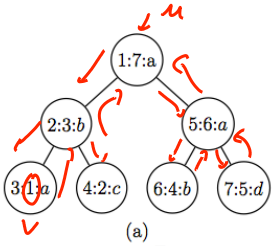
# XASR: eXtented Access Support Relations




(a)

| R | pre | post | parent_pre | lab |
|---|---|---|---|---|
| | 1 | 7 | ⊥ | a |
| | 2 | 3 | 1 | b |
| | 3 | 1 | 2 | a |
| | 4 | 2 | 2 | c |
| | 5 | 6 | 1 | a |
| | 6 | 4 | 5 | b |
| | 7 | 5 | 5 | d |

(b)

Figure 2: Tree (a) and XASR (b).

**Example 2.1** The tree shown in Figure 2 (a) can be represented by the XASR of Figure 2 (b). We can define the descendant axis as an SQL view

```
CREATE VIEW descendant AS
SELECT r1.pre, r2.pre FROM R r1, R r2
WHERE r1.pre < r2.pre AND r2.post < r1.post;
```

and the child axis as

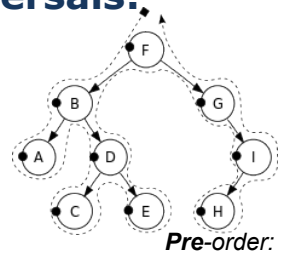
```
CREATE VIEW child AS
SELECT parent_pre, pre FROM R
WHERE parent_pre is not NULL;
```

[3] Processing queries on tree-structured data efficiently, Koch, C., PODS 2006.

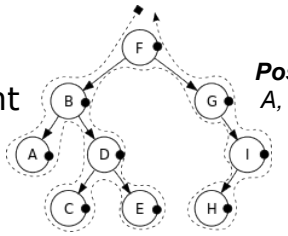


# Depth-First Tree Traversals:

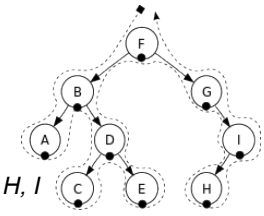
- **Pre**-order:
  - Visit the root
  - Traverse left child
  - Traverse right child
- **Post**-order
  - Traverse left
  - Traverse right
  - Visit root
- **In**-order
  - Traverse left
  - Visit root
  - Traverse right




***Pre**-order:*
*F, B, A, D, C, E, G, I, H*

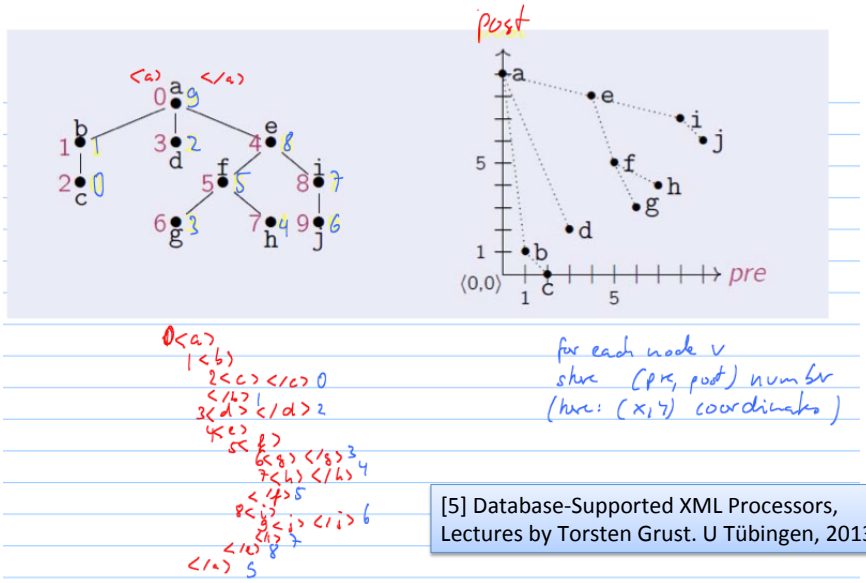



***Post**-order:*
*A, C, E, D, B, H, I, G, F*




***In**-order:*
*A, B, C, D, E, F, G, H, I*

# Pre- and Post-Order Traversal Numbers



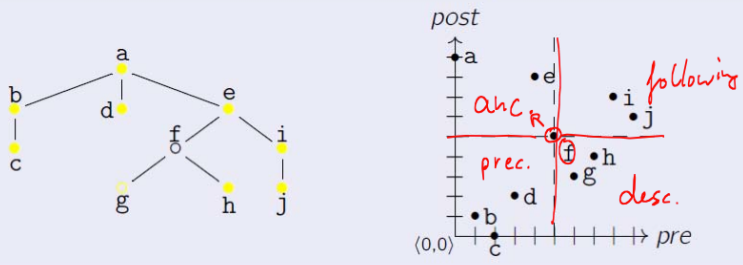

for each node v
store (pre, post) number
(here: (x,y) coordinates)

[5] Database-Supported XML Processors, Lectures by Torsten Grust. U Tübingen, 2013



## Know your descendants, ancestors, following and preceding nodes via their pre- and post-order coordinates …

### Plane partitions ≡ XPath axes, ○ is arbitrary!




### Pre/post plane regions ≡ major XPath axes

The **major XPath axes** `descendant`, `ancestor`, `following`, `preceding` correspond to rectangular **pre/post plane windows.**

[5] Database-Supported XML Processors, Lectures by Torsten Grust. U Tübingen, 2013

## Example

[5] Database-Supported XML Processors, Lectures by Torsten Grust. U Tübingen, 2013

```
(e,f)/descendant::node()
```

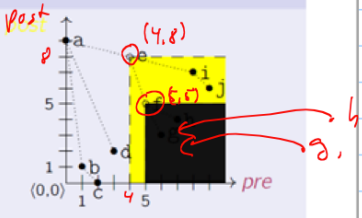
Context & frag. encodings

context

| pre | post | ··· |
|---|---|---|
| 5 | 5 | |
| 4 | 8 | |

accel

| pre | post | ··· |
|---|---|---|
| 0 | 9 | |
| 1 | 1 | |
| 2 | 0 | |
| 3 | 2 | |
| 4 | 8 | |
| 5 | 5 | |
| 6 | 3 | |
| 7 | 4 | |
| 8 | 7 | |
| 9 | 6 | |




SQL query with expanded *window*() predicate

```
SELECT    DISTINCT v1.*
FROM      context v, accel v1
WHERE     v1.pre > v.pre AND v1.post < v.post
ORDER BY  v1.pre
```



## References

- Lots of basic XPath material out there …
  - Google, e.g., "XPath tutorial"
- Some (quite) advanced resources:

[1] Monadic queries over tree-structured data, Gottlob, G. and Koch, C., Logic in Computer Science (LICS), 2002.

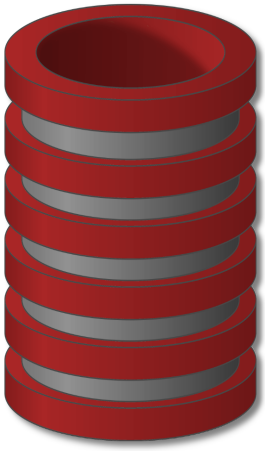
[2] Efficient Algorithms for Processing XPath Queries, Georg Gottlob, Christoph Koch, and Reinhard Pichler, ACM TODS, 2005

[3] Processing queries on tree-structured data efficiently, Koch, C., PODS 2006.

[4] An XML Toolkit for Light-weight XML Stream Processing, Yours Truly (TJ Green) et al., 2003, http://homes.cs.washington.edu/~suciu/XMLTK/

**[5] Database-Supported XML Processors, Torsten Grust, U Tübingen, Winter 2012/13**

http://db.inf.uni-tuebingen.de/teaching/ws1213/dbxml



# Querying XML

## XQuery

Db-class.org, Jennifer Widom, Stanford



**Querying XML**

Not nearly as mature as Querying Relational

- Newer
- No underlying algebra

Sequence of development

1. XPath ✓
2. ~~XSLT~~
3. XQuery ✓

//a/b/c ..

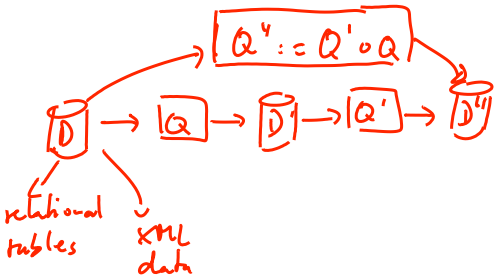
" for $a in ..
  for $b in ..
    for $c in ..
      ...

Db-class.org, Jennifer Widom, Stanford

XQuery

## XQuery

- Expression language (compositional)
- Each expression operates on & returns *sequence of elements*
- XPath is one type of expression






XQuery

## XQuery: FLWOR expression

```
For $var in expr
Let $var := expr
Where condition
Order By expr
Return expr
```

- All except **Return** are optional
- **For** and **Let** can be repeated and interleaved

XQuery

## Mixing queries and XML

```
<Result> { …query goes here… } </Result>
```

**Demo:** XQuery examples over bookstore data

Db-class.org, Jennifer Widom, Stanford

https://class.stanford.edu/courses/Engineering/db/2014_1/courseware/ch-querying_xml/seq-vid-xquery_demo/