NC STATE UNIVERSITY

ENGINEERING ONLINE

Lecture Notes

Course Number: CSC 513

Instructor: Dr. Singh

Lecture Number: 25

XQuery Declarations

The **declare** clause specifies things like

- Namespaces: declare namespace pref='value'
  - Predefined prefixes include XML, XML Schema, XML Schema-Instance, XPath, and **local**
- Settings: declare boundary-space preserve (or strip)
- Default collation: a URI to be used for collation when no collation is specified

Defining Functions

```
declare function local:itemftop($t)
 {
  local:itemf($t,())
 };
```

- Here **local:** is the namespace of the query
- The arguments are specified in parentheses
- All of XQuery may be used within the defining braces
- Such functions can be used in place of XPath expressions

REDUCED TO AFTER 200

207

Functions with Types

```
declare function local:itemftop($t as element())
    as element()*
{
    local:itemf($t,())
};
```

- Return types as above
- Also possible for parameters, but ignore such for this course

Fib (n)

fib = new int[ ]

n	0	1	2	3	4	5	6
Fib(n)	1	1	2	3	5	8	13

Space O(n)

---

fib(n) in Java

iterative ITERATIVE

O(1) space
O(n) time int current;

```
int fib( int n) {
  int previous = 1;   1
  int prevprev = 0;   1
  int current = 0;    1
  for (int i = 2 ; i <= n; i++) {
2     current = previous + prevprev;
1     prevprev = previous;
2     previous = current; }
  return current
}
```

```
fib(n)  {
  fibaux(n,1,0)
}

fibaux { n, next, result) {
    if(n=0)
    then result
    else  fibaux(n-1, next + result, next)
   }
```

call it with fib(x, y, z)

```
   fib($a, $b, $n) {
   if((n=0) or (n=1))
```

a := x
b := y
n := z

```
then1
   else if((a=0) and (b=0))
        then if n 1
        else if (n>1)
          result = a + b
          a = b
          b = result
          fib(a, b, n-1)
```

fib(b, result, a+b, n-1)

TAIL RECURSION

```
if (n > 1)
then
    result = a + b
    a = b
    b = result
    fib (a, b, n-1)
```

EXPRESS IN XQUERY

(i) `fib (b, a+b, n-1)`

(ii)
```
let result := a+b
let a := b
let b := result (?)
return fib(a,b,n-1)
```

IN XQUERY (any FUNCTIONAL lang)
- NO ASSIGNMENT STATEMENT

Can we write let $a twice

```
let $a := 1
let $a := 2
```

DIFFERENT from

```
let $a := 1
return
  (let $a := 2
   return $a,
   $a
  )
```

let $a := 1

let $a := $a + 1

FUNCTIONAL WEB

```
int x
.
.
fib( ) {
int x = x+1
```

XQuery Quantification: 1

- Two quantifiers **some** and **every**
- Each quantifier expression evaluates to true or false
- Each quantifier introduces a bound variable, analogous to **for**

```
for $x in ...
where some $y in ...
satisfies $y ... $x
return ...
```

for $s in singers...
where (some $song in $s/*
satisfies ($song/@lg='en'))
[return $s]

Here the second $x refers to the *same* variable as the first

[return $song]

∀ every ≅ ∧

$(\forall x : P(x) \rightarrow Q(x))$ $P(1) \wedge P(2) \wedge P(3) \ldots$

$(\exists x : P(x) \wedge Q(x))$ $P(1) \vee P(2) \vee P(3) \ldots$

```
and( P[n] ) { ...
  boolean result = true;
  for (int i = 0; i < n; i++){
    result = result && P[i]
  }
  return result)
  }
```

∧, ∨, +, *, min, max

−

or

initial: false

sum	0
mult	1
minimum	∞
max	−∞

||, +, *, min, max

ℕ