

partition

- children x
- disc
- parent *
- ancestors x x
- self x
- predecessor x
- siblings x
- following-
sibling ...

path/following::x =

path/ancestor-or-self::x /
following-sibling::x /
descendant-or-self::x

Dez 5-10:08

n

n+1

n+2

n+?

n+?-1

preorder

postorder

inorder

traverse: self
then, recursively its children

(5795)

(5795)

Dez 5-10:26

for binary trees

$(3 * 4) + (-8)$

inorder: left - node - right
 $(3 * 4) + (0 - 8)$
 ist's not binary!
 → need parentheses

preorder: $+ (* 3 4) (- 8)$
 12 8 4

postorder: $3 4 * 0 - 8 +$
 12 8 4

cf. function prog.

Dez 5-10:29

pre: $7 2 3 5 4 6 7 9 8 10 11 12 13$

post: $3 4 2 6 5 8 10 7 12 11 9 13$

and(x) = $\{y : pre(y) < pre(x) \wedge post(y) > post(x)\}$

prec(x) = $\{y : post(y) < post(x) \wedge pre(y) < pre(x)\}$

Following (x) = $\{y : post(y) > post(x) \wedge pre(y) > pre(x)\}$

Children: $\{th. <, >\}$
 sub: $\{=, =\}$

Dez 5-10:34

Store in DB:

- parent-child $e1 \rightarrow e1^*$
- (parent - text) also children
- parent attr. implicit

| el | attr name | attr val |
|-------|-----------|----------|
| 6.3.2 | color | red |
| 6.3.2 | size | 15 |

| p/c | | name |
|---------|---------|-------|
| 6.3 | 6.3.1 | sect. |
| 6.3 | 6.3.2 | sect. |
| 6.3.2 | 6.3.2.1 | head |
| 6.3.2.1 | | text |

id parent el.type text pk part

Dez 5-10:54

// conty [name = 'Germany'] / city name / text

Select ~~a.id~~, e.content 'Berlin', 'Gotha', ...

from T_a, T_b, T_c, T_d, T_e, T_f

Where a.type = 'conty'

and b.type = 'name' and b.parent = a.id

and c.type = null and c.content = 'Germany'

and c.parent = b.id

and d.type = 'city' and d.parent = a.id

and e.type = 'name' and e.parent = d.id

and f.type = null and f.parent = e.id

Dez 5-11:07