# The Elimination Tree of an Unsymmetric Matrix: Theory and Applications

**Stanley C. Eisenstat**
Yale University

**Joseph W. H. Liu**
York University

*CSC05 — 21 June 2005*

---

## *BASIC ASSUMPTIONS*

$A$ is a large, sparse, unsymmetric, and irreducible matrix with nonzero diagonal entries

Factorization $A = LU$ exists, where $L$ is unit lower triangular and $U$ is upper triangular

No accidental cancellation during factorization

No use of supernodes for efficiency

No pivoting for stability {will be relaxed later}

## *ELIMINATION TREE FOR SYMMETRIC A* [Schreiber]

**Parent Function Defines $T(A)$**

$$\rho(k) \equiv \min\left\{x \mid x > k \text{ and } x \stackrel{L}{\longmapsto} k \stackrel{L^t}{\longmapsto} x\right\}$$

$\{\rho(k)$ is the row index of the first nonzero below the diagonal in the $k$-th column of $L\}$

**Property**: The vertices in the subtree $\mathcal{T}[k]$ rooted at $k$ form a connected component of the subgraph of the undirected graph $G(A)$ induced by $\{1, 2, \ldots, k\}$

---

## *ELIMINATION TREE FOR UNSYMMETRIC A* [E+L]

**Parent Function Defines $T(A)$**

$$\rho(k) \equiv \min\left\{x \mid x > k \text{ and } x \stackrel{L}{\Longrightarrow} k \stackrel{U}{\Longrightarrow} x\right\}$$

{reduces to the previous definition if $A$ is symmetric}

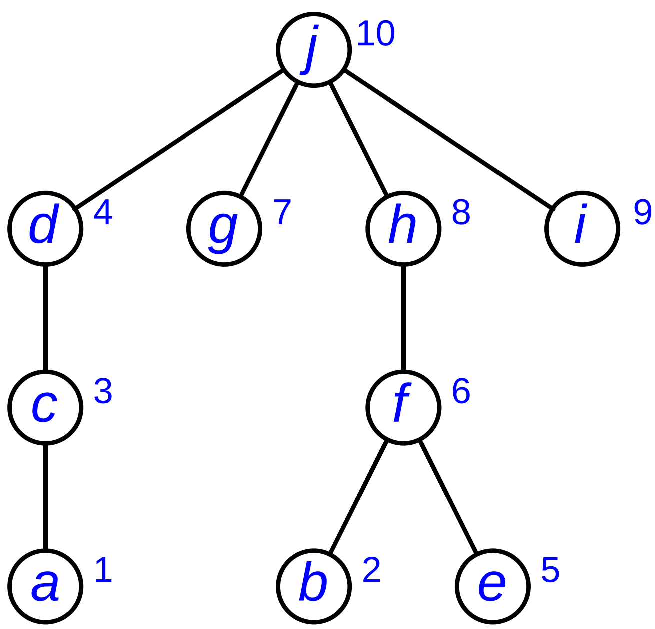
**Property**: The vertices in $\mathcal{T}[k]$ form a strongly connected component of the subgraph of the directed graph $G(A)$ induced by $\{1, 2, \ldots, k\}$

# *CANONICAL EXAMPLE*

## Filled Matrix

$$A^{+} \equiv L + U - I = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \begin{pmatrix} a & & \bullet & & & & & & & \\ \bullet & b & \circ & & \bullet & & & \bullet & & \bullet \\ \bullet & & c & \bullet & & & & & & \\ & & \bullet & d & & & & & \bullet & \\ & & & \bullet & e & \bullet & & \bullet & \circ & \bullet \\ & \bullet & \circ & \circ & \circ & f & & \circ & \circ & \circ \\ & & & \bullet & & & g & & \circ & \\ \bullet & \bullet & \bullet & \circ & \circ & \circ & & h & \circ & \circ \\ & & & & & & & & i & \bullet \\ & & & & \bullet & \circ & \bullet & \circ & \circ & j \end{pmatrix} \end{array}$$

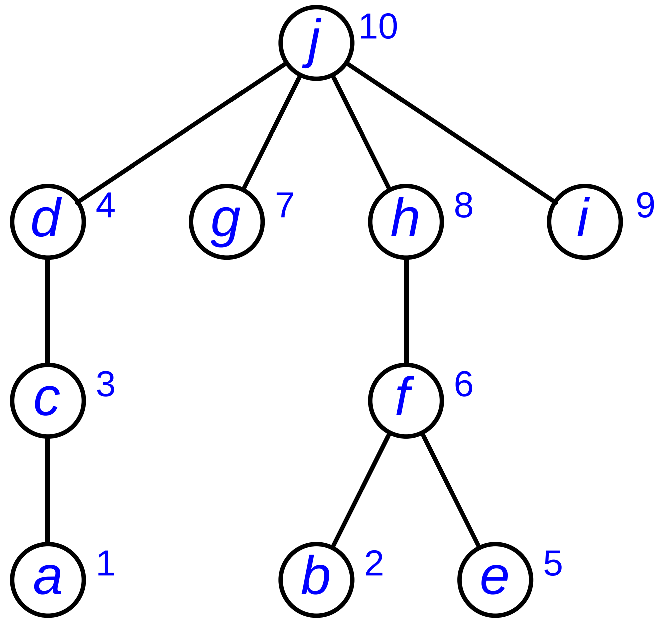
## Elimination Tree



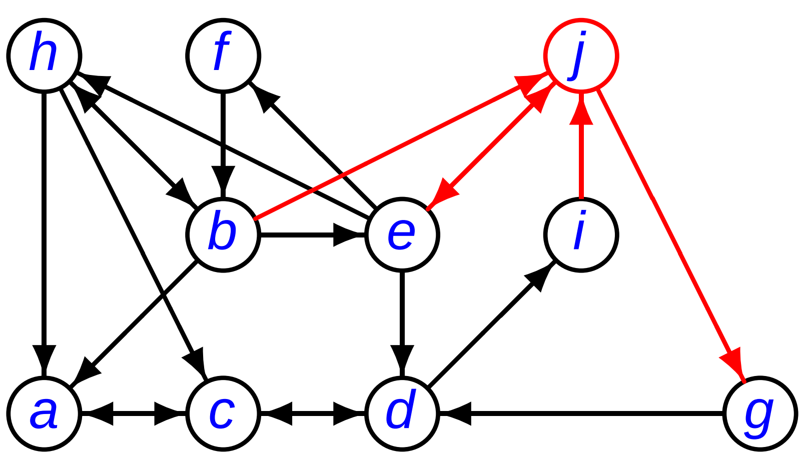

$j \overset{L}{\Longrightarrow} d \overset{U}{\Longrightarrow} j$ but neither $j \overset{L}{\longmapsto} d$ nor $d \overset{U}{\longmapsto} j$

# *CANONICAL EXAMPLE (continued)*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | *a* | | ● | | | | | | | |
| 2 | ● | *b* | | | ● | | | ● | | ● |
| 3 | ● | | *c* | ● | | | | | | |
| 4 | | | ● | *d* | | | | | ● | |
| 5 | | | | ● | *e* | ● | | ● | | ● |
| 6 | | ● | | | | *f* | | | | |
| 7 | | | | ● | | | *g* | | | |
| 8 | ● | ● | ● | | | | | *h* | | |
| 9 | | | | | | | | | *i* | ● |
| 10 | | | | | ● | | ● | | | *j* |

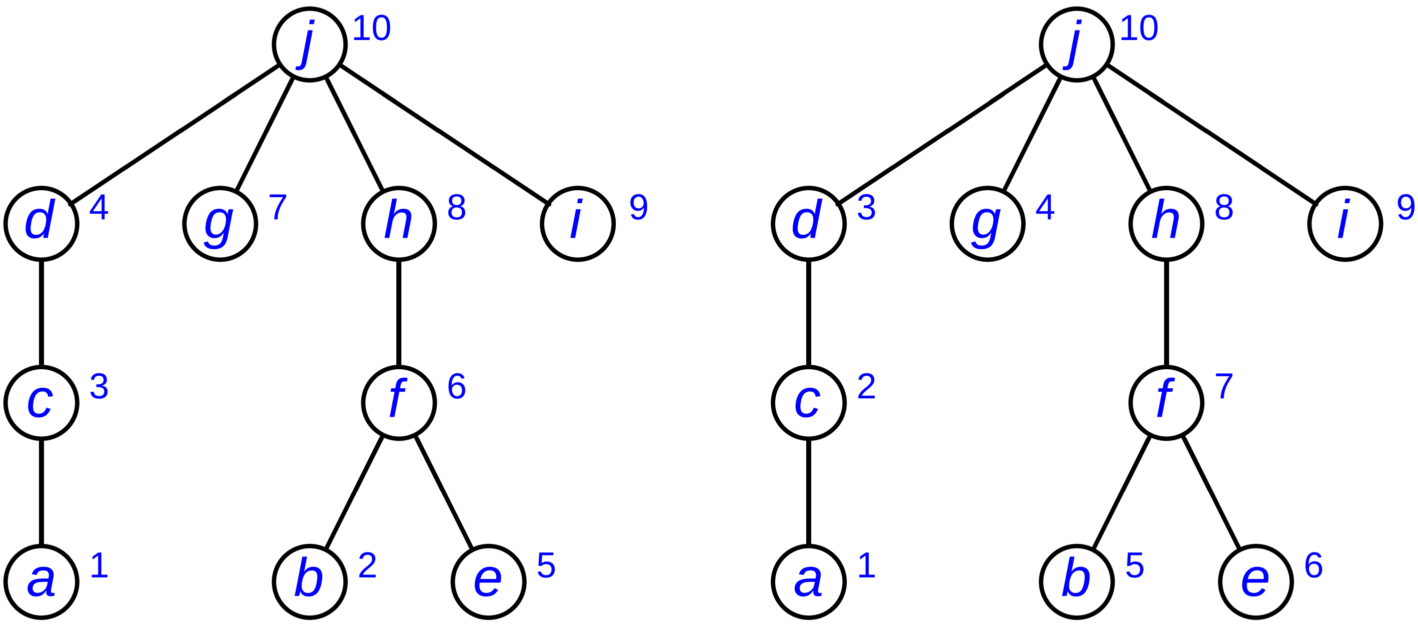



**Property**: The vertices in $\mathcal{T}[k]$ form a strongly connected component of the subgraph of the directed graph $G(A)$ induced by any set of vertices that contains $\mathcal{T}[k]$ but no ancestors of $k$ in $T(A)$



## *POSTORDERING OF $T(A)$*

If $c_1, \ldots, c_t$ are the children of $k$ in $T(A)$, then number the vertices within each subtree $\mathcal{T}[c_i]$ consecutively and number $k$ immediately after its descendants




Postordering does not preserve the filled graph
{more fill and work in practice?}

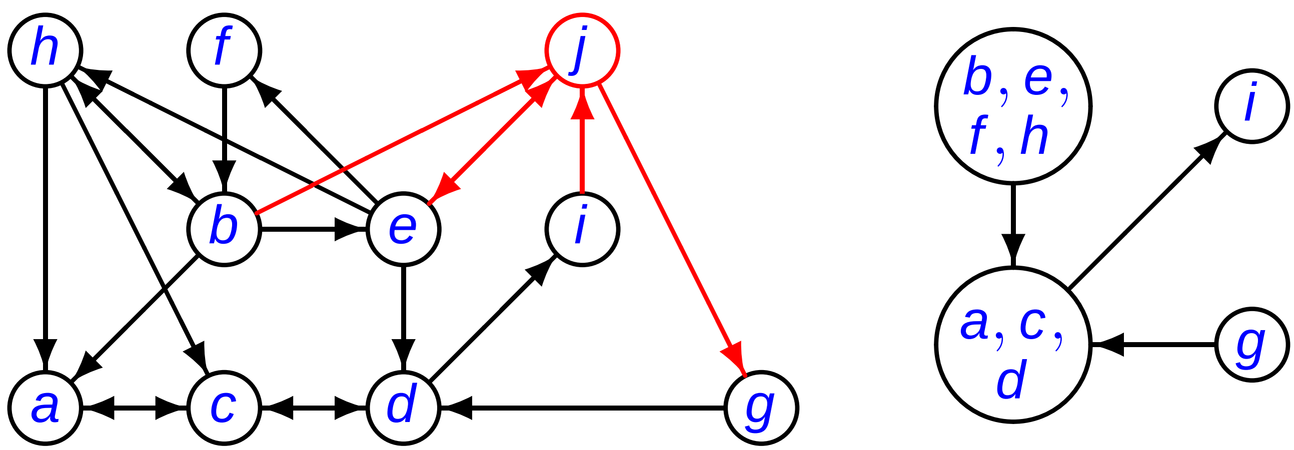
Postordering does preserve the structure of $T(A)$; i.e., $T(PAP^t)$ is isomorphic to $T(A)$

Postordering does preserve the values of the diagonal elements of $U$; i.e., $\text{diag}(U) = P^t \ \text{diag}(U(PAP^t))\, P$
{impact on numerical stability?}

## *UPPER BBT POSTORDERING*

Postordering does not specify the order in which the subtrees $\mathcal{T}[c_i]$ themselves are numbered

From the subgraph of $G(A)$ induced by the vertices in $\mathcal{T}[k] \setminus \{k\} = \cup_{i=1,t} \mathcal{T}[c_i]$, form the quotient graph $\mathcal{Q}_k(A)$ by coalescing the vertices in each $\mathcal{T}[c_i]$

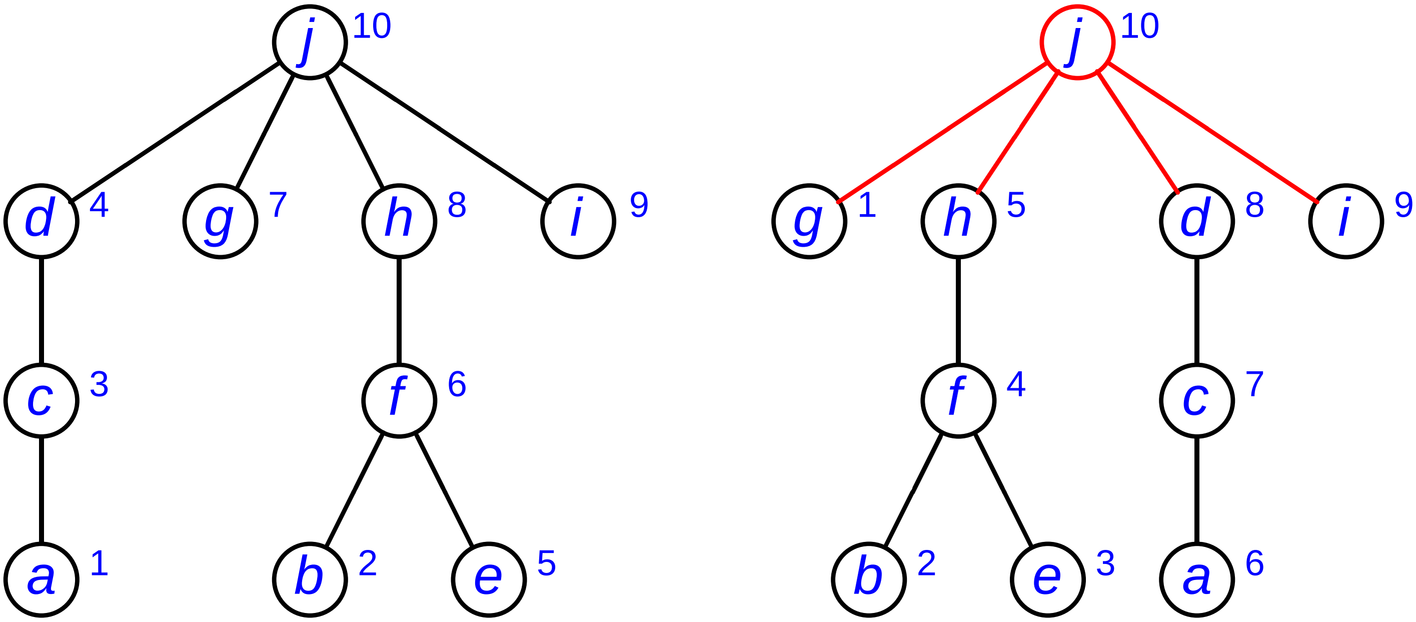



$\mathcal{Q}_k(A)$ is acyclic $\{$since each $\mathcal{T}[c_i]$ is a component$\}$

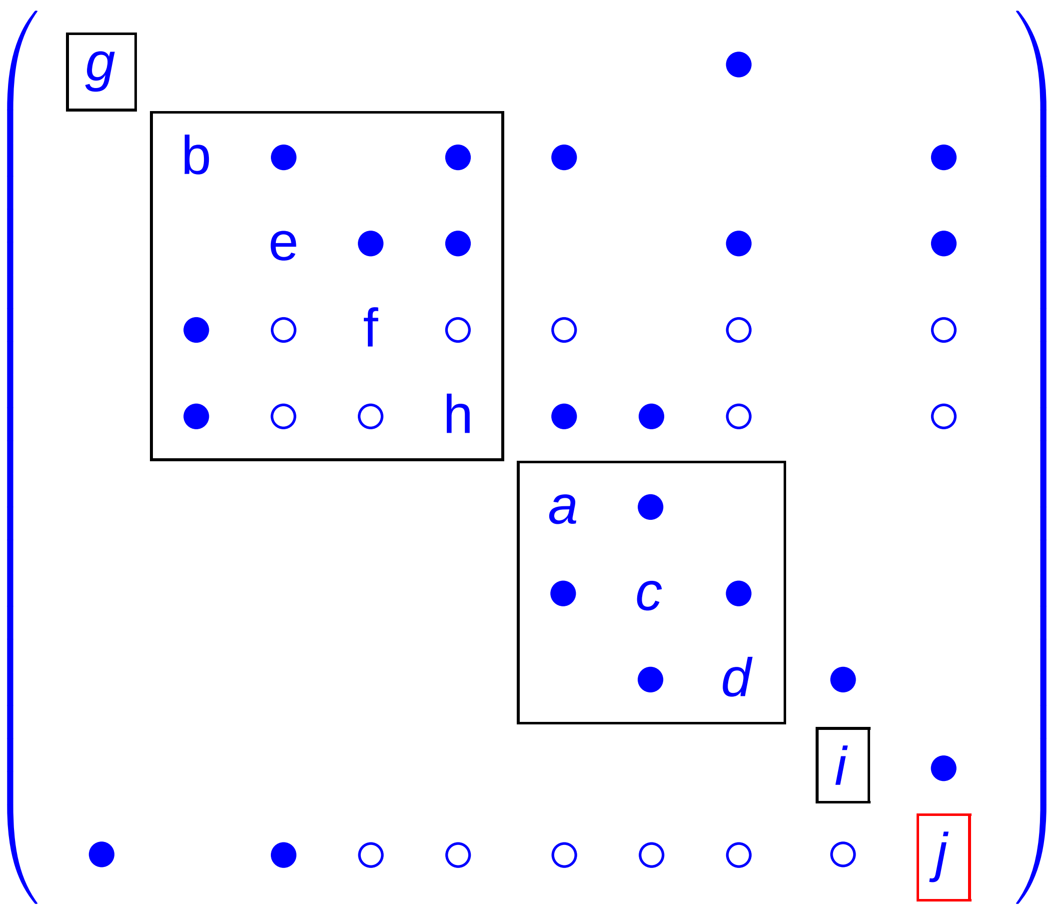
If we use a topological sort of $\mathcal{Q}_k(A)$ to order the $\mathcal{T}[c_i]$ during the postordering, every edge in $G(A)$ from a vertex in $\mathcal{T}[c_i]$ to a vertex in $\mathcal{T}[c_j]$ ($j \neq i$) goes from the lower-numbered to the higher-numbered vertex

Thus the permuted matrix is in upper bordered block triangular (BBT) form

# *UPPER BBT POSTORDERING (continued)*




$(PAP^t)^+ =$




{the bordered block triangular structure is recursive}

# *UPWARD-LOOKING LU FACTORIZATION*

## Algorithm

**for** $k$ := 1 **to** $n$ **do**
  $z = A_{k*}$
  **for** $i$ := 1 **to** $k - 1$ **do** {sparsely}
    **if** $z_i \neq 0$ **then**
      $\ell_{ki} = z_i / u_{ii};\ z = z - \ell_{ki} \times U_{i*}$
    **end if**
  **end for**
  $U_{k*} = z$
**end for**

If $\ell_{ki} \neq 0$, then the $k$-th row of $L$ and $U$ depends on the $i$-th row of $U$ (and, indirectly, the $i$-th row of $L$)

Therefore $G(L)$ captures the data dependencies among the rows of $A^+$ {i.e., the potential parallelism}

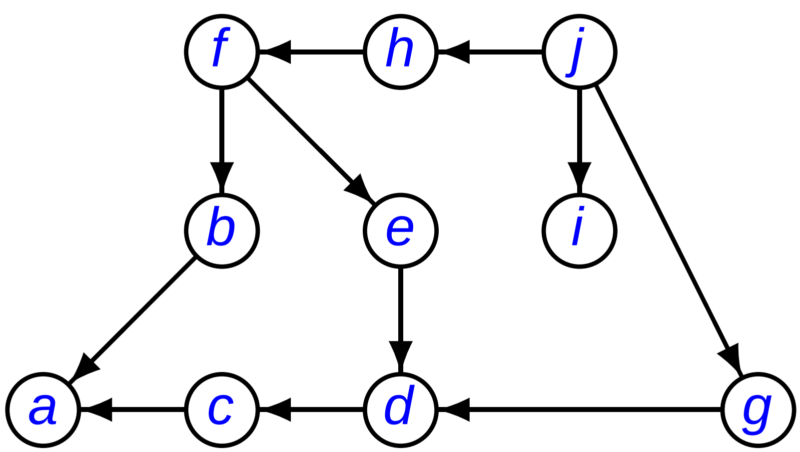
Its transitive reduction $G(L^o)$ preserves paths and thus is a minimal representation of these dependencies {$G(L^o)$ is one of the elimination dags [Gilbert+L]}

# *UPWARD-LOOKING FACTORIZATION (continued)*

## Filled Matrix

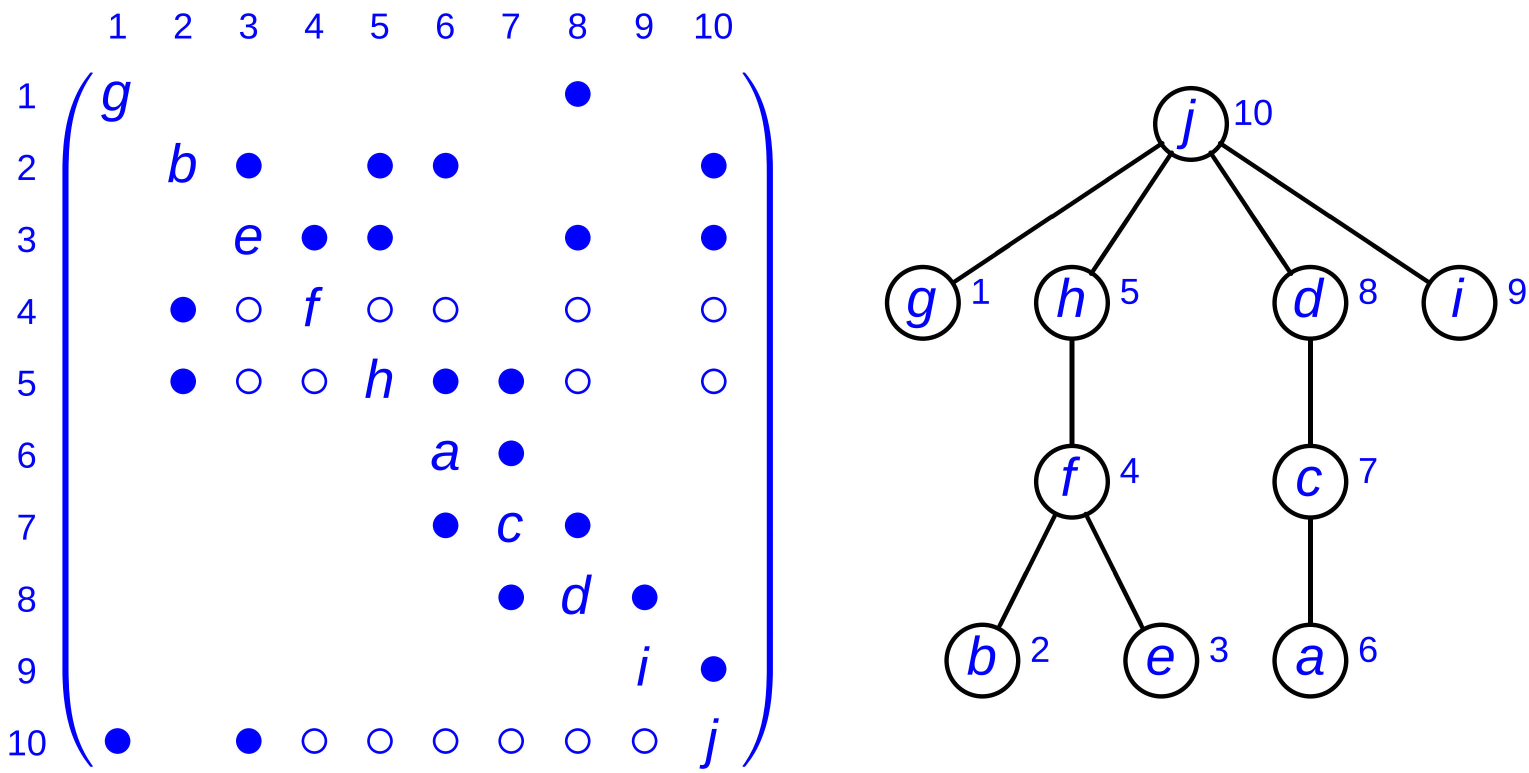
$$
A^{+} = \begin{array}{c|cccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & a & & \bullet & & & & & & & \\
2 & \bullet & b & \circ & & \bullet & & & \bullet & & \bullet \\
3 & \bullet & & c & \bullet & & & & & & \\
4 & & & \bullet & d & & & & & \bullet & \\
5 & & & & \bullet & e & \bullet & & \bullet & \circ & \bullet \\
6 & & \bullet & \circ & \circ & \circ & f & & \circ & \circ & \circ \\
7 & & & & \bullet & & & g & & \circ & \\
8 & \bullet & \bullet & \bullet & \circ & \circ & \circ & & h & \circ & \circ \\
9 & & & & & & & & & i & \bullet \\
10 & & & & & \bullet & \circ & \bullet & \circ & \circ & j
\end{array}
$$

## Elimination DAG $G(L^{o})$

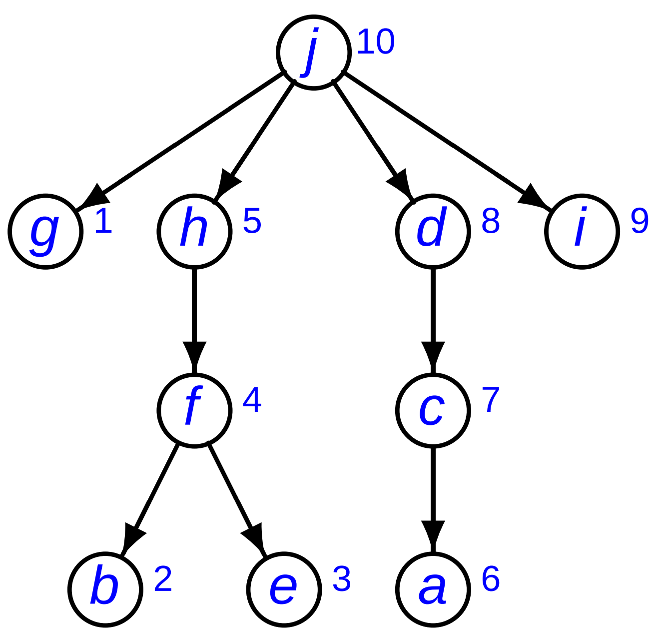


# *UPWARD-LOOKING FACTORIZATION (continued)*

**Definition:** $A$ is upper BBT ordered if the natural ordering is an upper BBT postordering of $T(A)$.



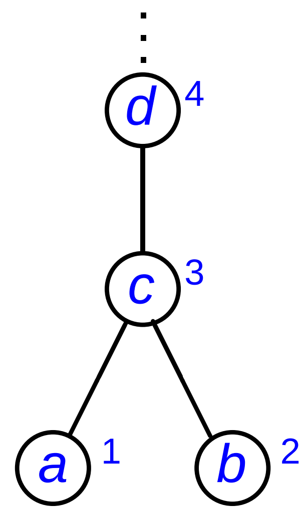

**Theorem:** If $A$ is upper BBT ordered, then $G(L^o)$ is $T(A)$ with edges directed from parent to child.



# ***SYMMETRIC MULTIFRONTAL*** [Speelpenning]

**Goal:** Use dense matrices as much as possible

$$A^{+} \equiv \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} a & & \bullet & & & \bullet \\ & b & \bullet & \bullet & & \\ \bullet & \bullet & c & \circ & \bullet & \circ \\ & \bullet & \circ & d & \bullet & \circ \\ & & \bullet & \bullet & e & \bullet \\ \bullet & & \circ & \circ & \bullet & f \end{pmatrix} \end{array}$$




$$\mathcal{F}_1 \equiv \begin{array}{c} \\ 1 \\ 3 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 6 \end{array} \\ \begin{pmatrix} a & \bullet & \bullet \\ \bullet & & \\ \bullet & & \end{pmatrix} \end{array}, \qquad \mathcal{U}_1 \equiv \begin{array}{c} \\ 3 \\ 6 \end{array} \begin{array}{c} \begin{array}{cc} 3 & 6 \end{array} \\ \begin{pmatrix} \circ & \circ \\ \circ & \circ \end{pmatrix} \end{array}$$

$$\mathcal{F}_2 \equiv \begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{pmatrix} b & \bullet & \bullet \\ \bullet & & \\ \bullet & & \end{pmatrix} \end{array}, \qquad \mathcal{U}_2 \equiv \begin{array}{c} \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cc} 3 & 4 \end{array} \\ \begin{pmatrix} \circ & \circ \\ \circ & \circ \end{pmatrix} \end{array}$$

$$\mathcal{F}_3 \equiv \begin{array}{c} \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} \bullet & \circ & \bullet & \circ \\ \circ & \circ & & \\ \bullet & & & \\ \circ & & & \circ \end{pmatrix} \end{array} = \begin{array}{c} \\ 3 \\ 5 \end{array} \begin{array}{c} \begin{array}{cc} 3 & 5 \end{array} \\ \begin{pmatrix} c & \bullet \\ \bullet & \end{pmatrix} \end{array} \updownarrow\!\!\!\!\leftrightarrow \mathcal{U}_1 \updownarrow\!\!\!\!\leftrightarrow \mathcal{U}_2$$

$\{ \updownarrow\!\!\!\!\leftrightarrow$ denotes the extend-add operator $\}$

# *SYMMETRIC MULTIFRONTAL (continued)*

## Algorithm

**for** $k := 1$ **to** $n$ **do**

Assemble frontal matrix $\mathcal{F}_k$ from the nonzeros in $[a_{ik}]_{k \le i \le n}$ and all updates $\mathcal{U}_c$ with $\rho(c) = k$

Factor $\mathcal{F}_k = \begin{pmatrix} \ell_{kk} & \\ [\ell_{ik}]_{k < i \le n,\, \ell_{ik} \ne 0} & I \end{pmatrix} \times \begin{pmatrix} 1 & \\ & \mathcal{U}_k \end{pmatrix}$

$$\times \begin{pmatrix} \ell_{kk} & [\ell_{jk}]^t_{k < j \le n,\, \ell_{jk} \ne 0} \\ & I \end{pmatrix}$$

**end for**

## Why It Works

$\rho(k)$ is the first row/column to which $\mathcal{U}_k$ contributes; and (nesting property)

$$\text{struct}\,(\ell_{pk}, \ldots, \ell_{nk}) \subseteq \text{struct}\,(\ell_{pp}, \ldots, \ell_{np}),$$

with $p = \rho(k)$, so $\mathcal{F}_{\rho(k)}$ can accommodate all of $\mathcal{U}_k$
$\{$entries not in row/column $\rho(k)$ are added to $\mathcal{U}_{\rho(k)}\}$

## *UNSYMMETRIC MULTIFRONTAL?*

### Algorithm

**for** $k$ := $1$ **to** $n$ **do**

Assemble frontal matrix $\mathcal{F}_k$ from nonzeros in $[a_{ik}]_{k \le i \le n}$, $[a_{kj}]_{k \le j \le n}$, and all unassembled updates $\mathcal{U}_m$ that contribute to row or column $k$

$$\text{Factor } \mathcal{F}_k = \begin{pmatrix} \ell_{kk} & \\ [\ell_{ik}]_{k<i\le n,\, \ell_{ik}\ne 0} & I \end{pmatrix} \times \begin{pmatrix} 1 & \\ & \mathcal{U}_k \end{pmatrix} \times \begin{pmatrix} u_{kk} & [u_{kj}]_{k<j\le n,\, u_{kj}\ne 0} \\ & I \end{pmatrix}$$

**end for**

### Implicit Assumptions

a) Rows (resp., columns) of $\mathcal{F}_k$ correspond to rows $i$ where $\ell_{ik} \neq 0$ (resp., columns $j$ with $u_{kj} \neq 0$)

b) Each update matrix $\mathcal{U}_m$ can be assembled into a *single* frontal matrix

# *EXAMPLE*

$$A^{+} = \begin{array}{c|cccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & a & & \bullet & & & & & & & \\
2 & \bullet & b & \circ & & \bullet & & & \bullet & & \bullet \\
3 & \bullet & & c & \bullet & & & & & & \\
4 & & & \bullet & d & & & & & \bullet & \\
5 & & & & \bullet & e & \bullet & & \bullet & \circ & \bullet \\
6 & & \bullet & \circ & \circ & \circ & f & & \circ & \circ & \circ \\
7 & & & & \bullet & & & g & & \circ & \\
8 & \bullet & \bullet & \bullet & \circ & \circ & \circ & & h & \circ & \circ \\
9 & & & & & & & & & i & \bullet \\
10 & & & & & \bullet & \circ & \bullet & \circ & \circ & j
\end{array}$$

$$\mathcal{F}_1 = \begin{array}{c|cc|}
 & 1 & 3 \\
\hline
1 & a & \bullet \\
2 & \bullet & \\
3 & \bullet & \\
8 & \bullet &
\end{array}, \quad
\mathcal{U}_1 = \begin{array}{c|c|}
 & 3 \\
\hline
2 & \circ \\
3 & \circ \\
8 & \circ
\end{array}, \quad
\mathcal{F}_2 = \begin{array}{c|ccccc|}
 & 2 & 3 & 5 & 8 & 10 \\
\hline
2 & b & \circ & \bullet & \bullet & \bullet \\
6 & \bullet & & & & \\
8 & \bullet & & & &
\end{array}$$

$\mathcal{U}_1$ contains a contribution to row 2 but $\mathcal{F}_2$ cannot accommodate the $(3, 3)$ entry of $\mathcal{U}_1$

Thus conditions (a) and (b) are incompatible in general
{but both are valid when $A$ is structurally symmetric}

## *(b) BUT NOT (a) APPROACH* [Duff+Reid]

Assume every nonzero entry in $A + A^t$ is structurally nonzero in $A$ $\{$i.e., make $A$ structurally symmetric$\}$

Expanded frontal matrix $\widetilde{\mathcal{F}}_k$ has the symmetric row/column set $\{i \mid i \geq k \text{ and } \widetilde{\ell}_{ik} \neq 0\}$, where $A + A^t = \widetilde{L}\widetilde{L}^t$

The updates come from the children of $k$ in $T(A + A^t)$

**Improvement:** [Amestoy+Puglisi] Suppress the rows and columns of $\widetilde{\mathcal{F}}_k$ that are structurally zero, e.g.,

$$\widetilde{\mathcal{F}}_2 \equiv \begin{array}{c} \\ 2 \\ 3 \\ 6 \\ 8 \end{array} \!\!\begin{array}{c} \begin{array}{ccccc} 2 & 3 & 5 & 8 & 10 \end{array} \\ \begin{pmatrix} b & \circ & \bullet & \bullet & \bullet \\ & \circ & & & \\ \bullet & & & & \\ \bullet & \circ & & & \end{pmatrix} \end{array} = \begin{array}{c} \\ 2 \\ 6 \\ 8 \end{array} \!\!\begin{array}{c} \begin{array}{cccc} 2 & 5 & 8 & 10 \end{array} \\ \begin{pmatrix} b & \bullet & \bullet & \bullet \\ \bullet & & & \\ \bullet & & & \end{pmatrix} \end{array} \Updownarrow\!\!\Leftrightarrow \mathcal{U}_1$$

$\{$rows 5 and 10 and column 6 are structurally zero$\}$

Since the entries in row/column $k$ can still be structurally zero, some entries in corresponding columns/rows of the update can also be zero (possibly extra storage and work)

***(a) BUT NOT (b) APPROACH*** [Hadfield+Davis; Gupta]

Decompose $\mathcal{U}_1$ as

$$\mathcal{U}_1 \equiv \begin{matrix} & 3 \\ 2 & \circ \\ 3 & \circ \\ 8 & \circ \end{matrix} = \begin{matrix} & 3 \\ 2 & (\circ) \end{matrix} \Updownarrow\!\Leftrightarrow \begin{matrix} & 3 \\ 3 & \circ \\ 8 & \circ \end{matrix} \equiv \mathcal{U}_1^2 \Updownarrow\!\Leftrightarrow \mathcal{U}_1^3$$

and assemble row $\mathcal{U}_1^2$ into $\mathcal{F}_2$ and column $\mathcal{U}_1^3$ into $\mathcal{F}_3$

Decompose $\mathcal{U}_k$ by repeatedly removing the row/column of lowest index $i$ and assembling it into $\mathcal{F}_i$

**Improvement:** Stop when $\mathcal{F}_i$ can accommodate all entries in what remains of $\mathcal{U}_k$, i.e., (nesting property)

$$\text{struct}\,(\ell_{ik}, \ldots, \ell_{nk}) \subseteq \text{struct}\,(\ell_{ii}, \ldots, \ell_{ni})$$
$$\text{struct}\,(u_{ki}, \ldots, u_{kn}) \subseteq \text{struct}\,(u_{ii}, \ldots, u_{in})$$

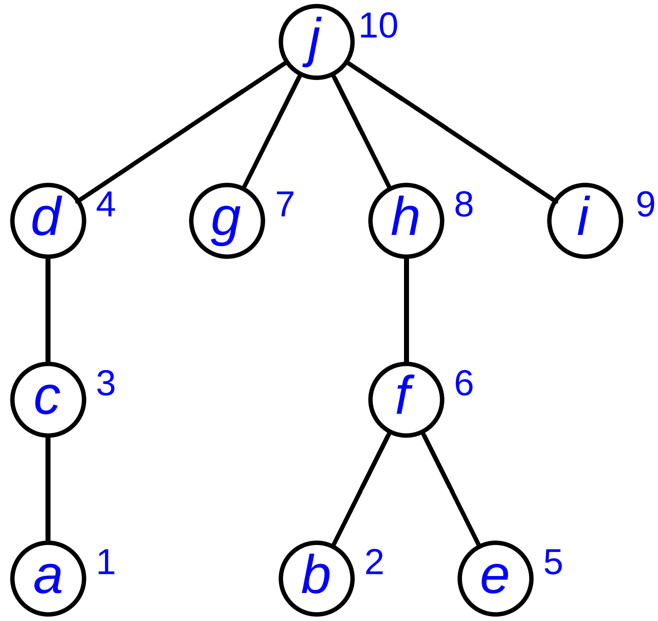
**Theorem:** [E+L]

$$\text{struct}\,(\ell_{pk}, \ldots, \ell_{nk}) \subseteq \text{struct}\,(\ell_{pp}, \ldots, \ell_{np})$$
$$\text{struct}\,(u_{kp}, \ldots, u_{kn}) \subseteq \text{struct}\,(u_{pp}, \ldots, u_{pn})$$

if $p = \rho(k) \equiv \min\{x \mid x > k \text{ and } x \overset{L}{\Longrightarrow} k \overset{U}{\Longrightarrow} x\}$.

# *EXAMPLE*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | *a* | | ● | | | | | | | |
| 2 | ● | *b* | ○ | | ● | | | ● | | ● |
| 3 | ● | | *c* | ● | | | | | | |
| 4 | | | ● | *d* | | | | | ● | |
| 5 | | | | ● | *e* | ● | | ● | ○ | ● |
| 6 | | ● | ○ | ○ | ○ | *f* | | ○ | ○ | ○ |
| 7 | | | | ● | | | *g* | | ○ | |
| 8 | ● | ● | ● | ○ | ○ | ○ | | *h* | ○ | ○ |
| 9 | | | | | | | | | *i* | ● |
| 10 | | | | | ● | ○ | ● | ○ | ○ | *j* |




---

$$\mathcal{F}_2 \equiv \begin{array}{c} \\ 2 \\ 6 \\ 8 \end{array} \begin{array}{c} \begin{array}{ccccc} 2 & 3 & 5 & 8 & 10 \end{array} \\ \begin{pmatrix} b & \circ & \bullet & \bullet & \bullet \\ \bullet & & & & \\ \bullet & & & & \end{pmatrix} \end{array} = \begin{array}{c} \\ 2 \\ 6 \\ 8 \end{array} \begin{array}{c} \begin{array}{cccc} 2 & 5 & 8 & 10 \end{array} \\ \begin{pmatrix} b & \bullet & \bullet & \bullet \\ \bullet & & & \\ \bullet & & & \end{pmatrix} \end{array} \mathbin{\updownarrow\!\!\!\leftrightarrow} \mathcal{U}_1^2$$

$$\mathcal{U}_2 \equiv \begin{array}{c} \\ 6 \\ 8 \end{array} \begin{array}{c} \begin{array}{cccc} 3 & 5 & 8 & 10 \end{array} \\ \begin{pmatrix} \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ \end{pmatrix} \end{array} = \begin{array}{c} \\ 6 \\ 8 \end{array} \begin{array}{c} 3 \\ \begin{pmatrix} \circ \\ \circ \end{pmatrix} \end{array} \mathbin{\updownarrow\!\!\!\leftrightarrow} \begin{array}{c} \\ 6 \\ 8 \end{array} \begin{array}{c} 5 \\ \begin{pmatrix} \circ \\ \circ \end{pmatrix} \end{array} \mathbin{\updownarrow\!\!\!\leftrightarrow} \begin{array}{c} \\ 6 \\ 8 \end{array} \begin{array}{c} \begin{array}{cc} 8 & 10 \end{array} \\ \begin{pmatrix} \circ & \circ \\ \circ & \circ \end{pmatrix} \end{array}$$

$$\equiv \mathcal{U}_2^3 \mathbin{\updownarrow\!\!\!\leftrightarrow} \mathcal{U}_2^5 \mathbin{\updownarrow\!\!\!\leftrightarrow} \mathcal{U}_2^{\rho(2)}$$

# *EXAMPLE (continued)*

$$\mathcal{F}_3 \equiv \begin{array}{r@{}l} & \begin{matrix}3 & 4\end{matrix} \\ \begin{matrix}3\\4\\6\\8\end{matrix} & \begin{pmatrix}\bullet & \bullet \\ \bullet & \\ \circ & \\ \bullet & \end{pmatrix}\end{array} = \begin{array}{r@{}l} & \begin{matrix}3 & 4\end{matrix} \\ \begin{matrix}3\\4\\8\end{matrix} & \begin{pmatrix} c & \bullet \\ \bullet & \\ \bullet & \end{pmatrix}\end{array} \oplus \mathcal{U}_1^3 \oplus \mathcal{U}_2^3$$

$$\mathcal{U}_3 = \begin{array}{r@{}l} & \begin{matrix}4\end{matrix} \\ \begin{matrix}4\\6\\8\end{matrix} & \begin{pmatrix}\circ \\ \circ \\ \circ\end{pmatrix}\end{array} \equiv \mathcal{U}_3^{\rho(3)}$$

---

$$\mathcal{F}_4 \equiv \begin{array}{r@{}l} & \begin{matrix}4 & 9\end{matrix} \\ \begin{matrix}4\\5\\6\\7\\8\end{matrix} & \begin{pmatrix}\bullet & \bullet \\ \bullet & \\ \circ & \\ \bullet & \\ \circ & \end{pmatrix}\end{array} = \begin{array}{r@{}l} & \begin{matrix}4 & 9\end{matrix} \\ \begin{matrix}4\\5\\7\end{matrix} & \begin{pmatrix} d & \bullet \\ \bullet & \\ \bullet & \end{pmatrix}\end{array} \oplus \mathcal{U}_3^4$$

$$\mathcal{U}_4 \equiv \begin{array}{r@{}l} & \begin{matrix}9\end{matrix} \\ \begin{matrix}5\\6\\7\\8\end{matrix} & \begin{pmatrix}\circ \\ \circ \\ \circ \\ \circ\end{pmatrix}\end{array} = \begin{array}{r@{}l} & 9 \\ 5 & (\circ)\end{array} \oplus \begin{array}{r@{}l} & 9 \\ 6 & (\circ)\end{array} \oplus \begin{array}{r@{}l} & 9 \\ 7 & (\circ)\end{array} \oplus \begin{array}{r@{}l} & 9 \\ 8 & (\circ)\end{array}$$

$$\equiv \mathcal{U}_4^5 \oplus \mathcal{U}_4^6 \oplus \mathcal{U}_4^7 \oplus \mathcal{U}_4^8$$

$\{\mathcal{U}_4^{\rho(4)}$ is the null matrix$\}$

# EXAMPLE (continued)

$$\mathcal{F}_5 \equiv \begin{array}{c} \\ 5 \\ 6 \\ 8 \\ 10 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 5 & 6 & 8 & 9 & 10 \end{array} \\ \left(\begin{array}{ccccc} e & \bullet & \bullet & \circ & \bullet \\ \circ & & & & \\ \circ & & & & \\ \bullet & & & & \end{array}\right) \end{array} = \begin{array}{c} \\ 5 \\ 10 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 5 & 6 & 8 & 10 \end{array} \\ \left(\begin{array}{cccc} e & \bullet & \bullet & \bullet \\ \bullet & & & \end{array}\right) \end{array} \updownarrow\!\!\!\!\leftrightarrow \mathcal{U}_2^5 \updownarrow\!\!\!\!\leftrightarrow \mathcal{U}_4^5$$

$$\mathcal{U}_5 = \begin{array}{c} \\ 6 \\ 8 \\ 10 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 6 & 8 & 9 & 10 \end{array} \\ \left(\begin{array}{cccc} \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ \end{array}\right) \end{array} \equiv \mathcal{U}_5^{\rho(5)}$$

---

$$\mathcal{F}_6 \equiv \begin{array}{c} \\ 6 \\ 8 \\ 10 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 6 & 8 & 9 & 10 \end{array} \\ \left(\begin{array}{cccc} f & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ \end{array}\right) \end{array} = \begin{array}{c} \\ 6 \end{array}\!\!\begin{array}{c} 6 \\ \left( f \right) \end{array} \updownarrow\!\!\!\!\leftrightarrow \mathcal{U}_2^6 \updownarrow\!\!\!\!\leftrightarrow \mathcal{U}_4^6 \updownarrow\!\!\!\!\leftrightarrow \mathcal{U}_5^6$$

$$\mathcal{U}_6 = \begin{array}{c} \\ 8 \\ 10 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 8 & 9 & 10 \end{array} \\ \left(\begin{array}{ccc} \circ & \circ & \circ \\ \circ & \circ & \circ \end{array}\right) \end{array} \equiv \mathcal{U}_6^{\rho(6)}$$

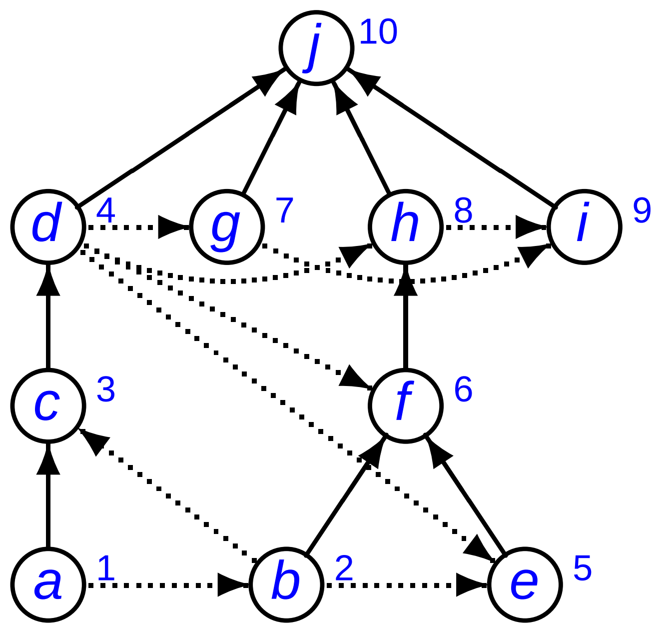
{nonzeros in $\mathcal{U}_2^6$ and $\mathcal{U}_5^6$ are not limited to the leftmost column and topmost row of $\mathcal{F}_6$}

## *DATAFLOW GRAPH* $D(A)$

Each vertex $k$ corresponds to a frontal matrix / update

Each solid edge is a tree edge and corresponds to a $\{$possibly null$\}$ update $\mathcal{U}_k^{\rho(k)}$ to frontal matrix $\mathcal{F}_{\rho(k)}$

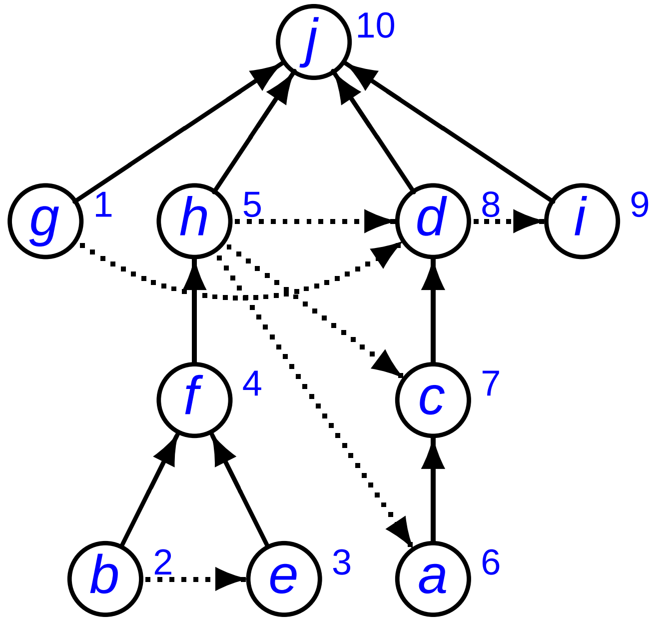
Each dotted cross edge corresponds to a nonnull update $\mathcal{U}_k^s$ to frontal matrix $\mathcal{F}_s$ with $k < s < \rho(k)$




The dataflow graph for the "(b) but not (a)" approach is $T(A + A^t)$ $\{$a chain for the example above$\}$

**Theorem:** If $x \longmapsto y$ is an edge in $D(A)$, then $y$ is an ancestor of $x$ in $T(A + A^t)$.
$\{$possibly more exploitable parallelism$\}$

## *UPPER BBT POSTORDERING REVISITED*




**Theorem:** When $A$ is upper BBT ordered,

$$\rho(k) = \min\{x \mid x > k \text{ and } x \stackrel{L}{\longmapsto} k \stackrel{U}{\Longrightarrow} x\};$$

the decomposition of $\mathcal{U}_k$ is by columns; and all cross edges lead to older siblings and their descendants.

**Advantages**

- simplifies the implementation
- enables a stack-based approach that reduces the storage required for updates

**Disadvantages**

- increased work and fill?

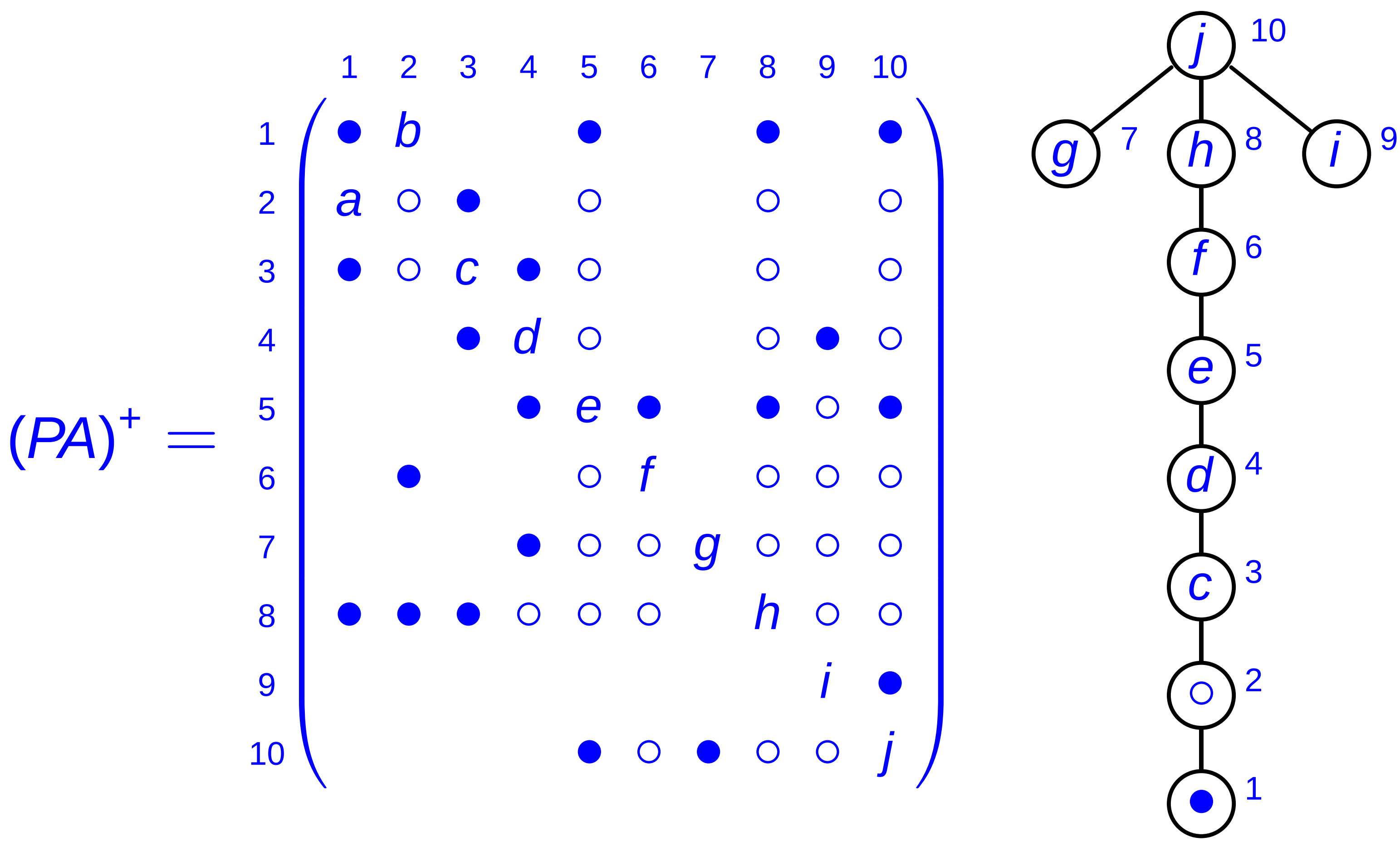
# *PIVOTING FOR STABILITY*

*SuperLU* **Approach** $\{$= No Pivoting$\}$ [Demmel+Li]:

Replace any pivot of magnitude less than $\sqrt{\epsilon}\,\|A\|_1$ by $\sqrt{\epsilon}\,\|A\|_1$ and use an extra stage of iterative refinement to compensate for the perturbation

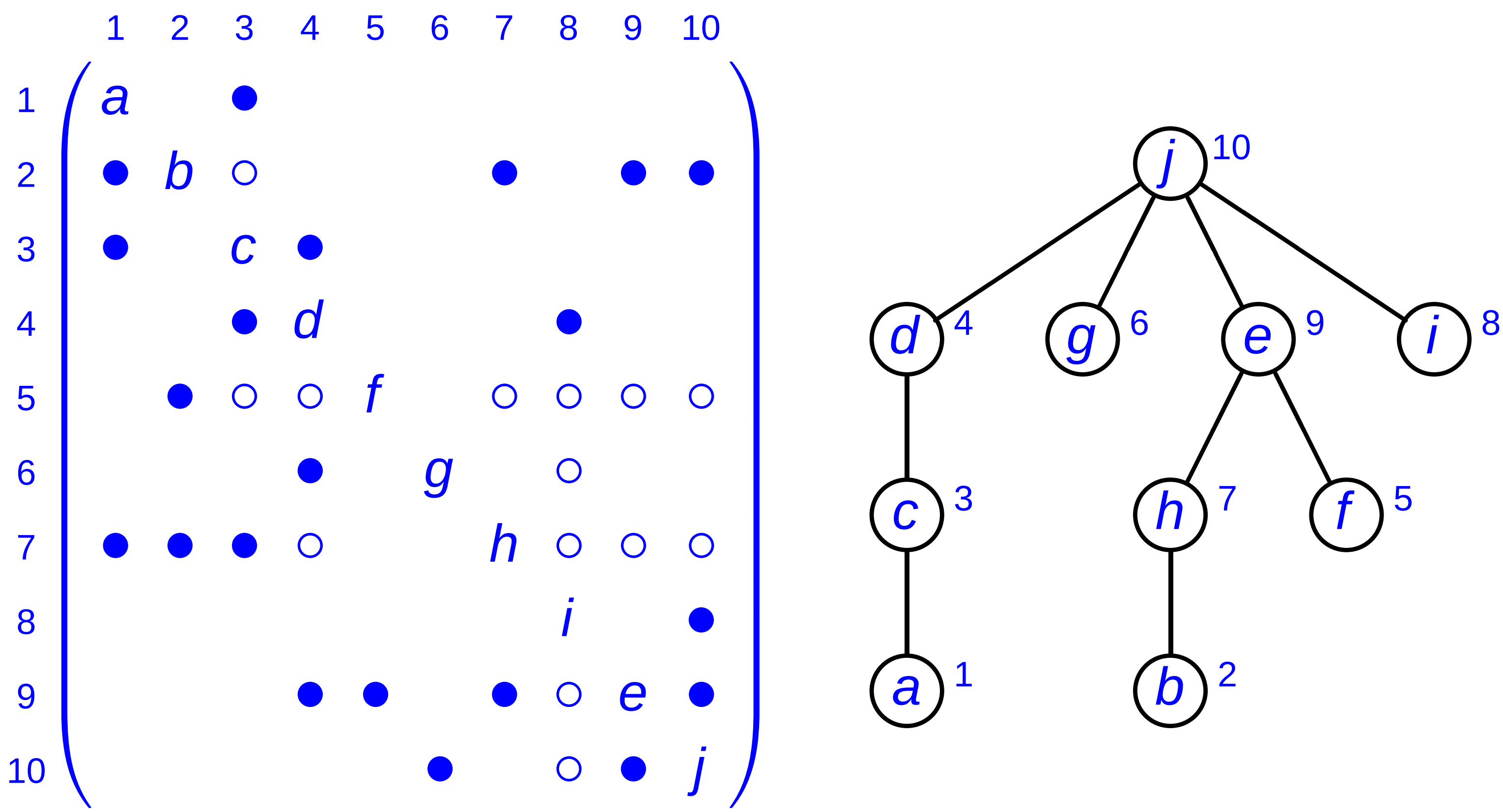
**Unrestricted Pivoting**

A single row or column interchange can completely destroy the structure of $T(A)$ and $D(A)$



# *PIVOTING FOR STABILITY (continued)*

**Delayed elimination**: Symmetrically permute $A$ so that row/column $k$ immediately follows row/column $m$; e.g., delaying $e$ after $i$,




**Theorem:** Let

$$S = \{k\} \cup \{x \mid k < x \leq m \text{ and } x \text{ ancestor of } k\}.$$

If $x \notin S$ and $\rho(x) \notin S$, then $x$ has the same parent in $T(PAP^t)$.

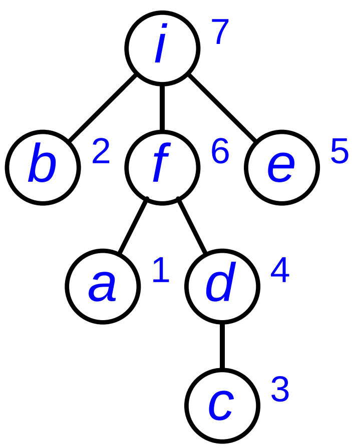
$\{$Local effect on $T(A)$; insufficient alone in general$\}$

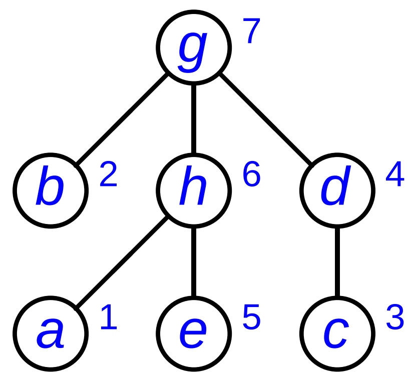
## *PIVOTING FOR STABILITY (continued)*

**Pivoting within diagonal blocks** (e.g., supernodes):

Exchange rows within a diagonal block; e.g.,

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} a & & & & & \bullet & \\ & b & & & & & \bullet \\ & & c & \bullet & & & \\ & & \bullet & d & & \bullet & \\ & & & & e & \bullet & \bullet \\ \bullet & & & \bullet & & f & g \\ \bullet & \bullet & & & \bullet & h & i \end{pmatrix} \end{array}, \quad QA = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} a & & & & & \bullet & \\ & b & & & & & \bullet \\ & & c & \bullet & & & \\ & & \bullet & d & & \bullet & \\ & & & & e & \bullet & \bullet \\ \bullet & \bullet & & & \bullet & h & i \\ \bullet & & & \bullet & & f & g \end{pmatrix} \end{array}$$




**Theorem:** Let $S = \{k, k+1, \ldots, m\}$ and let $Q$ be the permutation matrix that exchanges rows $k$ and $m$. If $a_{ij} \neq 0$ for $i, j \in S$, then $T(QA)$ is the same as $T(A)$, except that the parent of each child $s \notin S$ in $T(A)$ of a vertex in $S$ can now be any vertex in $S$.

$\{$Local effect on $T(A)$; insufficient alone in general$\}$

## *PIVOTING FOR STABILITY (continued)*

**Combined Approach** [Duff+Reid]

If pivot $k$ is unacceptable, delay that step of elimination until after the parent $p = \rho(k)$ of $k$ is eliminated $\{$there is no change to the pivot value until then$\}$

If pivot $p$ is also unacceptable, then treat $\{p, k\}$ as a "supernode" and pivot within the supernodal block $\{$any extra fill is restricted to rows/columns $p$ and $k\}$

The effect on the elimination tree of each delayed elimination or row exchange is local

---

## *PIVOTING IN UNSYMMETRIC MULTIFRONTAL*

**Problem:** $\mathcal{F}_k$ has already been assembled when pivot $k$ is evaluated and deemed unacceptable so the dataflow graph $D(PAP^t)$ or $D(QA)$ of the permuted matrix does not capture all flows

## *PIVOTING IN MULTIFRONTAL (continued)*

**Solution:** Keep the original dataflow graph $D(A)$ but make $k$ an assembly-only vertex at which we

- Assemble all updates contributing to row/column $k$ into frontal matrix $\mathcal{F}_k$
- Permute row $k$ immediately after row $p = \rho(k)$ $\{p \overset{L}{\longmapsto} k$ since $A$ is upper BBT ordered$\}$
- Permute column $k$ either immediately after column $p$ (if present) or to where column $p$ would be (otherwise)
- Treat the entire $\mathcal{F}_k$ as the update; that is, decompose it in the usual way with the pieces sent along edges in the dataflow graph to vertices between $k$ and $p$ in the original order

$\{$approach works when $A$ is upper BBT ordered$\}$

## *PIVOTING IN MULTIFRONTAL (continued)*

$$
(PAP^t)^+ = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8\\9\\10 \end{array}
\begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\
\left(\begin{array}{cccccccccc}
g & & & & & & & \bullet & & \\
 & b & \bullet & & \bullet & \bullet & & & & \bullet \\
 & & e & \bullet & \bullet & & & \bullet & & \bullet \\
 & \bullet & \circ & f & \circ & \circ & & \circ & & \circ \\
 & \bullet & \circ & \circ & h & \bullet & \bullet & \circ & & \circ \\
 & & & & & a & \bullet & & & \\
 & & & & & \bullet & c & \bullet & & \\
 & & & & & & \bullet & d & \bullet & \\
 & & & & & & & & i & \bullet \\
\bullet & & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & j
\end{array}\right) \end{array}
$$

Delaying elimination of vertex 5 until after its parent 10,

$$
\mathcal{U}_5 \equiv \begin{array}{c} \\ 10 \\ 5 \end{array}
\begin{array}{c} \begin{array}{ccccc} 6 & 7 & 8 & 10 & 5 \end{array} \\
\left(\begin{array}{ccccc} \circ & \cdot & \circ & \circ & \circ \\ \bullet & \bullet & \circ & \circ & h \end{array}\right) \end{array}
= \mathcal{U}_5^6 \Updownarrow\!\!\Leftrightarrow \mathcal{U}_5^7 \Updownarrow\!\!\Leftrightarrow \mathcal{U}_5^8 \Updownarrow\!\!\Leftrightarrow \mathcal{U}_5^{10}
$$

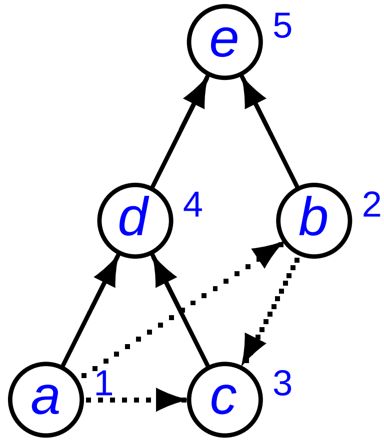
where $\cdot$ denotes an entry that would have filled in and

$$
\mathcal{U}_5^s = \begin{array}{c} \\ 10 \\ 5 \end{array}
\begin{array}{c} s \\ \left(\begin{array}{c} \circ \\ \bullet \end{array}\right) \end{array}, \quad 6 \le s \le 8; \qquad
\mathcal{U}_5^{10} = \begin{array}{c} \\ 10 \\ 5 \end{array}
\begin{array}{c} \begin{array}{cc} 10 & 5 \end{array} \\ \left(\begin{array}{cc} \circ & \circ \\ \circ & h \end{array}\right) \end{array}
$$

The recipient of $\mathcal{U}_5^s$ expects an update to row 10; and vertex 10 (= supernode $\{10, 5\}$) can accommodate an update to column 5

# *WHAT IF $A$ IS NOT BBT ORDERED?*

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & a & \bullet & & \bullet & \\
2 & & b & & & \bullet \\
3 & \bullet & \circ & c & \circ & \circ \\
4 & & & \bullet & d & \circ \\
5 & & & & \bullet & e
\end{array}
$$




$$
\begin{array}{c|ccccc}
 & 2 & 3 & 4 & 1 & 5 \\
\hline
2 & b & & & & \bullet \\
3 & & c & & \bullet & \\
4 & & \bullet & d & \circ & \\
1 & \bullet & & \bullet & a & \circ \\
5 & & & \bullet & \circ & e
\end{array}
$$

Delaying elimination of vertex 1 until after its parent 4,

$$
\mathcal{U}_1 \equiv \begin{array}{c} \\ 3 \\ 1 \end{array}\!\begin{array}{ccc} 2 & 4 & 1 \\ \cdot & \cdot & \bullet \\ \bullet & \bullet & a \end{array} = \begin{array}{c} \\ 3 \\ 1 \end{array}\!\begin{array}{c} 2 \\ \cdot \\ \bullet \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} {}_3\!\begin{array}{cc} 4 & 1 \\ \cdot & \bullet \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} {}_1\!\begin{array}{cc} 4 & 1 \\ \bullet & a \end{array}
$$

$$
\equiv \mathcal{U}_1^2 \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \mathcal{U}_1^3 \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \mathcal{U}_1^4
$$

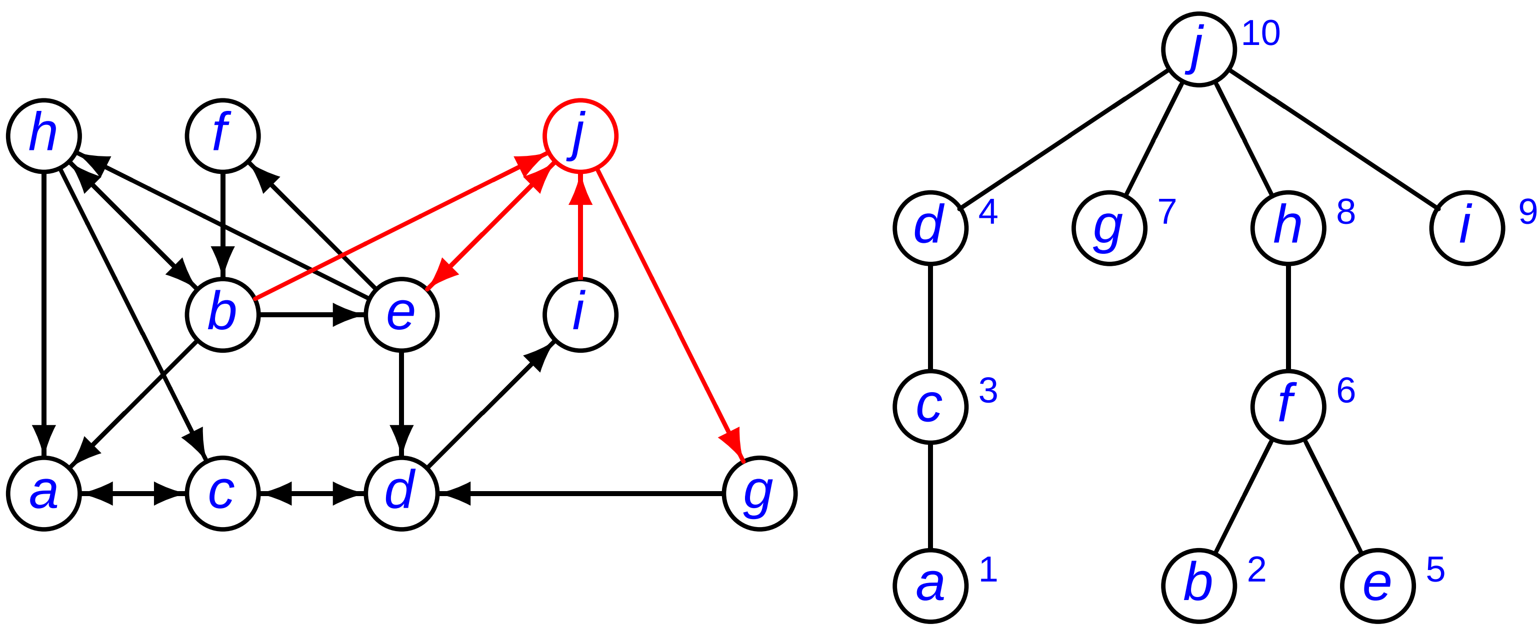
$$
\mathcal{F}_2 \equiv \begin{array}{c} \\ 2 \\ 3 \\ 1 \end{array}\!\begin{array}{cc} 2 & 5 \\ b & \bullet \\ \cdot & \\ \bullet & \end{array} = {}_2\!\begin{array}{cc} 2 & 5 \\ b & \bullet \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \mathcal{U}_1^2
$$

$$
\mathcal{U}_2 \equiv \begin{array}{c} \\ 3 \\ 1 \end{array}\!\begin{array}{c} 5 \\ \circ \\ \circ \end{array} = {}_3\!\begin{array}{c} 5 \\ \circ \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} {}_1\!\begin{array}{c} 5 \\ \bullet \end{array} \equiv \mathcal{U}_2^3 \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \mathcal{U}_2^4,
$$

There is no edge from vertex 2 to vertex 4 (into which vertex 1 was coalesced) over which to send $\mathcal{U}_2^4$

# *FINDING THE ELIMINATION TREE $T(A)$*

**Theorem:** $k$ is the parent of $c$ if and only if $k$ is the first vertex after $c$ such that $k$ and $c$ belong to the same strongly connected component of the subgraph $G_k(A)$ of $G(A)$ induced by $\{1, 2, \ldots, k\}$.




**Algorithm** *eTree* $\{\text{computes } \rho(k) \equiv fpnz(k)\}$

**for** $k = 1$ **to** $n$ **do**

Find the component $\mathcal{X}$ of $G_k(A)$ that contains $k$

**for each** $x \in \mathcal{X} \setminus \{k\}$ **do**

**if** $fpnz(x) = \infty$ **then** $fpnz(x) = k$

**end for**

$fpnz(k) = \infty$

**end for**

## *FINDING THE ELIMINATION TREE (continued)*

Time = $O(ne)$, where $e$ = the number of edges in $G(A)$

{Finding the strongly connected components of $G_k(A)$ takes time $O(e_k)$, where $e_k$ is the number of vertices and edges in $G_k(A)$ reachable from $k$ [Tarjan; Gabow; . . . ]}

---

## *REDUCING THE EFFECTIVE SIZE*

The strongly connected components of $G_k(A)$ induce an acyclic quotient graph $G(Q_k)$ that succinctly represents its connectivity [Pagallo+Maulino]:

> Let $x$ and $y$ belong to components $\mathcal{X}$ and $\mathcal{Y}$;
> then $x \Rightarrow y$ if and only if $\mathcal{X} = \mathcal{Y}$ or $\mathcal{X} \Rightarrow \mathcal{Y}$

**Theorem:** If $\mathcal{X}$ is a component of $G_k(A)$, and $m$ is the highest-numbered vertex in $\mathcal{X}$, then $\mathcal{X} = \mathcal{T}[m]$.

## *REDUCING THE EFFECTIVE SIZE (continued)*

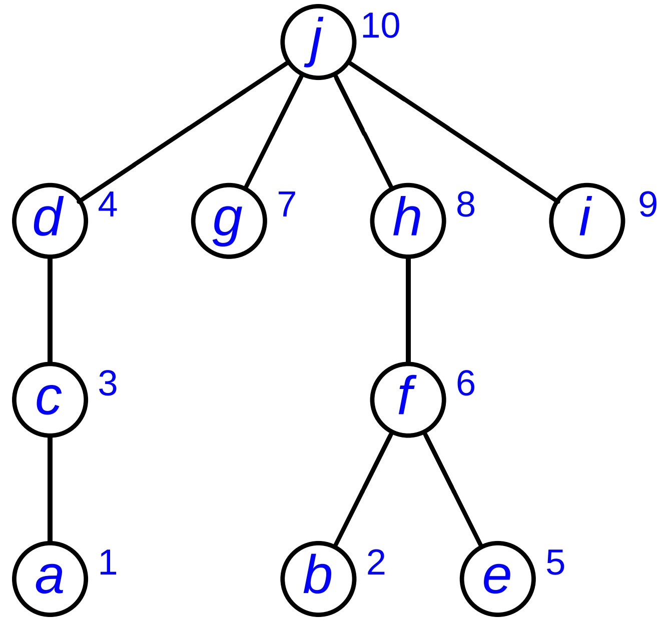
$G_k(A)$ can be obtained from $G_{k-1}(A)$ by

- adding vertex $k$;
- adding all edges $u \overset{A}{\longmapsto} k$ with $u < k$
- adding all edges $k \overset{A}{\longmapsto} v$ with $v < k$

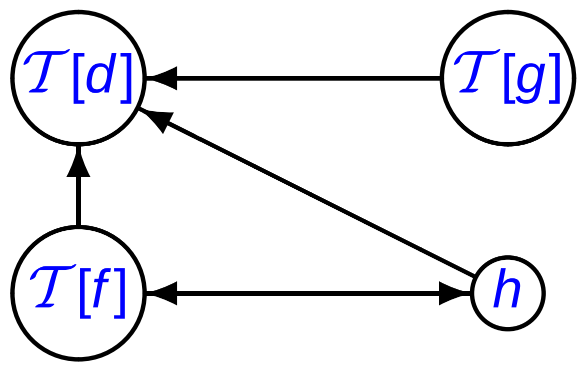
$G(Q_k)$ can be obtained from $G(Q_{k-1})$ in two steps:

1. Form intermediate graph $G(Q'_k)$ from $G(Q_{k-1})$ by
   - adding vertex $k$;
   - for each vertex $\mathcal{T}[u]$, adding edge $\mathcal{T}[u] \mapsto k$ if $x \overset{A}{\longmapsto} k$ for some $x \in \mathcal{T}[u]$; and
   - for each vertex $\mathcal{T}[v]$, adding edge $k \mapsto \mathcal{T}[v]$ if $k \overset{A}{\longmapsto} y$ for some $y \in \mathcal{T}[v]$
2. Form $G(Q_k)$ from $G(Q'_k)$ by coalescing $k$ and all vertices in the strongly connected component of $G(Q'_k)$ that contains it into the new vertex $\mathcal{T}[k]$

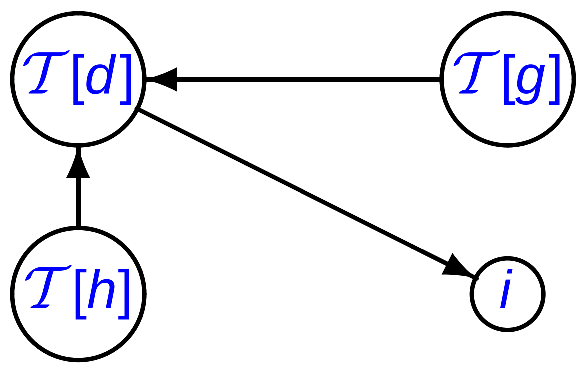
$$
\begin{array}{c|cccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & a & & \bullet & & & & & & & \\
2 & \bullet & b & & & \bullet & & & \bullet & & \bullet \\
3 & \bullet & & c & \bullet & & & & & & \\
4 & & & \bullet & d & & & & & \bullet & \\
5 & & & & \bullet & e & \bullet & & \bullet & & \bullet \\
6 & & \bullet & & & & f & & & & \\
7 & & & & \bullet & & & g & & & \\
8 & \bullet & \bullet & \bullet & & & & & h & & \\
9 & & & & & & & & & i & \bullet \\
10 & & & & & \bullet & & \bullet & & & j
\end{array}
$$



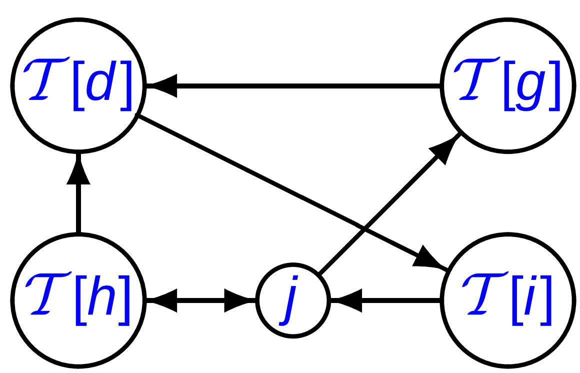

$$
Q'_8 = \left(\begin{array}{ccc|c}
\mathcal{T}[d] & & & \\
\bullet & \mathcal{T}[f] & & \bullet \\
\bullet & & \mathcal{T}[g] & \\
\hline
\bullet & \bullet & & h
\end{array}\right)
$$




$$
Q'_9 = \left(\begin{array}{ccc|c}
\mathcal{T}[d] & & & \bullet \\
\bullet & \mathcal{T}[g] & & \\
\bullet & & \mathcal{T}[h] & \\
\hline
 & & & i
\end{array}\right)
$$




$$
Q'_{10} = \left(\begin{array}{cccc|c}
\mathcal{T}[d] & & & \bullet & \\
\bullet & \mathcal{T}[g] & & & \\
\bullet & & \mathcal{T}[h] & & \bullet \\
 & & & \mathcal{T}[i] & \bullet \\
\hline
 & \bullet & \bullet & & j
\end{array}\right)
$$



## *FINDING THE ELIMINATION TREE (continued)*

**Theorem:** If $\mathcal{T}[c]$ is a vertex of $G(Q_k')$ that is coalesced into $\mathcal{T}[k]$, then $c$ is a child of $k$.

**Algorithm** *eTreeQ* {computes $\rho(k) \equiv \mathit{fpnz}(k)$}

$G(Q_0) =$ empty graph

**for** $k = 1$ **to** $n$ **do**

Create $G(Q_k')$ from $G(Q_{k-1})$ by adding $k$ and its incident edges

Find the component $\mathcal{X}$ of $G(Q_k')$ that contains $k$

**for each** $\mathcal{T}[c] \in \mathcal{X} \setminus \{k\}$ **do**

$\mathit{fpnz}(c) = k$

**end for**

Create $G(Q_k)$ from $G(Q_k')$ by coalescing the vertices in $\mathcal{X}$ into the new vertex $\mathcal{T}[k]$

$\mathit{fpnz}(k) = \infty$

**end for**

{We can take advantage of the fact that $G(Q_{k-1})$ is an acyclic subgraph of $G(Q_k')$ to find the component $\mathcal{X}$ of $G(Q_k')$ that contains $k$}

## *FINDING THE ELIMINATION TREE (continued)*

{Initialization prior to first call to *scc* ($k$)}

**for each** $v$ in $G(A)$ **do**

  $visited[v] = inSC[v] = 0$

**end for**

**Algorithm** *scc* ($k$)

  $inSC[k] = k$

  $SC = \{k\}$

  dfs($k$)

  **return** $SC$ {= component of $G(Q_k')$ containing $k$}

  **procedure** *dfs* ($v$)

    $visited[v] = k$

    **for each** $w$ adjacent to $v$ in $G(Q_k')$ **do**

      **if** $visited[w] \neq k$ **then** $dfs(w)$

      **if** $inSC[w] = k$ **then** $inSC[v] = k$

      {$inSC[w] = k$ if and only if $w \Rightarrow k$ in $G(Q_k')$}

    **end for**

    **if** $inSC[v] = k$ **then** $SC = SC \cup \{v\}$

  **end procedure**

# *FINDING AN UPPER BBT ORDERING (continued)*

| Name | $n$ | $\frac{nz(A)}{n}$ | $symm$ | $t_s(A)$ (ms) | $\frac{t_t(A)}{t_s(A)}$ |
|---|---|---|---|---|---|
| Averous/epb1 | 14734 | 6.45 | 72.94 | 15.7 | .47 |
| Bai/rw5151 | 5146 | 3.92 | 0.01 | 5.5 | .49 |
| Goodwin/goodwin | 7319 | 44.37 | 26.68 | 37.8 | .48 |
| Graham/graham1 | 8398 | 36.59 | 29.47 | 38.3 | .49 |
| Grund/bayer02 | 11710 | 4.63 | 7.44 | 7.2 | 1.24 |
| Grund/bayer10 | 10803 | 5.76 | 8.86 | 6.2 | 1.24 |
| HB/gemat11 | 4578 | 6.86 | 70.96 | 1.4 | 1.50 |
| Hamrle/Hamrle2 | 5952 | 3.72 | 12.18 | 1.5 | 1.87 |
| Hohn/fd12 | 6787 | 3.90 | 2.98 | 5.7 | .74 |
| Hohn/fd15 | 10645 | 3.92 | 3.03 | 10.5 | .67 |
| Hohn/fd18 | 15367 | 3.94 | 2.51 | 17.2 | .66 |
| Hohn/sinc12 | 6974 | 38.43 | 22.65 | 262.8 | .07 |
| Hollinger/g7jac040 | 11194 | 9.36 | 8.81 | 63.0 | .85 |
| Lucifora/cell1 | 7055 | 4.26 | 79.63 | 4.1 | .71 |
| Lucifora/cell2 | 7055 | 4.26 | 79.63 | 4.3 | .67 |
| Nasa/barth | 5711 | 3.49 | 13.66 | 1.8 | 1.56 |
| Nasa/barth4 | 5826 | 3.90 | 11.11 | 2.4 | 1.58 |
| Nasa/barth5 | 12960 | 3.93 | 7.61 | 6.2 | 1.90 |
| Shen/e40r0100 | 17281 | 32.03 | 88.69 | 39.3 | .68 |
| Shen/shermanACd | 6042 | 6.99 | 61.65 | 3.2 | .88 |
| TOKAMAK/utm5940 | 5794 | 14.35 | 28.94 | 17.0 | .31 |

## *FINDING AN UPPER BBT ORDERING*

Let $c_1, \ldots, c_t$ be the children of $k$ in $T(A)$

The subtrees $\mathcal{T}[c_1], \ldots, \mathcal{T}[c_t]$ are the vertices of $G(Q_{k-1})$ that are coalesced into $\mathcal{T}[k]$ during the formation of $G(Q_k)$

For an upper BBT postordering $c_i$ must be numbered before $c_j$ if $x \longmapsto y$ for some $x \in \mathcal{T}[c_i]$ and $y \in \mathcal{T}[c_j]$; i.e., $\mathcal{T}[c_i] \longmapsto \mathcal{T}[c_j]$

In Algorithm *scc*, vertices are added to $SC$ in the following order:

- Add vertex $k$ first
- If $\mathcal{T}[c_i] \longmapsto \mathcal{T}[c_j]$ in $G(Q_k')$ and $\mathcal{T}[c_j] \in SC$, then add $\mathcal{T}[c_i]$ to $SC$ $\{$i.e., after $\mathcal{T}[c_j]\}$

That is, ignoring vertex $k$, the reverse of this order satisfies the condition for an upper BBT postordering

$\{$The additional runtime is negligible$\}$

## *OTHER APPLICATIONS*

**Diagonal Markowitz Ordering** [Amestoy+Li+Ng]

The algorithm for finding $T(A)$ can be extended to maintain a compact and efficient representation of the nonzero structure of the reduced matrix, which can be used to select diagonal pivots that locally minimize the work

**Finding Supernodes**

Supernodes (i.e., dense diagonal blocks) form chains in $T(A)$, which suggests both how to find them (via postordering) and how to relax their definition (cf. [Ashcraft+Grimes])

**Symbolic Factorization**

Using $\rho(k)$ and the nesting property, more edges can be pruned, making the algorithm more efficient {path-symmetric symbolic factorization [E+L]}

## *SUMMARY*

- The elimination tree $T(A)$ of an unsymmetric matrix
- Properties of upper BBT postorderings of $T(A)$
- A dataflow graph for unsymmetric multifrontal
- The impact of pivoting for stability
- Algorithms for finding $T(A)$
- Other applications

---

## *CONCLUSION*

$$\rho(k) \equiv \min\{x \mid x > k \text{ and } x \overset{L}{\Longrightarrow} k \overset{U}{\Longrightarrow} x\}$$

is the right generalization of elimination tree to unsymmetric matrices

## ***REFERENCES***

*The theory of elimination trees for sparse unsymmetric matrices,* SIAM Journal on Matrix Analysis and Applications, 26(3):686–705, 2005

*A tree-based dataflow model for the unsymmetric multifrontal method* (to appear in Electronic Transactions on Numerical Analysis)

*Algorithmic aspects of elimination trees for sparse unsymmetric matrices* (in progress)