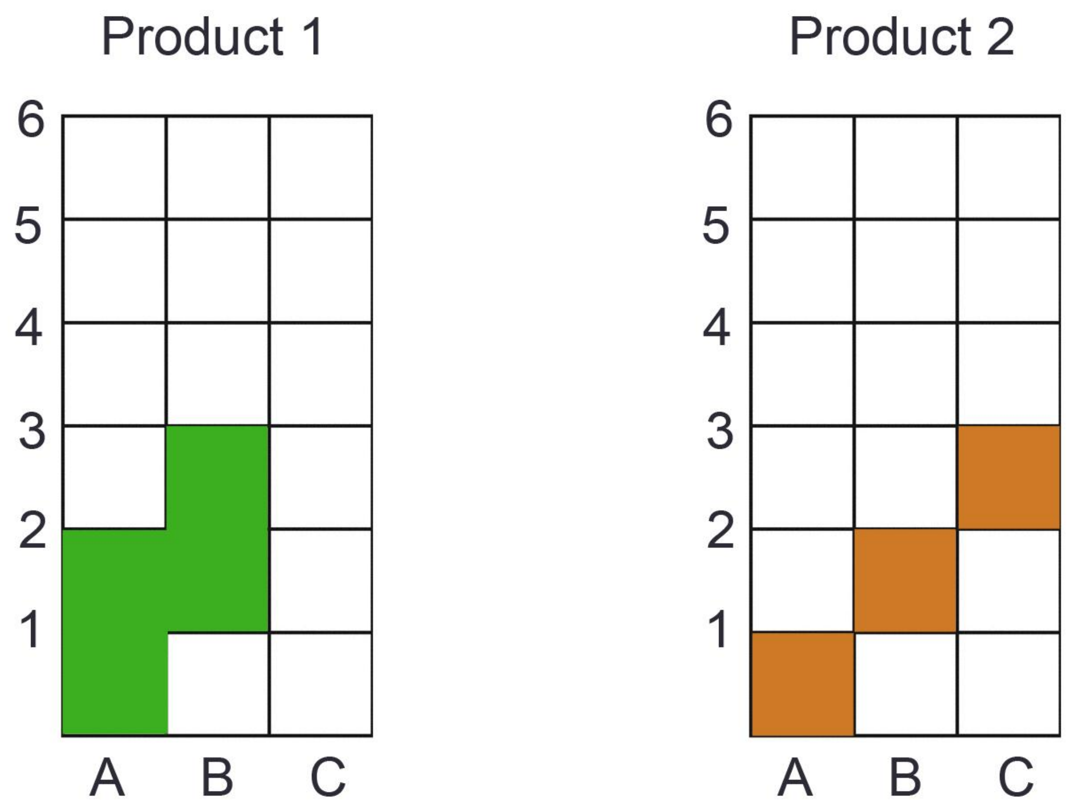
Exploring the Inherent Memory Interval of Tetris-like Systems

Nicholai Christensen

Information & Decision Algorithms Laboratories

Tetris



Information & Decision Algorithms Laboratories

Max-plus Operators

$$a \otimes b = a + b$$

$$a \oplus b = max(a, b)$$

Max-plus Operators

$$e = 0$$

$$\epsilon = -\infty$$

$$a \otimes b = a + b$$

$$a \oplus b = max(a, b)$$

Matrix Multiplication

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mp} \end{pmatrix}$$

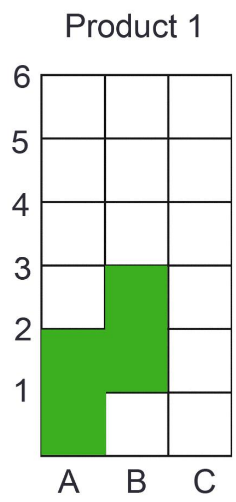
$$\mathbf{AB} = \begin{pmatrix} (AB)_{11} & (AB)_{12} & \cdots & (AB)_{1p} \\ (AB)_{21} & (AB)_{22} & \cdots & (AB)_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ (AB)_{n1} & (AB)_{n2} & \cdots & (AB)_{np} \end{pmatrix}$$

$$[A \otimes B]_{ij} = \bigoplus_{k=1}^{p} [A]_{ik} \otimes [B]_{kj} = \max([A]_{i1} + [B]_{1j}, \ldots, [A]_{ip} + [B]_{pj})$$

Contours

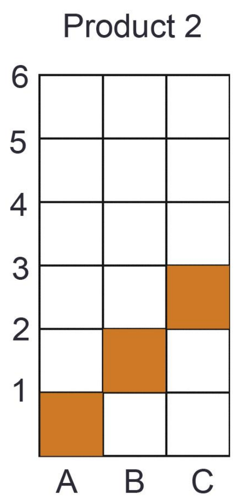
$U(1)=(2, 3, \varepsilon)^T$

$L(1)=(e, 1, \varepsilon)^T$



$U(1)=(1, 2, 3)^T$

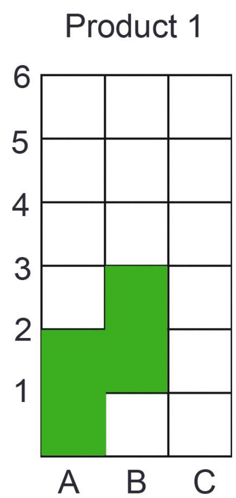
$L(1)=(e, 1, 2)^T$



Contours

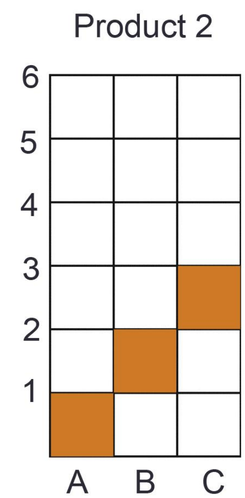
$U(1)=(2, 3, \varepsilon)^T$

$L(1)=(e, 1, \varepsilon)^T$



$U(1)=(1, 2, 3)^T$

$L(1)=(e, 1, 2)^T$

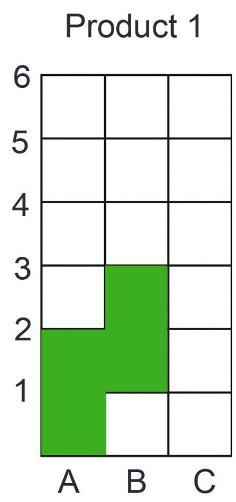


$$I = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

Contours

$U(1)=(2, 3, \varepsilon)^T$

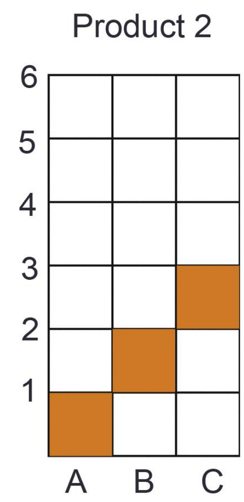
$L(1)=(e, 1, \varepsilon)^T$



$$I = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

$L(1)=(e, 1, 2)^T$

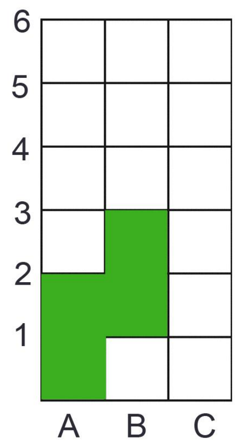


Contours

$U(1)=(2, 3, \varepsilon)^T$

$L(1)=(e, 1, \varepsilon)^T$

Product 1

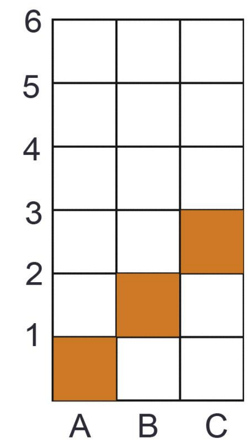


$$P_1 = \begin{bmatrix} 2 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

$L(1)=(e, 1, 2)^T$

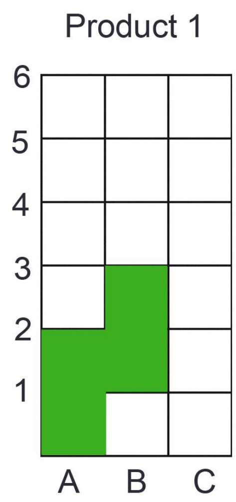
Product 2



Contours

$U(1)=(2, 3, \varepsilon)^T$

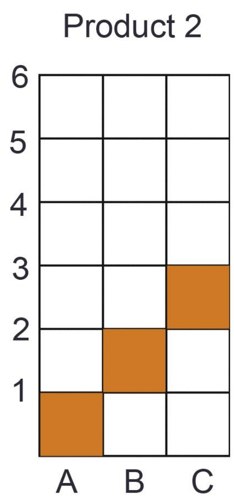
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

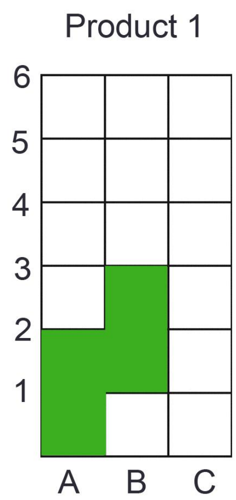
$L(1)=(e, 1, 2)^T$



Contours

$U(1)=(2, 3, \varepsilon)^T$

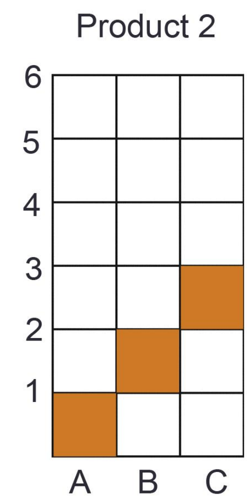
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

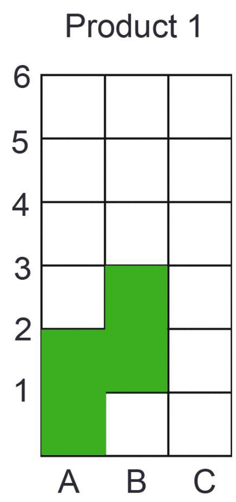
$L(1)=(e, 1, 2)^T$



Contours

$U(1)=(2, 3, \varepsilon)^T$

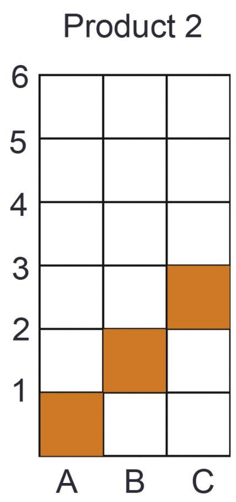
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

$L(1)=(e, 1, 2)^T$

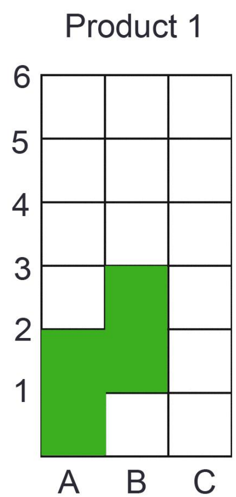


Information & Decision Algorithms Laboratories

Contours

$U(1)=(2, 3, \varepsilon)^T$

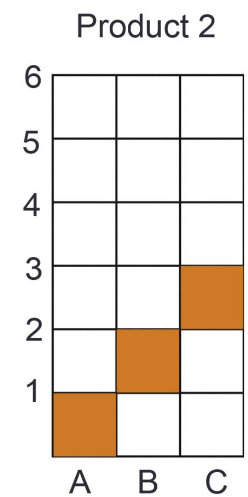
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

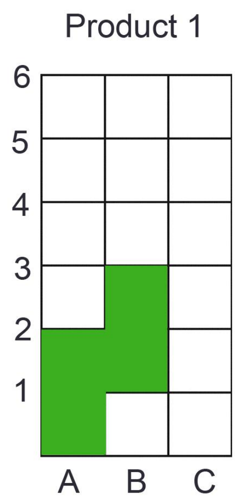
$L(1)=(e, 1, 2)^T$



Contours

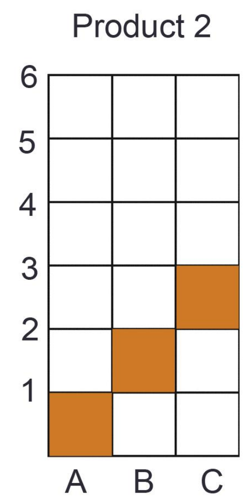
$U(1)=(2,\ 3,\ \varepsilon)^T$

$L(1)=(e,\ 1,\ \varepsilon)^T$



$U(1)=(1,\ 2,\ 3)^T$

$L(1)=(e,\ 1,\ 2)^T$

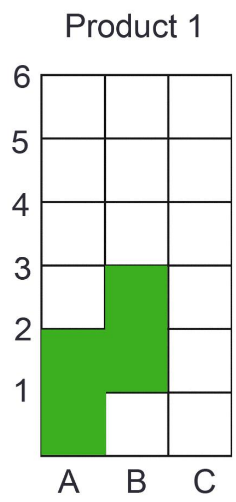


$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

Contours

$U(1)=(2, 3, \varepsilon)^T$

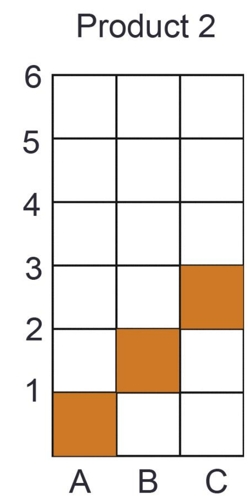
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

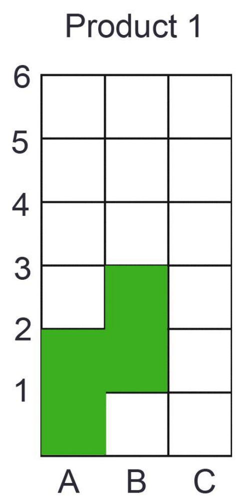
$L(1)=(e, 1, 2)^T$



Contours

$U(1)=(2,\ 3,\ \varepsilon)^T$

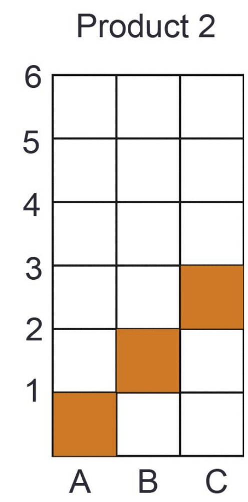
$L(1)=(e,\ 1,\ \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1,\ 2,\ 3)^T$

$L(1)=(e,\ 1,\ 2)^T$

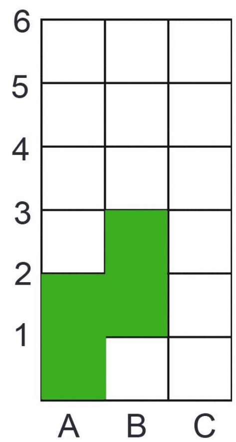


Contours

$U(1)=(2, 3, \varepsilon)^T$

$L(1)=(e, 1, \varepsilon)^T$

Product 1

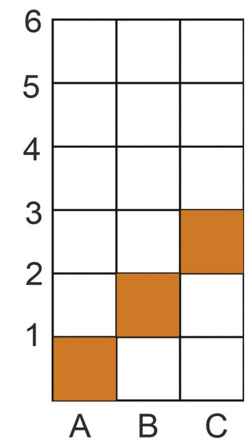


$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & 2 & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

$L(1)=(e, 1, 2)^T$

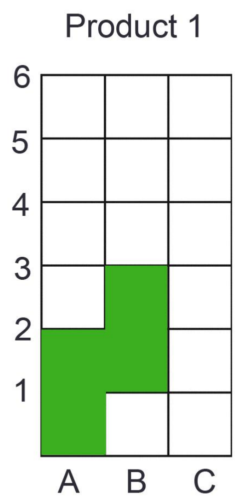
Product 2



Contours

$U(1)=(2, 3, \varepsilon)^T$

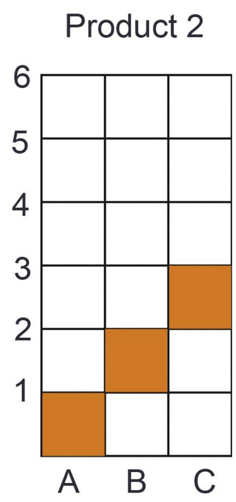
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & 2 & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

$L(1)=(e, 1, 2)^T$

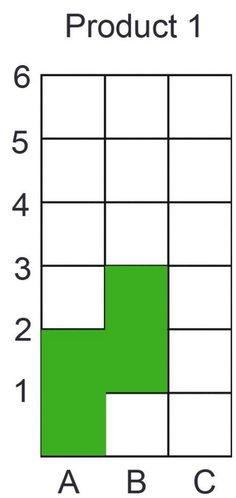


IDeA Labs
Information & Decision Algorithms Laboratories

Contours

$U(1)=(2,\ 3,\ \varepsilon)^T$

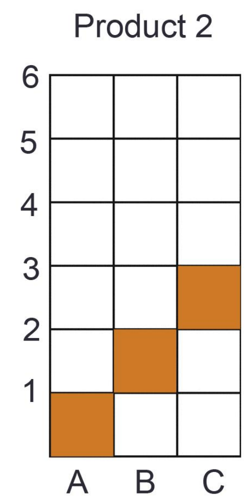
$L(1)=(e,\ 1,\ \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & 2 & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1,\ 2,\ 3)^T$

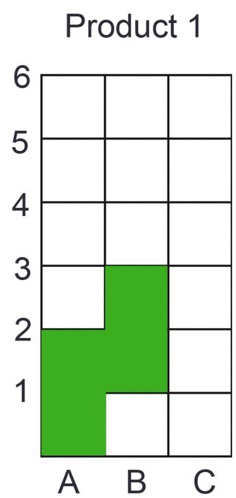
$L(1)=(e,\ 1,\ 2)^T$



Contours

$U(1)=(2, 3, \varepsilon)^T$

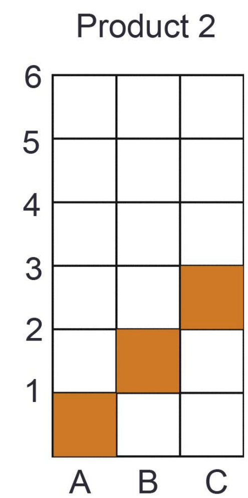
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & 2 & \varepsilon \\ \color{red}{\varepsilon} & \color{red}{\varepsilon} & \color{red}{e} \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

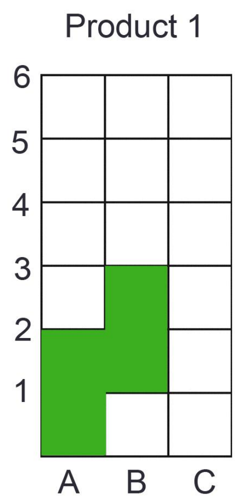
$L(1)=(e, 1, 2)^T$



Contours

$U(1)=(2, 3, \varepsilon)^T$

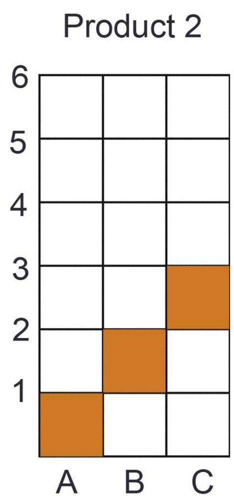
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & 2 & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

$U(1)=(1, 2, 3)^T$

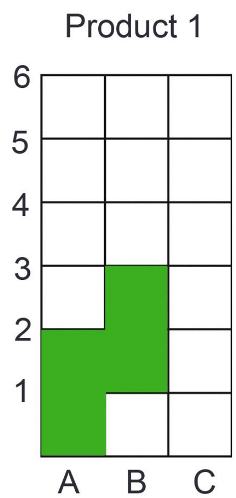
$L(1)=(e, 1, 2)^T$



Contours

$U(1)=(2, 3, \varepsilon)^T$

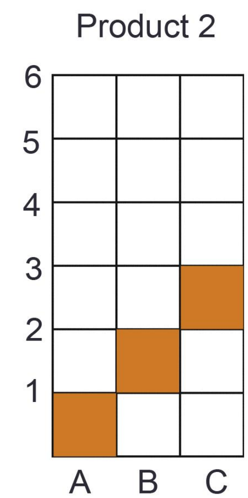
$L(1)=(e, 1, \varepsilon)^T$



$$P_1 = \begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & 2 & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix}$$

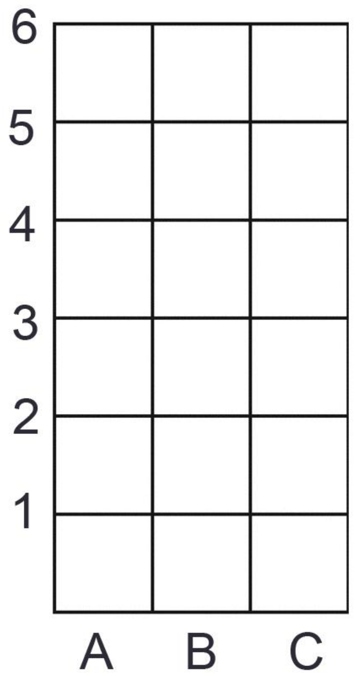
$U(1)=(1, 2, 3)^T$

$L(1)=(e, 1, 2)^T$



$$P_2 = \begin{bmatrix} 1 & e & - \\ 1 & & \\ 2 & 1 & e \\ 3 & 2 & 1 \end{bmatrix}$$

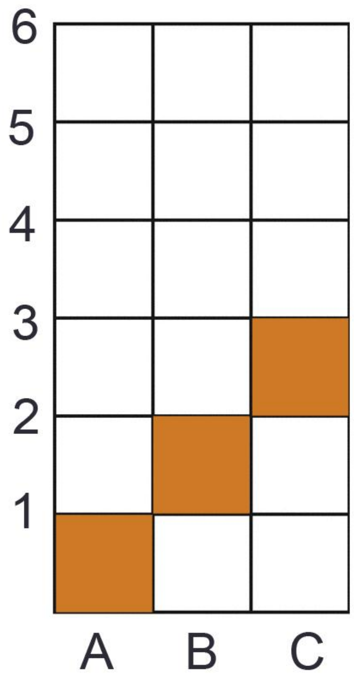
Heap Model



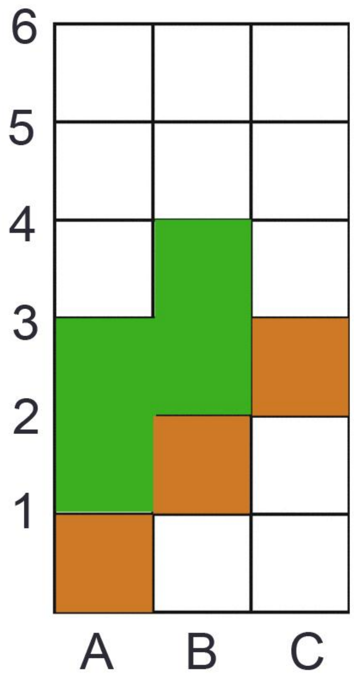
$$x_0 = \begin{bmatrix} e \\ e \\ e \end{bmatrix}$$

Heap Model

$$\begin{bmatrix} 1 & e & -1 \\ 2 & 1 & e \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} e \\ e \\ e \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$



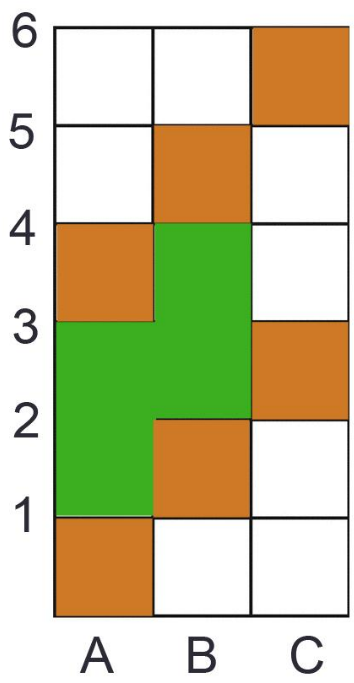
Heap Model



$$\begin{bmatrix} 1 & e & -1 \\ 2 & 1 & e \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} e \\ e \\ e \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & 2 & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 3 \end{bmatrix}$$

Heap Model



$$\begin{bmatrix} 1 & e & -1 \\ 2 & 1 & e \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} e \\ e \\ e \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & \varepsilon \\ 3 & 2 & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & e & -1 \\ 2 & 1 & e \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Future Work

- **The Inherent Memory Problem:** Given a portfolio of block shapes, how many previous blocks do you need to look back to compute the top contour of the entire stack?
- **Approaches:**
  - Primitive matrices
  - Cushion
- **Importance:** Saves computation in scheduling problems

Information & Decision Algorithms Laboratories