TD 2 : P-RAM

1 List.filter

Question 1. Le L be a list containing n objects coloured either in blue or in red. Write an efficient EREW algorithm (Exclusive Read, Exclusive Write) separating the blue elements from the red (i.e. that builds a new list containing only the blue elements).

2 Finding roots

We give here a problem separating the models EREW and CREW. Consider a forest of binary trees. Each node i of a tree corresponds to a processor $P(i)$ and has a pointer to his parent $parent(i)$. We are looking for EREW and CREW algorithms so that each nodes knows the root of its tree $root(i)$, to prove the interest of concurrent reads (CR).

Question 2. Give a CREW algorithm so that each node i will know $root(i)$. Show the algorithm only uses concurrent reads and derive its complexity.

Question 3. Why can't EREW algorithms achieve such a complexity?

3 Unknown procedure

We define the following two operators for an array $\texttt{A} = [a_0, a_1, \ldots, a_{n-1}]$ of n integers:

- `prescan(A)` returns the array $[0, a_0, a_0 + a_1, a_0 + a_1 + a_2, \ldots, a_0 + a_1 + \ldots + a_{n-2}]$,
- `scan(A)` returns the array $[a_0, a_0 + a_1, a_0 + a_1 + a_2, \ldots, a_0 + a_1 + \ldots + a_{n-1}]$.

We have seen in lesson how to implement these two operators in time $O(\log n)$ on a EREW P-RAM. Consider the following procedure:

```
split(A, flags):
    idown := prescan(not(flags))
    iup   := n - reverse(scan(reverse(flags)))
    for i = 1 to n in parallel do:
        if flags[i]
            index[i] := iup[i]
        else
            index[i] := idown[i]
        fi
    done
    result := permute(A, index)
    return result
```

Question 4. Apply `split` to the following input. What does `split` seems to do on this example?

```
A     = [ 5 7 3 1 4 2 7 2 ]
flags = [ 1 1 1 1 0 0 1 0 ]
```

Question 5. Prove it actually does it for every possible input. What is the cost of `split`?

Consider now the procedure `mystery` below:

```
mystery(A, number_of_bits):
    for i = 0 to number_of_bits-1:
        bit[i] := [i-th bit of A[0], ..., i-th bit of A[n-1]]
        A := split(A, bit[i])
```

Question 6. Run `mystery` on `A = [ 5 7 3 1 4 2 7 2 ]` with `number_of_bits = 3`.

Question 7. What does `mystery` do?

Question 8. With inputs of size $O(\log n)$ bits, what is the complexity on n processors? What about p processors? Which are the most interesting values of p?