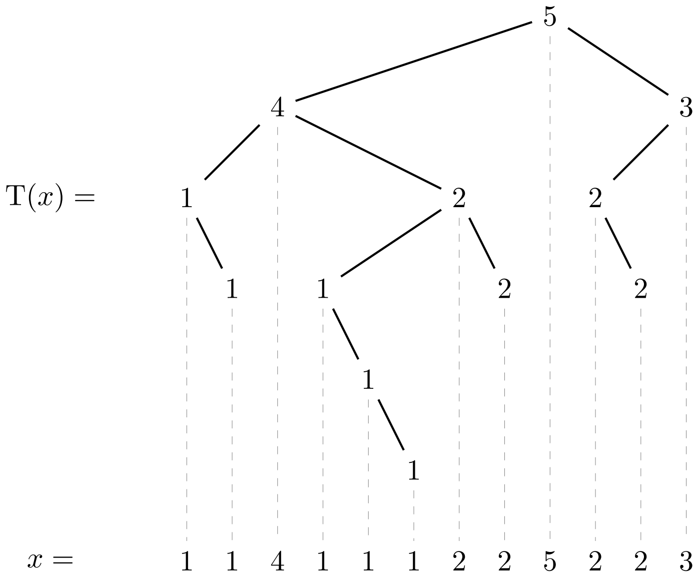
The in-order sequence of T is defined recursively by:

$$x(\emptyset) = \emptyset \qquad \text{and} \qquad x(T) \;=\; x(L)\;\mathfrak{l}(r)\;x(R).$$

"In-order traverse T and write down the label of each visited node".

$\mathrm{T}(x) =$




T(x) =
x = 1 1 4 1 1 1 2 2 5 2 2 3
5
4 3
1 2 2
1 1 2 2
1
1

Properties of $\mathrm{T}(x)$

1. **ROOTED.**

Properties of $T(x)$

1. **ROOTED.**
2. **BINARY**: Each node has either:
   - 0 children;
   - 1 child, which is either the left or the right child;
   - 2 children, namely a left child and a right child.

Properties of $\mathrm{T}(x)$

1. **ROOTED**.
2. **BINARY**: Each node has either:
   - 0 children;
   - 1 child, which is either the left or the right child;
   - 2 children, namely a left child and a right child.
3. **LABELED**:
   - Each node v has a positive integer label $\mathfrak{l}(v)$.
   - The set of labels is an interval.

Properties of T(x)

1. **ROOTED.**
2. **BINARY**: Each node has either:
   - 0 children;
   - 1 child, which is either the left or the right child;
   - 2 children, namely a left child and a right child.
3. **LABELED**:
   - Each node v has a positive integer label $\mathfrak{l}(v)$.
   - The set of labels is an interval.
4. **DECREASING TO THE RIGHT.**
5. **STRICTLY DECREASING TO THE LEFT.**

Properties of $T(x)$

1. **ROOTED.**

2. **BINARY**: Each node has either:
   - 0 children;
   - 1 child, which is either the left or the right child;
   - 2 children, namely a left child and a right child.

3. **LABELED**:
   - Each node v has a positive integer label $\mathfrak{l}(v)$.
   - The set of labels is an interval.

4. **DECREASING TO THE RIGHT.**

5. **STRICTLY DECREASING TO THE LEFT.**

6. $\{\text{Leftmost node}\} \cup \{\text{Nodes with a left child}\} = \{\text{Leftmost copies}\}$.

Properties of T(x)

1. **ROOTED.**
2. **BINARY**: Each node has either:
   - 0 children;
   - 1 child, which is either the left or the right child;
   - 2 children, namely a left child and a right child.
3. **LABELED**:
   - Each node v has a positive integer label $\mathfrak{l}(v)$.
   - The set of labels is an interval.
4. **DECREASING TO THE RIGHT.**
5. **STRICTLY DECREASING TO THE LEFT.**
6. $\{$Leftmost node$\} \cup \{$Nodes with a left child$\} = \{$Leftmost copies$\}$.

A **Fishburn tree** is a tree that satisfies all the above properties.

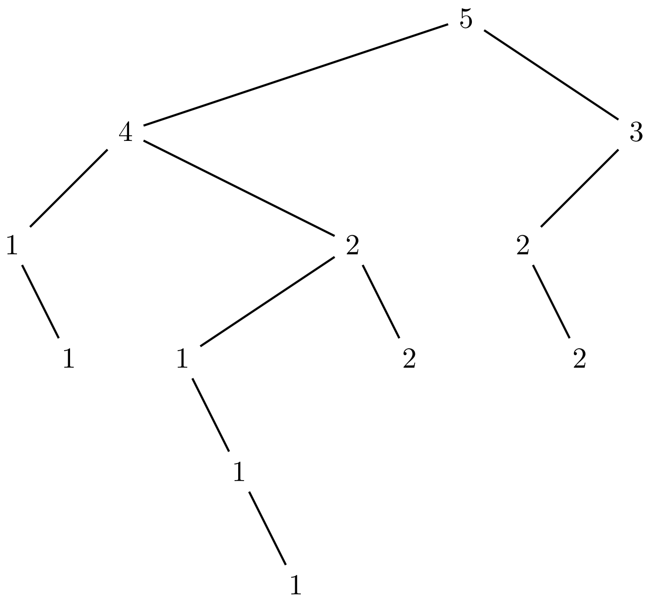
Maximal-right-path decomposition

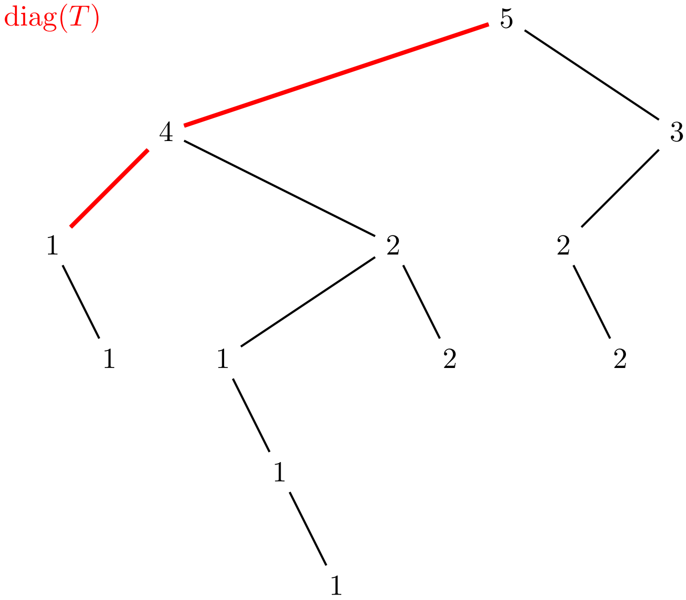
The **diagonal** of T is the path from the root to the leftmost node.

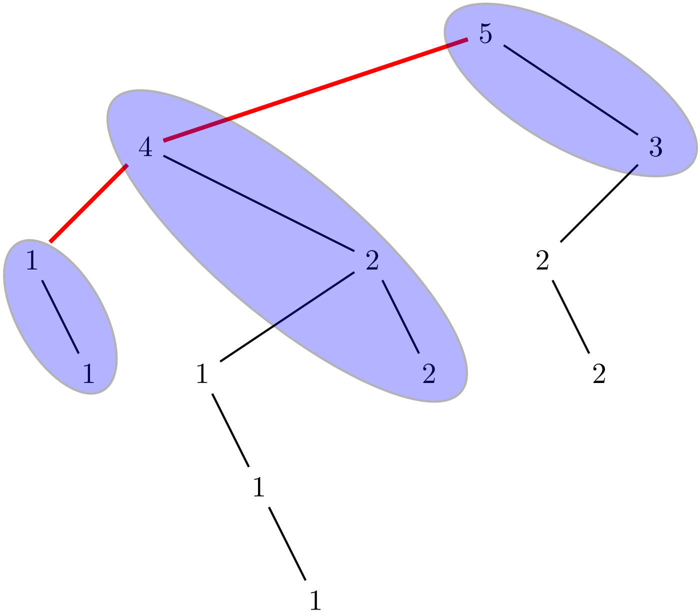
Maximal-right-path decomposition

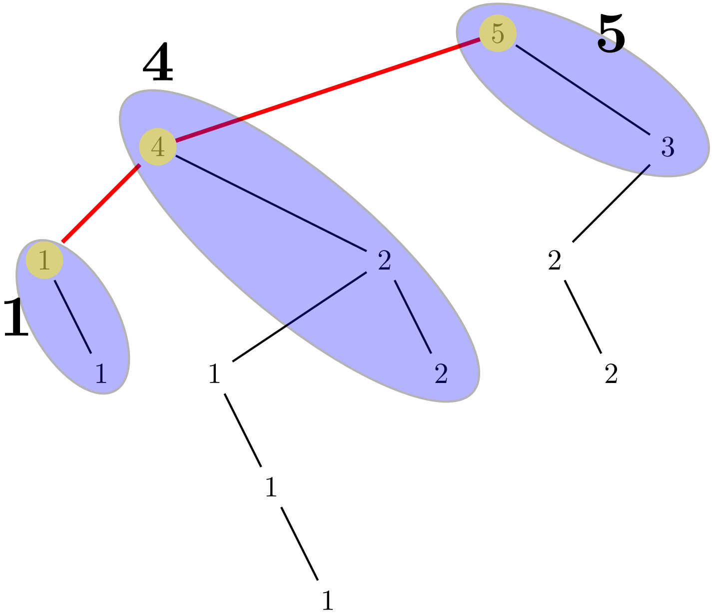
The **diagonal** of T is the path from the root to the leftmost node.

- Decompose T in maximal right paths.
- In each path, the topmost node v is either:
  (i) on the diagonal;
  (ii) or, the left child of a node that is not on the diagonal.
- Number each maximal right path by the label of:
  (i) its top node, if the top node is on the diagonal;
  (ii) the father of its top node, otherwise.

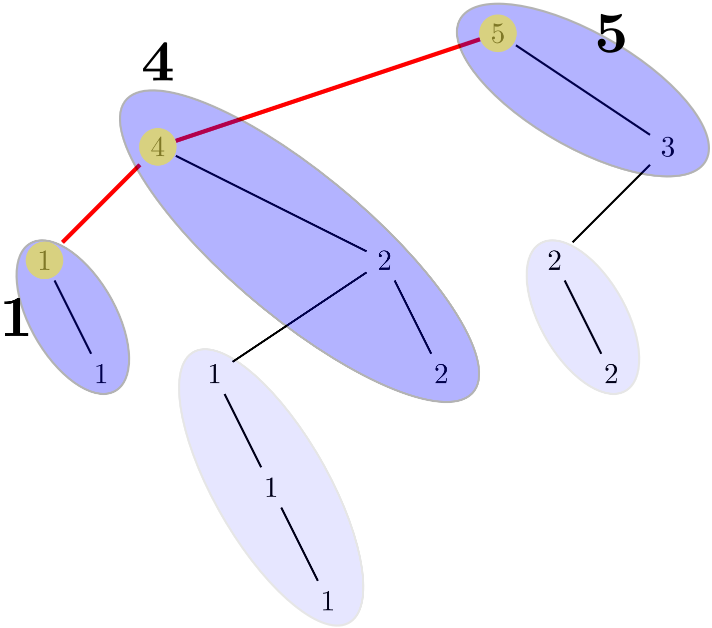

5
4
3
1
2
2
1
1
2
2
1
1

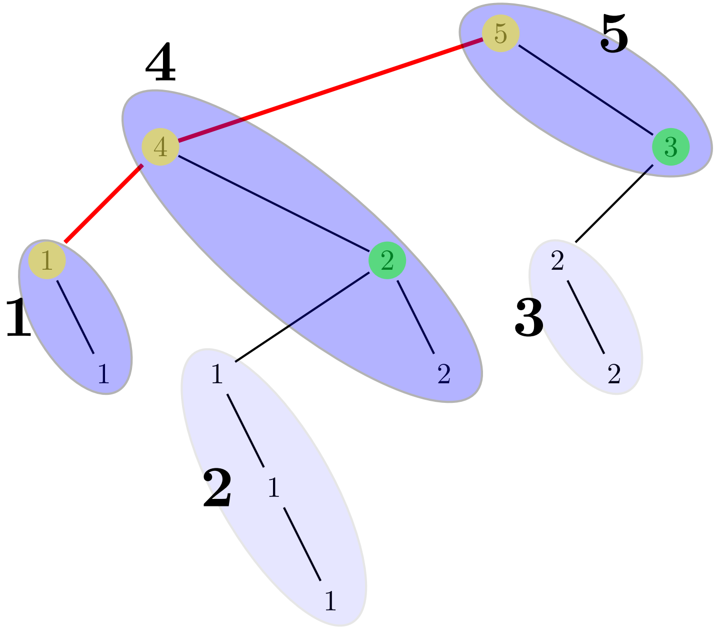

diag(T)
5
4
3
1
2
2
1
1
2
2
1
1


5
4
3
1
2
2
1
1
2
2
1
1



5
4
5
4
3
1
2
2
1
1
2
1
2
1
1


5
4
1
5
3
4
2
1
1
2
1
2
2
1
1



4
5
5
4
3
1
2
1
3
1
1
2
2
2
1
2
1
1

Maximal-right-path decomposition

Theorem

Let k be the maximal value of a label in T.

- There are k maximal-right-paths.
- Each one corresponds to a unique number in $\{1, 2, \ldots, k\}$.

Maximal-right-path decomposition

Theorem

Let k be the maximal value of a label in T.

- There are k maximal-right-paths.
- Each one corresponds to a unique number in $\{1, 2, \ldots, k\}$.

Each node v of T gets two labels:

The node label $\mathfrak{l}(v)$; The path label $\mathfrak{b}(v)$.

Maximal-right-path decomposition

Theorem

Let k be the maximal value of a label in T.

- There are k maximal-right-paths.
- Each one corresponds to a unique number in $\{1, 2, \ldots, k\}$.

Each node v of T gets two labels:

The node label $\mathfrak{l}(v)$; The path label $\mathfrak{b}(v)$.

Theorem

$$\mathfrak{l}(v) \leq \mathfrak{b}(u).$$

Proof. A Fishburn tree is decreasing.

Fishburn matrices

- Lower triangular;
- Non-negative integer entries;
- Every row and column contains at least one non-zero entry.

- The size of a Fishburn matrix is the sum of its entries.

Fishburn matrices

- Lower triangular;
- Non-negative integer entries;
- Every row and column contains at least one non-zero entry.

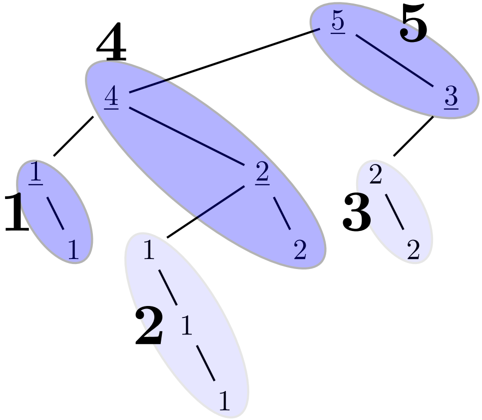
- The size of a Fishburn matrix is the sum of its entries.

Fishburn matrices of size 3

$$[3]\,,\begin{bmatrix}2 & \\ \cdot & 1\end{bmatrix},\begin{bmatrix}1 & \\ \cdot & 2\end{bmatrix},\begin{bmatrix}1 & \\ 1 & 1\end{bmatrix},\begin{bmatrix}1 & & \\ \cdot & 1 & \\ \cdot & \cdot & 1\end{bmatrix}$$

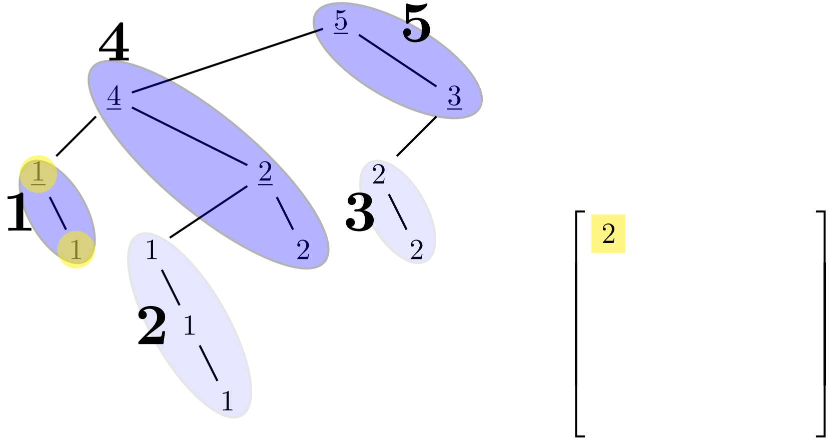
The Fishburn matrix $A = (a_{ij})$ associated with T is:

$$\begin{aligned} a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\ &= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|. \end{aligned}$$



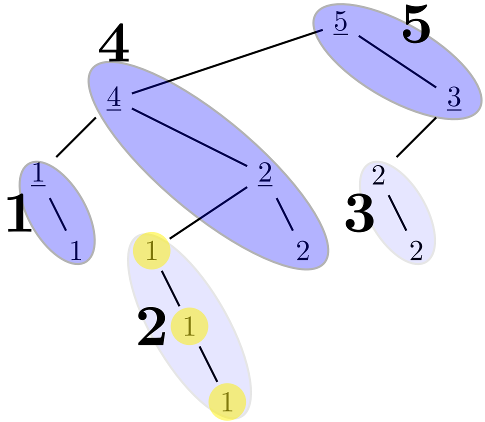
The Fishburn matrix $A = (a_{ij})$ associated with T is:

$$\begin{aligned} a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\ &= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|. \end{aligned}$$



The Fishburn matrix $A = (a_{ij})$ associated with T is:

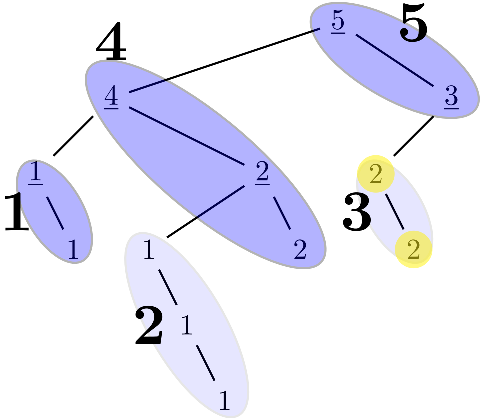
$$\begin{aligned} a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\ &= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|. \end{aligned}$$



$$\begin{bmatrix} 2 & & & & \\ 3 & & & & \\ & & & & \\ & & & & \end{bmatrix}$$

The Fishburn matrix $A = (a_{ij})$ associated with T is:

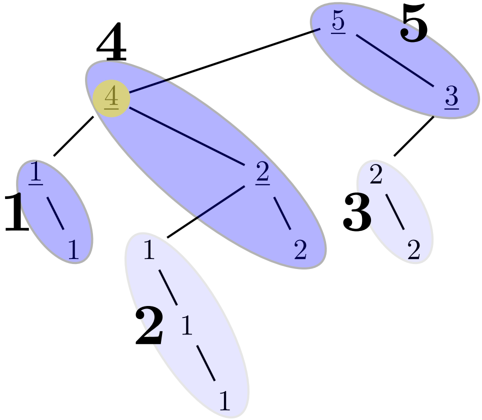
$$\begin{aligned} a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\ &= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|. \end{aligned}$$



$$\begin{bmatrix} 2 & & & & \\ 3 & & & & \\ & 2 & & & \\ & & & & \\ & & & & \end{bmatrix}$$

The Fishburn matrix $A = (a_{ij})$ associated with T is:

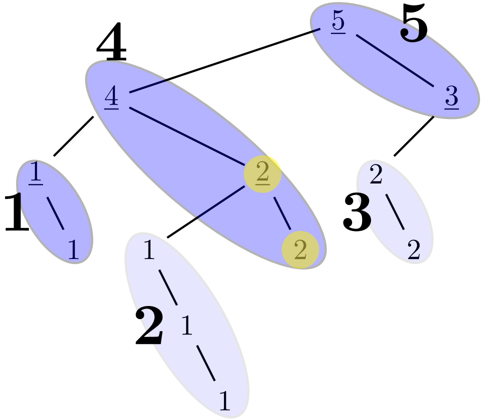
$$\begin{aligned} a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\ &= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|. \end{aligned}$$



$$\begin{bmatrix} 2 & & & & \\ 3 & & & & \\ & 2 & & & \\ & & & 1 & \\ & & & & \end{bmatrix}$$

The Fishburn matrix $A = (a_{ij})$ associated with T is:

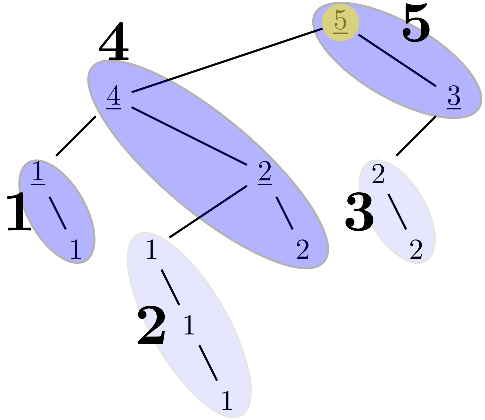
$$
\begin{aligned}
a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\
&= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|.
\end{aligned}
$$



$$
\begin{bmatrix}
2 & & & & \\
3 & & & & \\
 & 2 & & & \\
 & 2 & & 1 & \\
 & & & &
\end{bmatrix}
$$

The Fishburn matrix $A = (a_{ij})$ associated with T is:

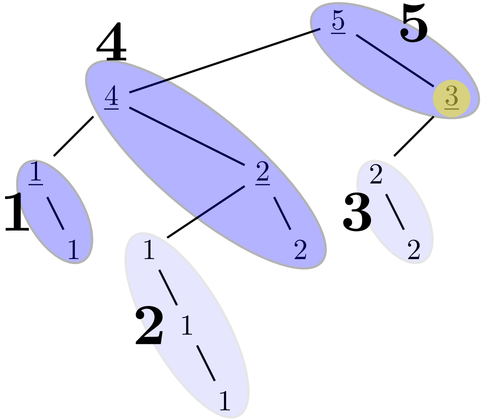
$$\begin{aligned} a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\ &= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|. \end{aligned}$$



$$\begin{bmatrix} 2 & & & & \\ 3 & & & & \\ & 2 & & & \\ & 2 & & 1 & \\ & & & & 1 \end{bmatrix}$$

The Fishburn matrix $A = (a_{ij})$ associated with T is:

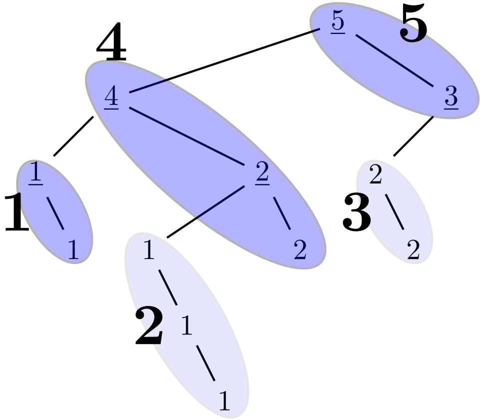
$$\begin{aligned} a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\ &= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|. \end{aligned}$$



$$\begin{bmatrix} 2 & & & & \\ 3 & & & & \\ & 2 & & & \\ & 2 & & 1 & \\ & & 1 & & 1 \end{bmatrix}$$

The Fishburn matrix $A = (a_{ij})$ associated with T is:

$$\begin{aligned} a_{ij} &= |\{\text{nodes with label } j \text{ contained in the } i\text{th path of } T\}| \\ &= |\{v : \mathfrak{l}(v) = i, \mathfrak{b}(v) = j\}|. \end{aligned}$$



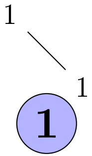
$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

From Fishburn matrices to Fishburn trees

$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

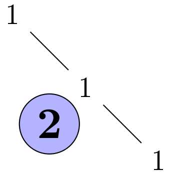
From Fishburn matrices to Fishburn trees

$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

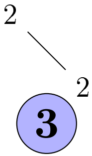


From Fishburn matrices to Fishburn trees

$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

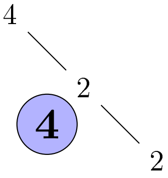


$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$



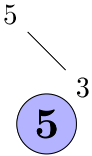
From Fishburn matrices to Fishburn trees

$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

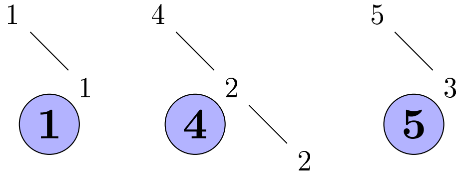


From Fishburn matrices to Fishburn trees

$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

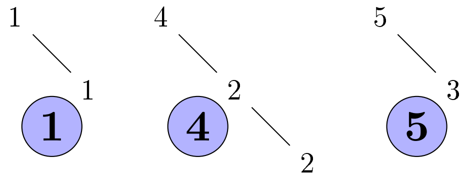


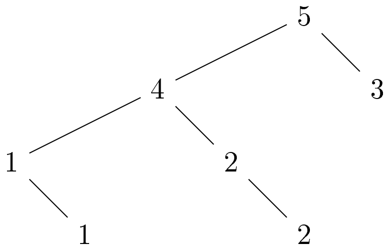
$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$



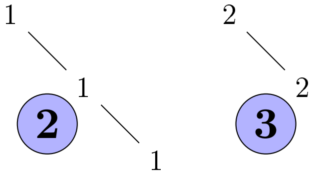
From Fishburn matrices to Fishburn trees

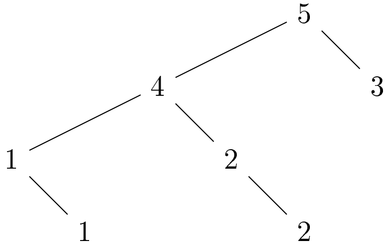
$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$





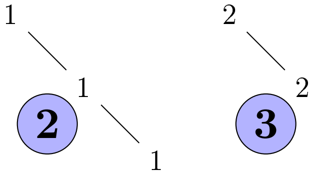
$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

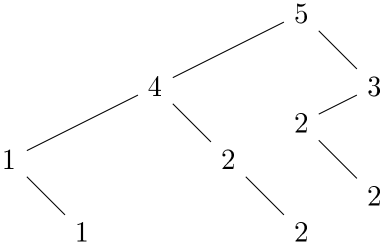




From Fishburn matrices to Fishburn trees

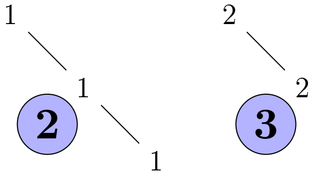
$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

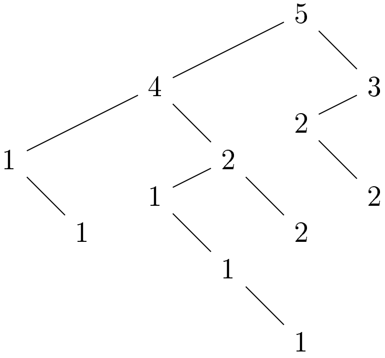




From Fishburn matrices to Fishburn trees

$$\begin{bmatrix} 2 & & & & \\ 3 & \cdot & & & \\ \cdot & 2 & \cdot & & \\ \cdot & 2 & \cdot & 1 & \\ \cdot & \cdot & 1 & \cdot & 1 \end{bmatrix}$$

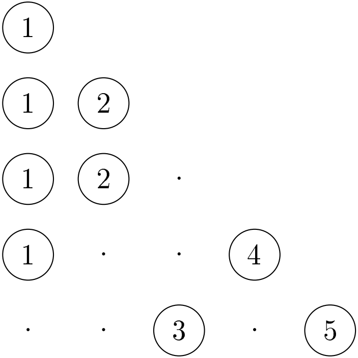




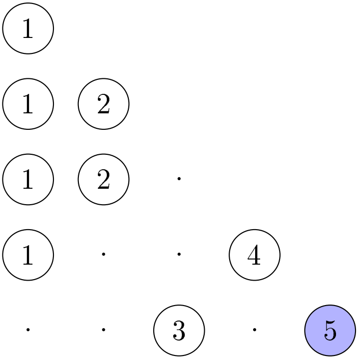
“Embedding” of trees on binary matrices

1				
1	1			
1	1	·		
1	·	·	1	
·	·	1	·	1

1				
1	1			
1	1	·		
1	·	·	1	
·	·	1	·	1

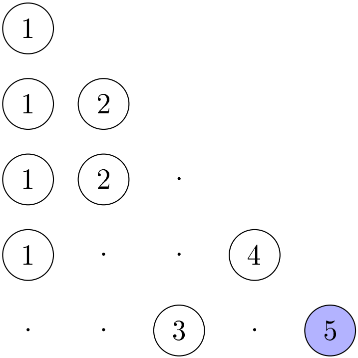

1
1 2
1 2
1 4
3 5

"Embedding" of trees on binary matrices



-Root: bottom-right node

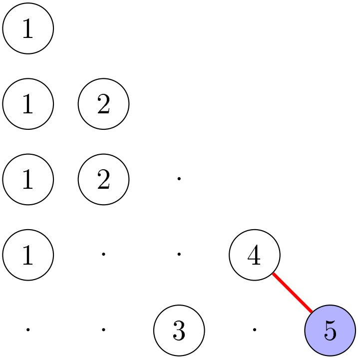
“Embedding” of trees on binary matrices



-Root: bottom-right node

-Left children: go up the diagonal

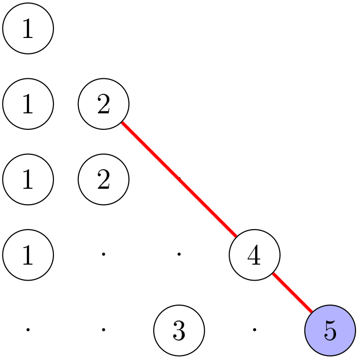
"Embedding" of trees on binary matrices



-Root: bottom-right node

-Left children: go up the diagonal

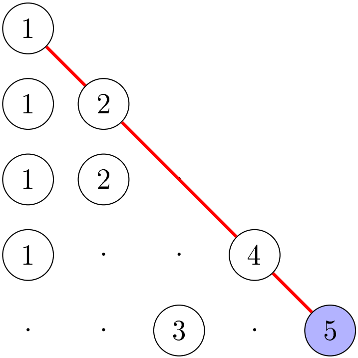
“Embedding” of trees on binary matrices



-Root: bottom-right node

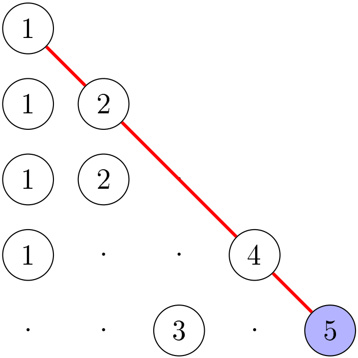
-Left children: go up the diagonal

“Embedding” of trees on binary matrices



-Root: bottom-right node

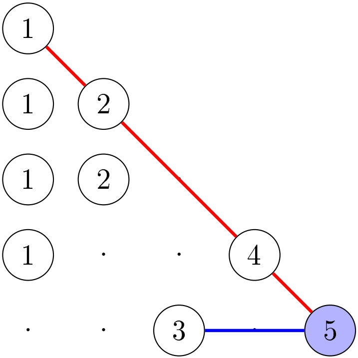
-Left children: go up the diagonal



-Root: bottom-right node

-Left children: go up the diagonal

-Right children: go back on each row

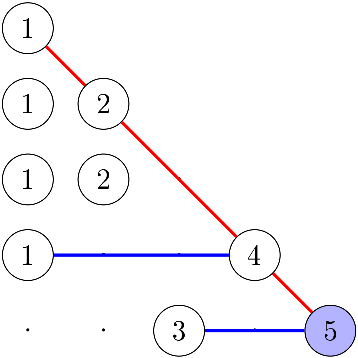


-Root: bottom-right node

-Left children: go up the diagonal

-Right children: go back on each row

"Embedding" of trees on binary matrices

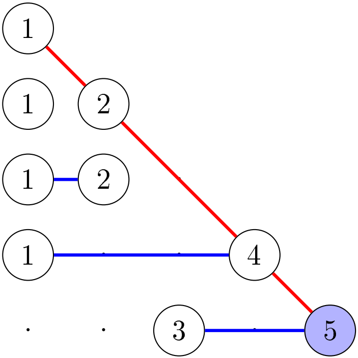


-Root: bottom-right node

-Left children: go up the diagonal

-Right children: go back on each row

"Embedding" of trees on binary matrices

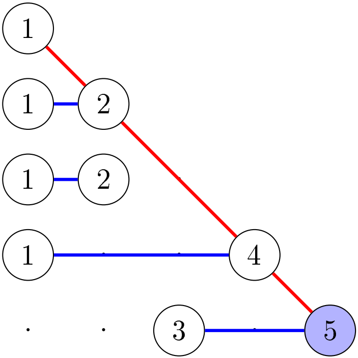


-Root: bottom-right node

-Left children: go up the diagonal

-Right children: go back on each row

"Embedding" of trees on binary matrices

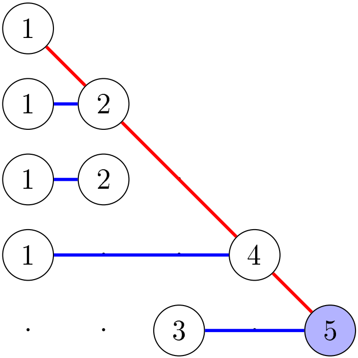


-Root: bottom-right node

-Left children: go up the diagonal

-Right children: go back on each row

"Embedding" of trees on binary matrices



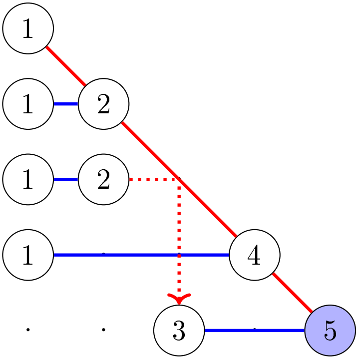
-Root: bottom-right node

-Left children: go up the diagonal

-Right children: go back on each row

-Left children: bounce off the diagonal

“Embedding” of trees on binary matrices



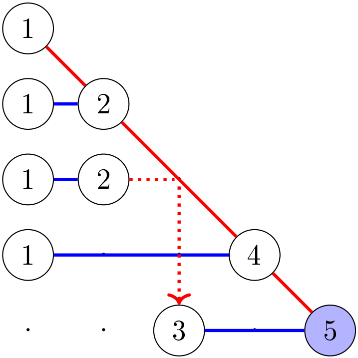
-Root: bottom-right node

-Left children: go up the diagonal

-Right children: go back on each row

-Left children: bounce off the diagonal

"Embedding" of trees on binary matrices



-Root: bottom-right node

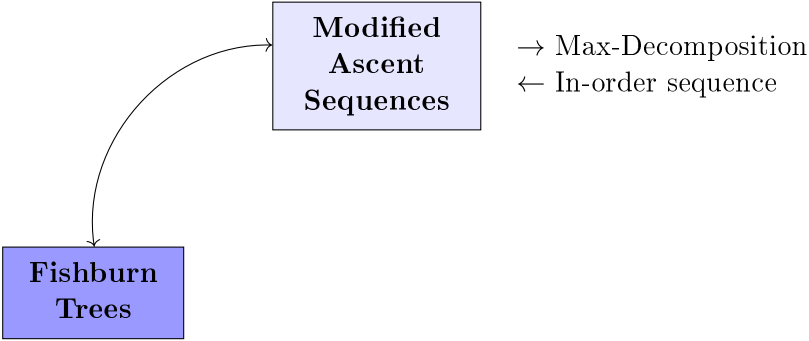
-Left children: go up the diagonal

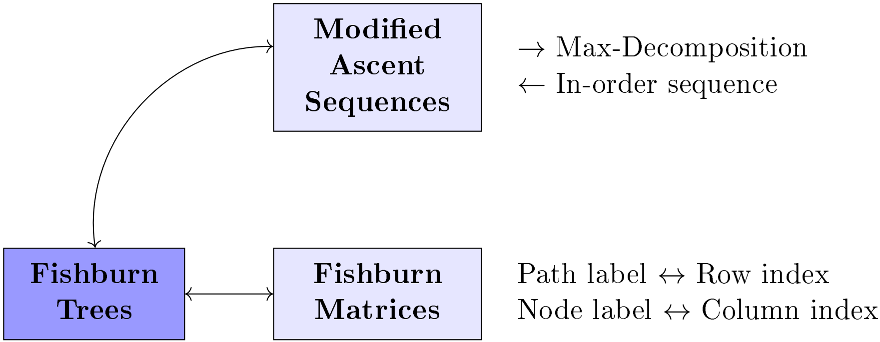
-Right children: go back on each row

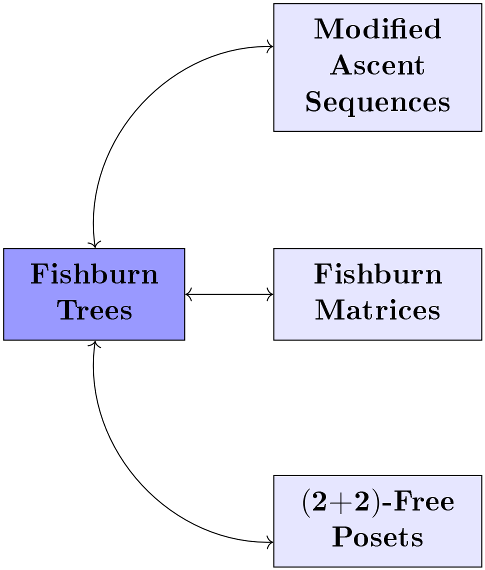
-Left children: bounce off the diagonal

Rotate the matrix by 90° to obtain the Fishburn tree!

Fishburn Trees


Modified Ascent Sequences
→ Max-Decomposition
← In-order sequence
Fishburn Trees


Modified Ascent Sequences
→ Max-Decomposition
← In-order sequence
Fishburn Trees
Fishburn Matrices
Path label ↔ Row index
Node label ↔ Column index



→ Max-Decomposition
← In-order sequence

Path label ↔ Row index
Node label ↔ Column index

Path label ↔ Down-sets
Node label ↔ Levels

Applications: flip and sum

flip and sum

- Duality (`flip`) acts as an involution on Fishburn posets.
- On Fishburn matrices, the `flip` corresponds to the reflection of a matrix in its antidiagonal.
- The `sum` of two Fishburn matrices is a Fishburn matrix.

Applications: `flip` and `sum`

`flip` and `sum`

- Duality (`flip`) acts as an involution on Fishburn posets.
- On Fishburn matrices, the `flip` corresponds to the reflection of a matrix in its antidiagonal.
- The `sum` of two Fishburn matrices is a Fishburn matrix.

1. How do `flip` and `sum` act on the corresponding ascent sequences?
2. How do `flip` and `sum` act on Fishburn trees?
3. Is there a natural involution on the set of Fishburn trees?

Applications: Subfamilies

Fishburn trees	**Modified seq.**	**Fishburn mat.**	**(2+2)-free posets**
Strictly decreasing	Primitive	Binary	Primitive
Comb-shaped	Self-modified	Positive diagonal	∃ Chain of max length
(⋆)	$\hat{\mathcal{A}}(212, 312)$	NW-free	N-free (Series-parallel)
(†)	$\hat{\mathcal{A}}(231)$	SW-free	(**3**+**1**)-free (Semiorders)
By intersection of the above two rows	$\hat{\mathcal{A}}(212, 312, 231)$	(NW,SW)-free	N- and (**3**+**1**)-free

$$(\star)\ \nexists\, u, v : \mathfrak{l}(u) < \mathfrak{l}(v) \leq \mathfrak{b}(u) < \mathfrak{b}(v)$$

$$(\dagger)\ \nexists\, u, v : \mathfrak{b}(u) < \mathfrak{b}(v),\ \mathfrak{l}(u) > \mathfrak{l}(v)$$

Applications: Statistics

Fishburn trees	Modified seq.	Fishburn mat.	(2+2)-free posets
# nodes	Length	Sum of entries	# elements
Biggest node label	Maximum value	# rows/cols	# levels
$\lvert\{u : \mathfrak{l}(u) = j\}\rvert$	# copies of j	$\sum_i a_{i,j}$	$\lvert L_j\rvert$
$\lvert\{u : \mathfrak{b}(u) = \mathfrak{l}(u)\}\rvert$	# weak ltr-max	Trace	$\sum_i \lvert L_i \cap D_{i+1}\rvert$
$\lvert\{u : \mathfrak{l}(u) = 1\}\rvert$	# copies of 1	$\sum_i a_{i,1}$	# minimal elements
$\lvert\text{rpath}(r)\rvert$	# weak rtl-max	$\sum_j a_{k,j}$	# maximal elements
$\mathfrak{l}(v_n)$	x_n	index	Minimal level of a maximal element

Applications: Statistics

Fishburn trees	Modified seq.	Fishburn mat.	(2+2)-free posets
# nodes	Length	Sum of entries	# elements
Biggest node label	Maximum value	# rows/cols	# levels
$\lvert\{u : \mathfrak{l}(u) = j\}\rvert$	# copies of j	$\sum_i a_{i,j}$	$\lvert L_j \rvert$
$\lvert\{u : \mathfrak{b}(u) = \mathfrak{l}(u)\}\rvert$	# weak ltr-max	Trace	$\sum_i \lvert L_i \cap D_{i+1}\rvert$
$\lvert\{u : \mathfrak{l}(u) = 1\}\rvert$	# copies of 1	$\sum_i a_{i,1}$	# minimal elements
$\lvert\text{rpath}(r)\rvert$	# weak rtl-max	$\sum_j a_{k,j}$	# maximal elements
$\mathfrak{l}(v_n)$	x_n	index	Minimal level of a maximal element

Thanks!