Efficient Algorithms
Summer term 2013

Institut für Theoretische Informatik
Leibniz Universität Hannover

Solution to exercise sheet 8

28.05.2013

Exercise 1: Given the problem

Problem: OR^n
Input: n Boolean values $x_0, \ldots, x_{n-1}$
Task: Compute $\bigvee_{i=0}^{n-1} x_i$

Show that $\mathrm{OR}^n \in \mathrm{CRCW}(n, 1)$.

Solution: We use a common CRCW-PRAM with n processors. If one of the values is

Algorithm 1: OR^n ⟨common CRCW-PRAM⟩

global S: Boolean, A: array$[0, \ldots, n-1]$ of Boolean;
local p, n: integer;
if $p = 0$ **then** $S =$ false;
if $A[p]$ **then** $S =$ true;

true, then the n-ary OR is true, otherwise it is false. For $k \leq n$ trues k processors write k-times in parallel a true into the field S. This is allowed for CRC-PRAMs. □

Exercise 2: In the shared memory of a PRAM a variable X is stored. What is the fastest way to distribute the value of X into the local memory of each processor on a PRAM? Construct for each type of the possible PRAMs an algorithm with a runtime as efficient as possible.

Solution:

Algorithm 2: distribute ⟨CR{C,W}W-PRAM⟩

global X : real;
local p, n: integer, x: real;
$x \leftarrow X$;

Hence distribute is in CR{C,W}W$(n, 1)$.
For the EREW-PRAM we will use a binary tree technique, i.e., at first X is copied to processor 0 then to 1, then cpu 0 and 1 copy the value to 2 and 3 so on and so forth.

Algorithm 3: distribute ⟨EREW-PRAM⟩

global X: real, $temp$: array$[0, n-1]$ of real; // assume $n = 2^j$ for some $j \in \mathbb{N}$
local p, n, i: integer, x: real;
if $p = 0$ **then** $temp[0] = X$;
for $i = 0$ *to* $(\log n) - 1$ **do**
  if $p < 2^i$ **then** $temp[p + 2^i] \leftarrow temp[p]$;
$x \leftarrow temp[p]$;

Here the runtime is $O(\log n)$. □

Exercise 3: Given the problem

Problem: AND^n
Input: n Boolean values $x_0, \dots, x_{n-1}$
Task: Compute $\bigwedge_{i=0}^{n-1} x_i$

Show that AND$^n \in$ CRCW$(m, \frac{n}{m})$ for $m \leq n$.

Solution: **Idea:** For $m < n$ processors the first m bits are checked in parallel and one writes a false into the global field if a false was read. Afterwards the array fields with the numbers $m+1$ to $2 \cdot m$ are checked until all bits have been considered. The runtime is $O\left(\lceil \frac{n}{m} \rceil\right)$. For the case $n = m$ the runtime is analogously to the OR-algorithm in Exercise 1 $O(1)$.

Algorithm 4: AND^n ⟨common CRCW-PRAM⟩

global: A : array$[0, \dots, n-1]$ of integer, S, n, m : integer;
local: p, i : integer;
$i \leftarrow 0, S \leftarrow$ true;
while $i < n$ **do**
  if $A[p+i]$ *is false* **then** $S \leftarrow$ false;
  $i \leftarrow i + m$;

Example. $n = 10, N = 4$.

ProzessorID	i_1	i_2	i_3
0	0	4	8
1	1	5	9
2	2	6	10
3	3	7	11

Arrayfeld	0	1	2	3	4	5	6	7	8	9
i_k	1	1	1	1	2	2	2	2	3	3

□