Priority Queues on Parallel Machines

Gerth Stølting Brodal

BRICS
Computer Science Department
Aarhus University
Denmark

July 1996

The problem

Question

How can parallelism be adopted to priority queue operations?

Assumptions

- Comparison model.
- Computational models: EREW and CREW PRAM.
- n denotes the size of a priority queue.

Priority queue operations

- MakeQueue
- FindMin(Q)
- Insert(Q, e)
- Meld(Q_1, Q_2)
- ExtractMin(Q)
- Delete(Q, e)*
- DecreaseKey(Q, e, e')*, $\quad e' \leq e$
- Build$(e_1, \ldots, e_n)$
- MultiInsert$(Q, e_1, \ldots, e_k)$
- MultiDelete(Q, k)

* Assume that it is known where element e is stored.

- The common sequential priority queue operations.

Previous and new results

	Pinotti Pucci '92	Pinotti Das Crupi '96	Chen Hu '94	Ranade *et al.* '94	This talk
Model	EREW	EREW*	EREW	Array	CREW
FindMin	1	$\log\log n$	1	1	1
Insert	$\log\log n$	$\log\log n$	–	1	1
ExtractMin	$\log\log n$	$\log\log n$	–	1	1
Meld	–	$\log\log n$	$\log\log \frac{n}{k} + \log k$	–	1
Delete	–	$\log\log n$	–	–	1
DecreaseKey	–	$\log\log n$	–	–	1
Build	$\log n$	–	$\log \frac{n}{k} \log k$	–	$\log n$
MultiInsert	–	–	$\log\log \frac{n}{k} + \log k$	–	$\log k$
MultiDelete	–	–	$\log\log \frac{n}{k} + \log k$	–	–

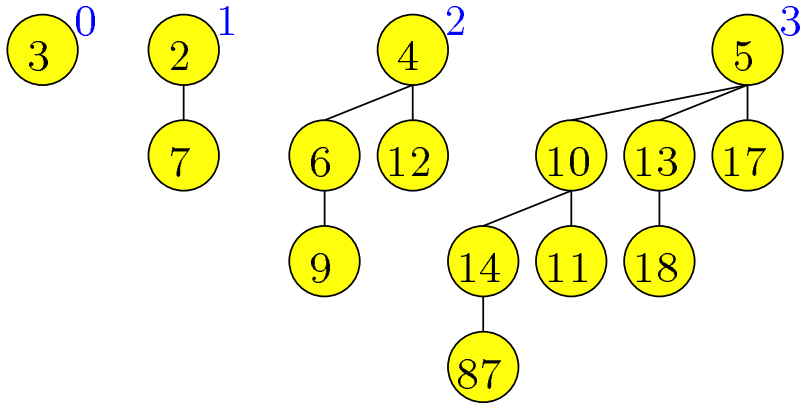
*Delete and DecreaseKey require the CREW PRAM

A simple priority queue

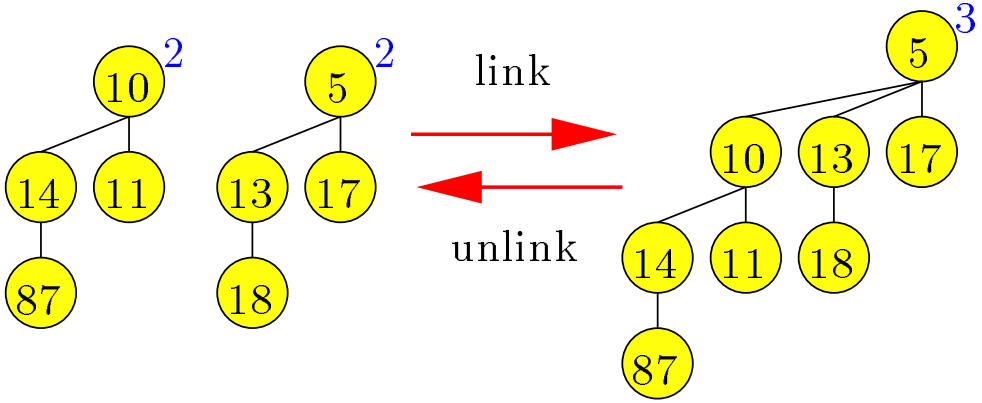
Operations: INSERT, MELD, EXTRACTMIN.

Basic representation: A forest of heap ordered binomial trees.

Binomial trees:



Linking:

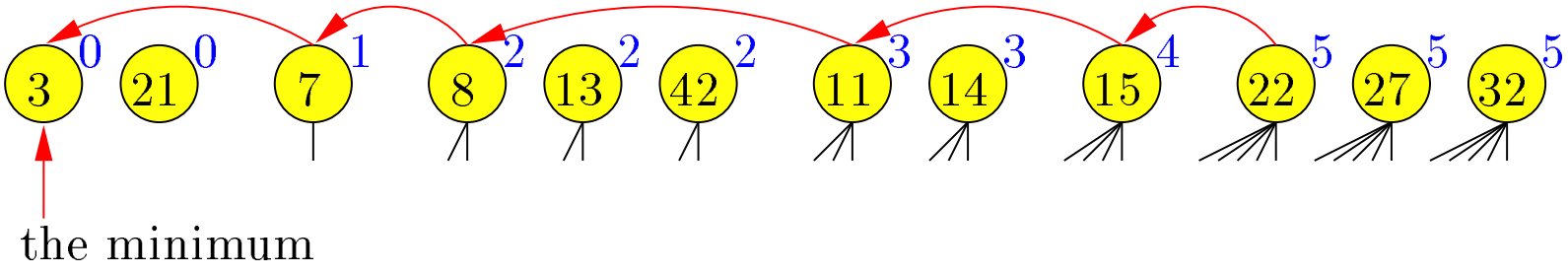


The root invariant

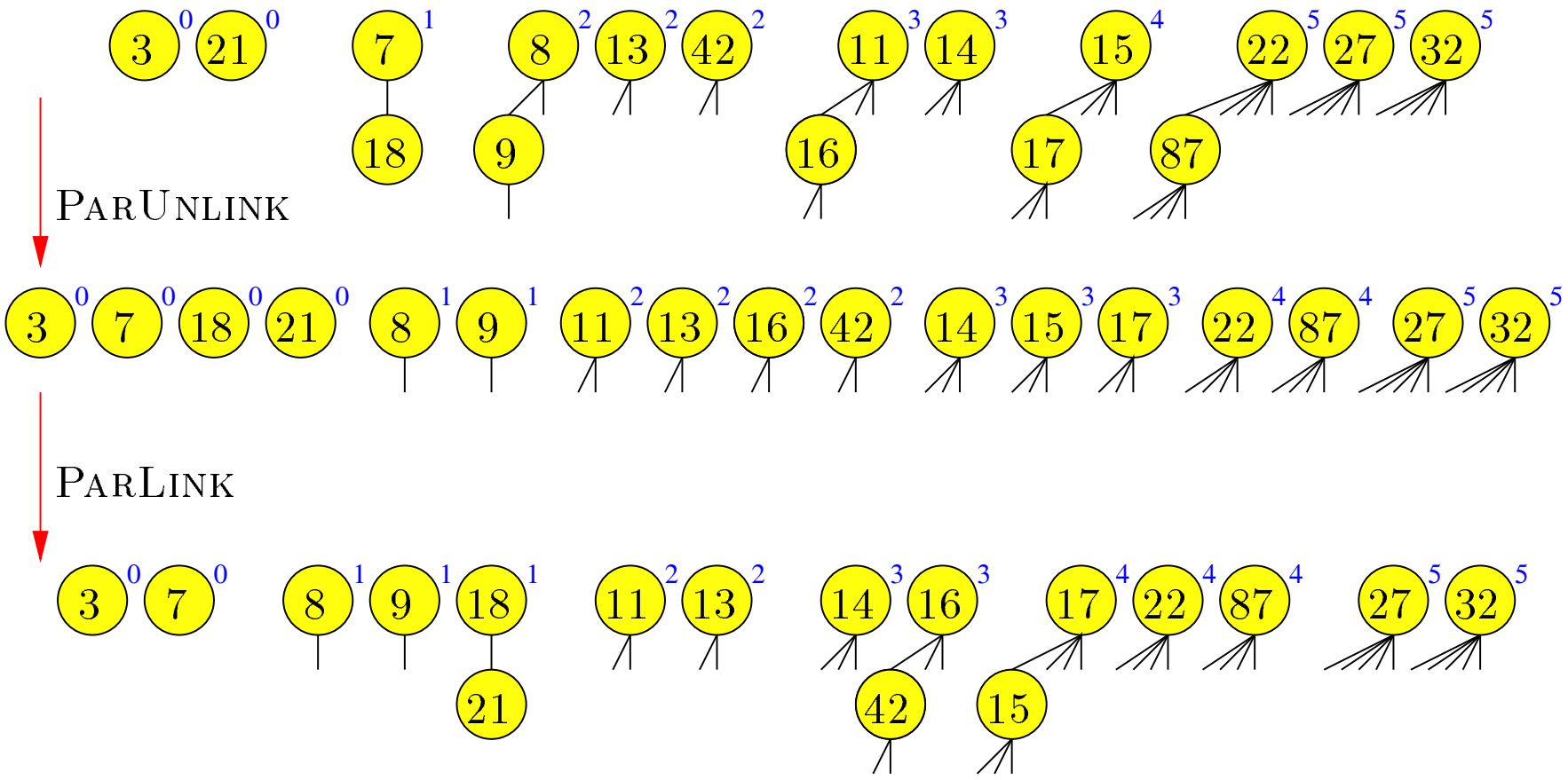
The invariant

- There are 1, 2 or 3 trees of each rank.
- The minimum root of rank i is smaller than all roots of rank $\geq i$.

Example



Parallel linking and unlinking



Notice that ParUnlink + ParLink is *not* the identify.

Priority queue operations

FindMin(Q)

    return $\min(Q.L[0])$

Insert(Q, e)

    $Q.L[0] := Q.L[0] \cup \{e\}$

    ParLink(Q)

Meld(Q_1, Q_2)

    for $p := 0$ **to** $\log n$ **pardo** $Q_1.L[p] := Q_1.L[p] \cup Q_2.L[p]$

    do 3 times ParLink(Q_1)

ExtractMin(Q)

    $e :=$ delete-min$(Q.L[0])$

    ParUnlink(Q)

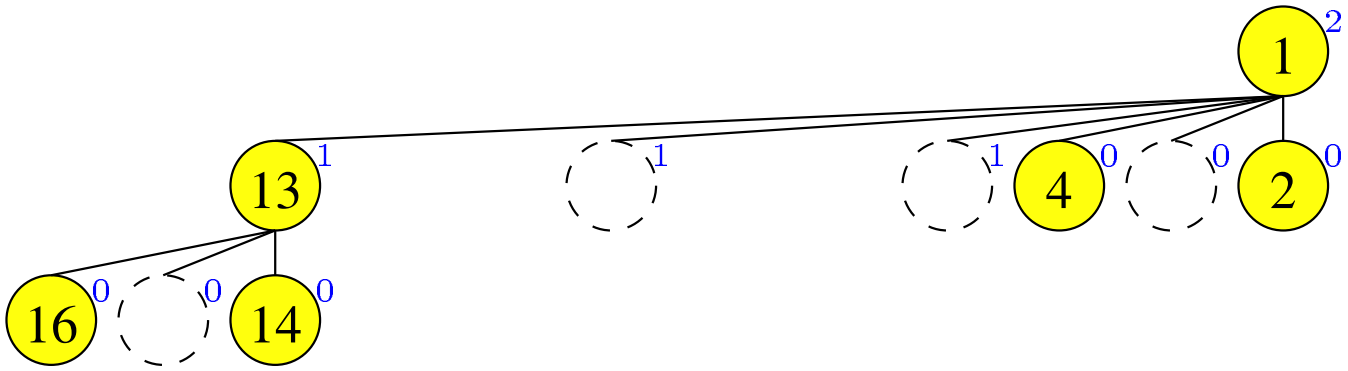
    ParLink(Q)

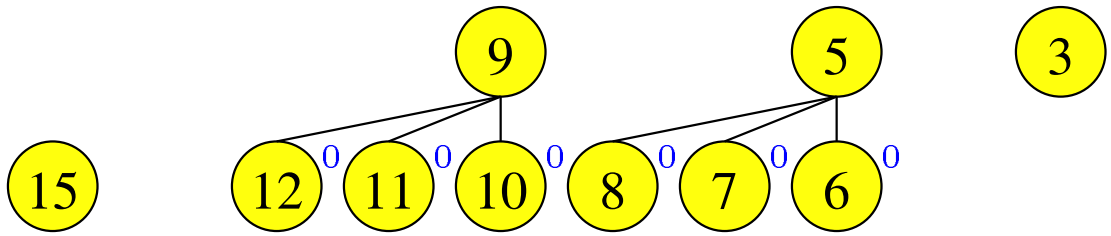
The DELETE operation

Basic ideas

- A node of rank r has three children of each rank $0, \ldots, r-1$.
- Subtrees can be missing — holes.
- Invariant: At most two holes have equal rank in the forest.

Example

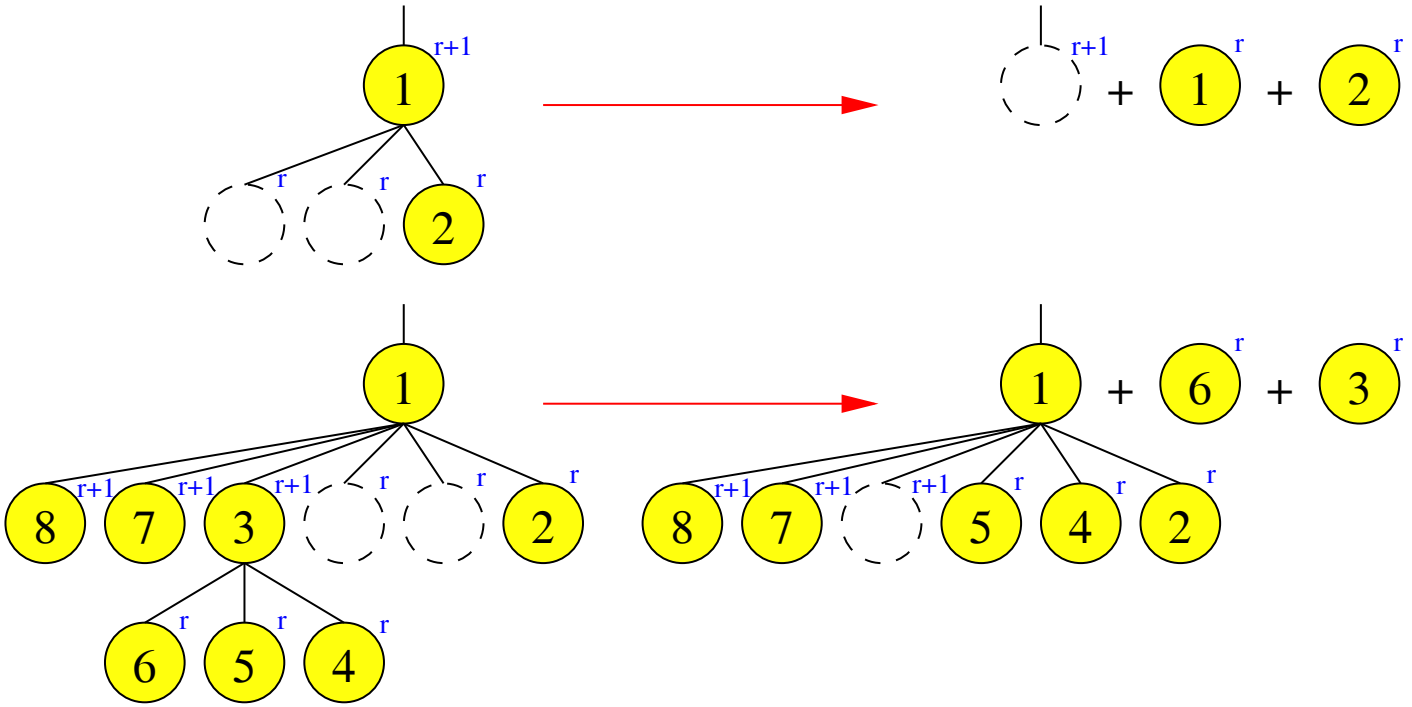



9
5
3
15
12
0
11
0
10
0
8
0
7
0
6
0

The elimination of holes

Basic idea: Replace two rank r holes by a rank $r+1$ hole.

Assume w.l.o.g. that the two holes are siblings.



Performing the transformations in parallel for each possible r bounds the #holes of each rank by two.

Conclusion

Summary of results

A CREW PRAM priority queue implementation supporting:

Operations	Time	Work
Insert, Meld, ExtractMin	$O(1)$	$O(\log n)$
Delete, DecreaseKey	$O(1)$	$O(\log n)$
Build	$O(\log n)$	$O(n)$
MultiInsert	$O(\log k)$	$O(k + \log n)$

Open problems

- Can the work be reduced to the best known sequential bounds?
- MultiDelete-operations?