Technische Universität München
Institut für Informatik
Theoretische Informatik

WS 2015/16
Worksheet
14.12.2015

Fundamental Algorithms

K-Exercise 1

The following parallel implementation of the BucketSort algorithm is proposed:

```
BucketSortPRAM(A: Array [0..n−1]) {
   // Model: PRAM with n processors

   Create Array B[0..n−1] of Buckets;
   // all Buckets B[i] are created empty

   for i from 0 to n−1 do in parallel {
      insert A[i] into Bucket B[index(A[i])];
   }

   for i from 0 to n−1 do in parallel {
      BubbleSort( Bucket B[i] );
   }

   A := Concat( B[0], B[1], ..., B[n−1] );
}
```

Explanation: A Bucket may contain an arbitrary number of elements inserted successively using the insert ... into Bucket ... statement (Buckets could be implemented via linked lists, for example). Concat is a sequential algorithm that concatenates all buckets given as parameters in the provided order. For simplicity, we assume that Concat can have any number of parameters.

The function **index** has to distribute the array elements *evenly* into buckets. Hence, its range of values is $1, \ldots, n$, and elements are assigned to buckets with about the same probability (for the expected distribution of elements).

a) For the two parallel loops in BucketSortPRAM, state for both arrays A and B whether there is concurrent or exclusive read or write access to their elements.

b) Describe a parallel algorithm, for an EREW PRAM, to concatenate the buckets in the last step, using as many processors as possible. What is the parallel complexity to concatenate all buckets (depending on n), and how many processors can your algorithm use?