

Solution to exercise sheet 9

11.06.2013

Exercise 1: Given an array $A[0, \dots, n-1]$ of elements and an Boolean array $B[0, \dots, n-1]$ s.t. $B[n-1] = 1$ holds. The array B induces a division of A , i.e., every subarray of A ends on an index j with $B[j] = 1$.

Now we are looking for the segmented prefix sums, i.e., the prefix sums of the by B induced subarray $A[i+1, \dots, j]$ of A such that the following holds

1. $i = -1$ or $B[i] = 1$,
2. $B[r] = 0$ for $i < r < j$,
3. $B[j] = 1$.

Construct an algorithm which runs in $O(\log n)$ steps on an EREW-PRAM with $O(n)$ processors.

Note: You can use the principle of pointer jumping.

Solution: Idea: Use principle of pointer jumping, where the B -array is used for the linking. Hence set

$$B[i] = \begin{cases} \text{nil} & , \text{ if } B[i] = 1 \\ i + 1 & , \text{ otherwise} \end{cases}$$

Algorithm 1: segprefix (EREW-PRAM)

```

1 global  $A, s$ : array[0, ...,  $n-1$ ] of real,  $B$ : array[0, ...,  $n-1$ ] of integer;
2 local  $p, N, i$ : integer;                                //  $p$  and  $N$  initialized,  $N = n$ 
3  $S[p] := A[p]$ ;                                           // copy array
4 if  $B[p] = 1$  then  $B[p] := \text{nil}$ ;
5 else  $B[p] := p + 1$ ;
6 while  $B[p] \neq \text{nil}$  do
7    $S[B[p]] := S[B[p]] + S[p]$ ;
8    $B[p] := B[B[p]]$ ;
```

□

Exercise 2: Every element of an n -array list L is colored either blue or red.

Construct an efficient algorithm running on a CREW-PRAM which divides L into two lists such that one consists of only red and the other of only blue elements.

Two global variables **blist** and **rlist** shall contain the initial pointers to these lists at the end of the computation.

Solution: The following algorithm solves the task. During the process he builds two lists **rnext**, **bnext** which are at the end the lists of red/blue elements. They contain the indices of the separated lists. Initially $\text{rnext}[p] = p + 1$ if $A[p] \neq \text{nil}$; and similarly for **bnext**. \square

Algorithm 2: list-sort $\langle \text{CREW-PRAM} \rangle$

```

1 global rnext, bnext: array[0, ..., n - 1] of integer;
2       color: array[0, ..., n - 1] of (blue, red);
3       blist, rlist: integer;
4 local p, N, i: integer; // p, N initialized n = N
5 for i := 0 to  $\lfloor \log N \rfloor$  do
6   | if rnext[p]  $\neq$  nil and color[rnext[p]]  $\neq$  red then rnext[p] := rnext[rnext[p]];
7   | if bnext[p]  $\neq$  nil and color[bnext[p]]  $\neq$  blue then bnext[p] := bnext[bnext[p]];
8 if p = 0 then
9   | if color[0] = red then blist := bnext[blist], rlist := 0;
10  | else rlist = rnext[rlist], blist := 0;

```
