

Ref: 15210 CMU Course website

- Integration example in ML

- Sorting competition (ML (Milton) code runs in  $\frac{1}{3}$  s)

(Java: 1.1 sec w/pairs  
0.6 sec w/floats.)

Updating w/out mutation

For arrays, we need to copy (linear time)

update:  $\alpha \text{ seq} \times (\text{int} \times \alpha) \rightarrow \alpha \text{ seq}$

def update(A, (i, v)) =

tabulate  $\lambda j. \text{if } j=i \text{ then } v \text{ else } A[j]$  |A|

$W = O(|A|)$   $S = O(1)$

inject:  $\alpha \text{ seq} \times ((\text{int} \times \alpha) \text{ seq}) \rightarrow \alpha \text{ seq}$

inject((a b c d), ((b x) ((z y)) (o z))) = (z, b, y, d)

$W = O(|A| + |V|)$   $S = O(\log |V|)$

Claims

If we want these costs, we must make inject primitive.

If we don't care about cost, we could implement it sequentially like this:

inject(A, v) = iterate A update v

Parallel filter

filter:  $(\alpha \rightarrow \text{Bool}) \rightarrow \alpha \text{ seq} \rightarrow \alpha \text{ seq}$

$\langle a, b, c, d, e, f \rangle$

map f s =  $\langle \text{t}, \text{f}, \text{f}, \text{t}, \text{t}, \text{f} \rangle$

$\langle a, d, e \rangle \leftarrow \text{result of filter.}$

Goal in terms of work, assuming f eval is const:  $W(|S|)$ .

Goal for span is  $\log n$ .

using tabulate, we can generate the sequence

$(\langle A \rangle, \langle \rangle, \langle \rangle, \langle d \rangle, \langle z \rangle, \langle \rangle)$

from  $\langle a, b, \dots, f \rangle$  &  $\langle t, f, f, t, t, f \rangle$ .

Then do:

reduce append  $\langle \langle A \rangle, \langle \rangle \langle \rangle \langle d \rangle, \langle z \rangle, \langle \rangle \rangle$

which has  $\log n$  span.  $\text{span} = O(\log |S|)$

$$W = O(|S| \log |S|).$$

inject  $(A, v)$

$\langle a \ b \ c \ d \ e \ g \rangle$
$\langle t \ f \ f \ t \ t \ f \rangle$
want $a \quad d \ e$ in positions
$\downarrow \quad \downarrow \quad \downarrow \quad \leftarrow$ $0 \quad 1 \quad 2$

The prefix sum should help here.

$\langle a \ b \ c \ d \ e \ g \rangle$
$\langle 1 \ 0 \ 0 \ 1 \ 1 \ 0 \rangle$
scan $\langle \langle 0 \ 1 \ 1 \ 2 \ 3 \rangle, 3 \rangle$
$= \quad = \quad = \quad \leftarrow$ are the correct indices for $a, d, e$ in result.

fun filter f A = let

```

{ val f1 = map (fn x => if f(x) 1 else 0) A
  val (offset, total) = scan 0 2(x,y).x+y f1
  val v = tabulate (fn i => (offset(i), A[i])) |A|
  } in inject (tabulate (fn _ => a(0)) total v
end

```

Homework write flatten fn that runs in linear work.

flatten  $A : (\alpha \text{ seq}) \text{ seq} \rightarrow \alpha \text{ seq}$

fun flatten\_first\_try A = reduce  $\langle \rangle$  append A

$O(|R| \log |A|)$   $\text{span} = O(\log |A|)$

R = result