Lecture 6 In Class

Note Title 1/24/2008

```
Xs = 1 | Ys
Ys = 1 | Zs                                   ? :
declare Fib
fun {Fib X|Xr Y|Yr Lim}
      X+Y | if X+Y > Lim then nil else
                  {Fib Xr Yr Lim} end end
Zs = {Fib 'Xs Ys 10000}
```

4.3.3.1 (p. 261)

```
declare Fib L Xs Ys Zs
proc {Fib L X|Xr Y|Yr Z|Zr}
      Z = X+Y
      {Fib L Xr Yr Zr} end
   Xs = 1|Ys
   Ys = 1|Zs
   thread {Fib L Xs Ys Zs} end
```

why does block?

```
declare Nth
fun {Nth Ls N} L|Lr = Ls in
      if N==0 then L else
      {Delay 500} {Nth Lr N-1} end end
```

Assignment 2: due 1/31 5PM

Exercises 17 & 19 at back of chapter 4

for Ex. 17 — do it in your favorite imperative language — Java, C, C++ — for the simple case we do in class today.

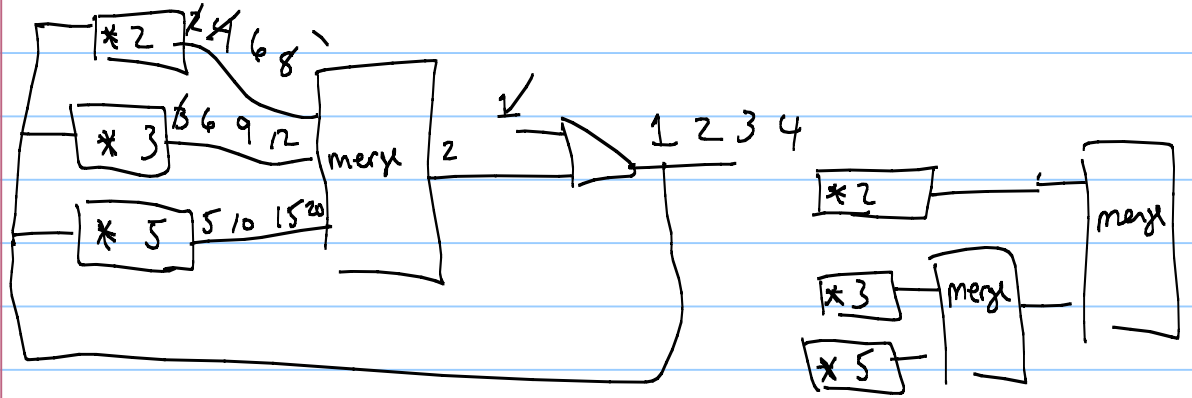
Grad Students: Consider the problem discussed in sec. 4.8.2 concerning adding a count of # of times a proc. is invoked.

— Discuss using a thread to maintain the count.

Hamming problem (simple version)

generate the first n integers of the form

$2^i 3^j 5^k$




```
fun lazy {Times N H}
   case H of X|Hr then N*X | {Times N Hr} end end
```

```
fun lazy {Merge Xs Ys}
   case Xs # Ys of (X | Xr) # (Y | Yr) then
      if X < Y then X | {Merge Xr Ys}
   elseif X > Y then Y | {Merge Xs Yr}
      else Y | {Merge Xr Yr} end end
end

H = 1 | {Merge {Times 2 H} {Merge {Times 3 H}
                                  {Times 5 H}}}
        }
```

Time

temporal specifications

real-time

hard RT – every response is timely according to spec.

soft RT – contemplates some responses are allowed to be untimely

Oz support for S.R.T:

{Delay N} – delay at least N ms.

{Alarm N U} – bind U to unit after at least N ms (in separate thread)

{Time.time} – # secs since start of year

SRT task generate a list of something
at a rate of 1 element / second

```
fun {Ticker}
   fun {Loop} X={Time.time} in {Delay 1000} X|{Loop} end
in
   Thread {Loop} end
end

H={Ticker}
```

Repairs: 4.6.2

```
fun {Ticker}
   {fun Loop N}
      T = {Time.time}
   in
     if T>N Then {Delay 900}
   elseif T<N Then {Delay 1100}
    else {Delay 1000}
    end
      N | {Loop N+1}
   end
in
thread {Loop {Time.time}} end
```

Can't build Hard RT system on SRT base

Threading a variable through a computation.