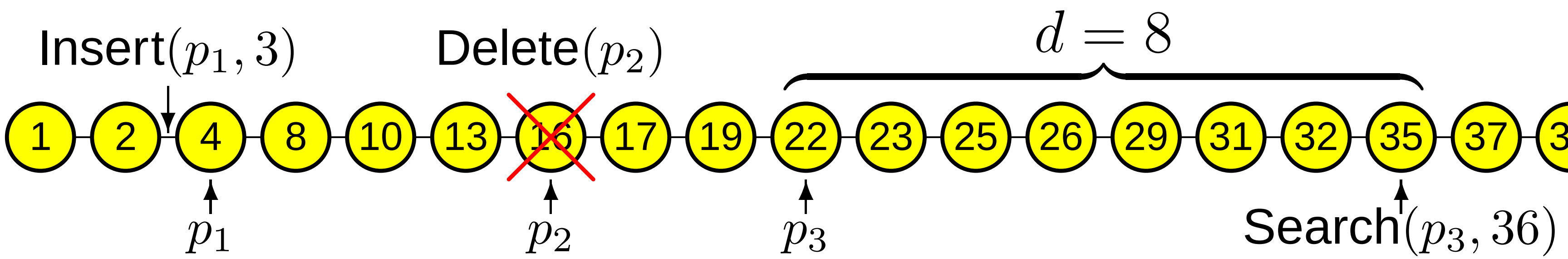
Optimal Finger Search Trees in the Pointer Machine

Gerth Stølting Brodal
BRICS
University of Aarhus

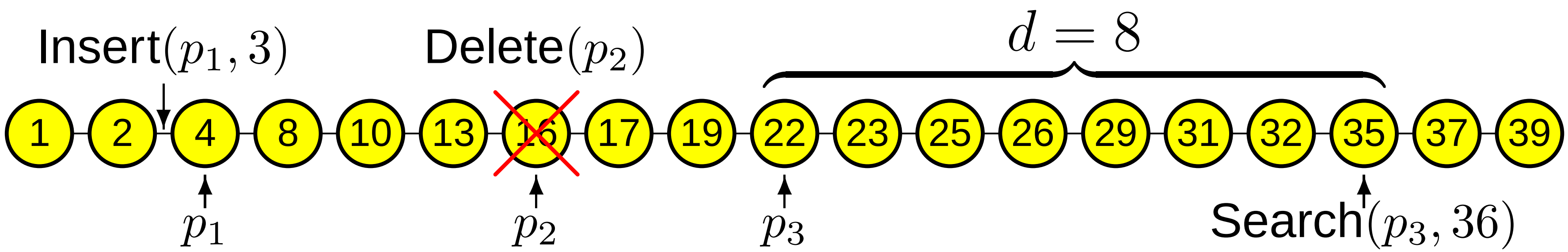
George Lagogiannis Christos Makris
Athanasios Tsakalidis Kostas Tsichlas
University of Patras

Dagstuhl seminar on "Data Structures", March 1, 2002

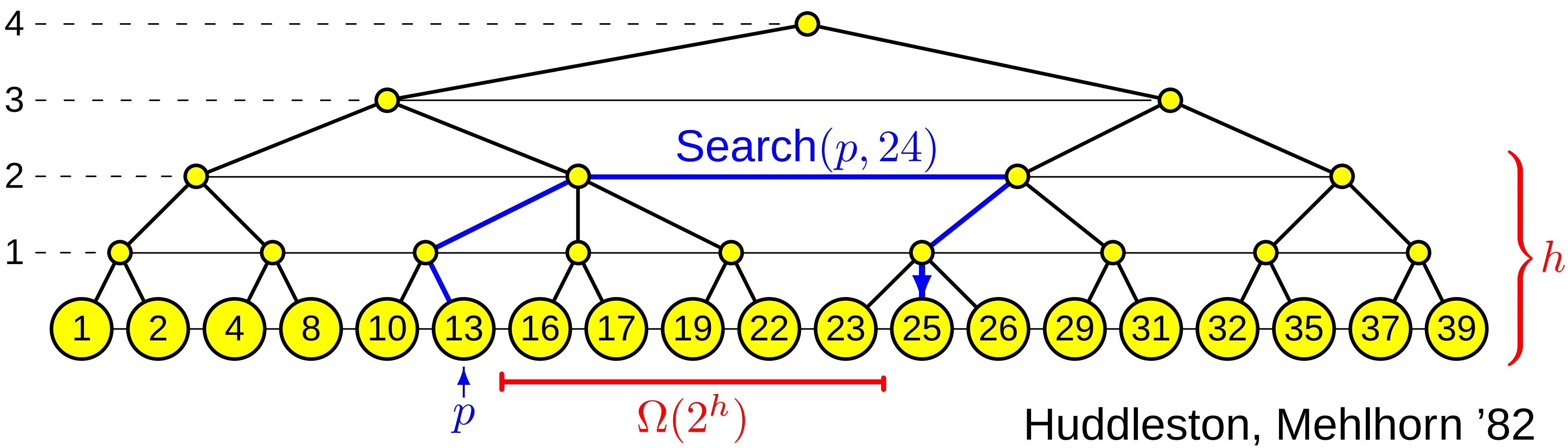
Finger Search Trees



Finger Search Trees

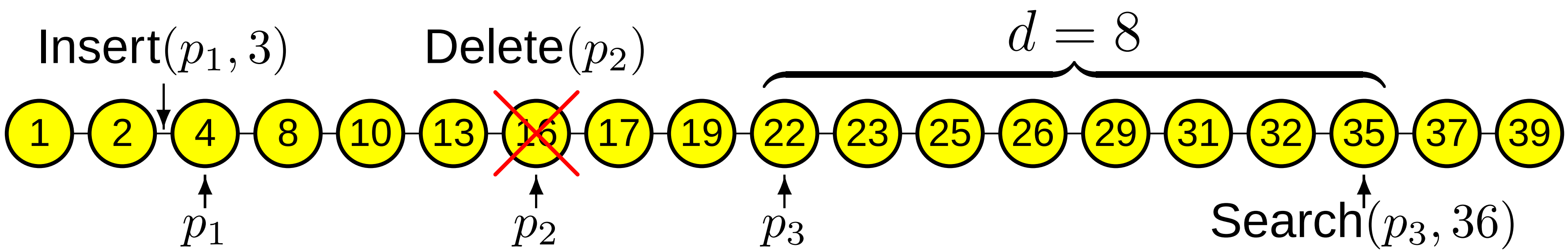


Insert, Delete	$O(1)$
Search	$O(\log d)$



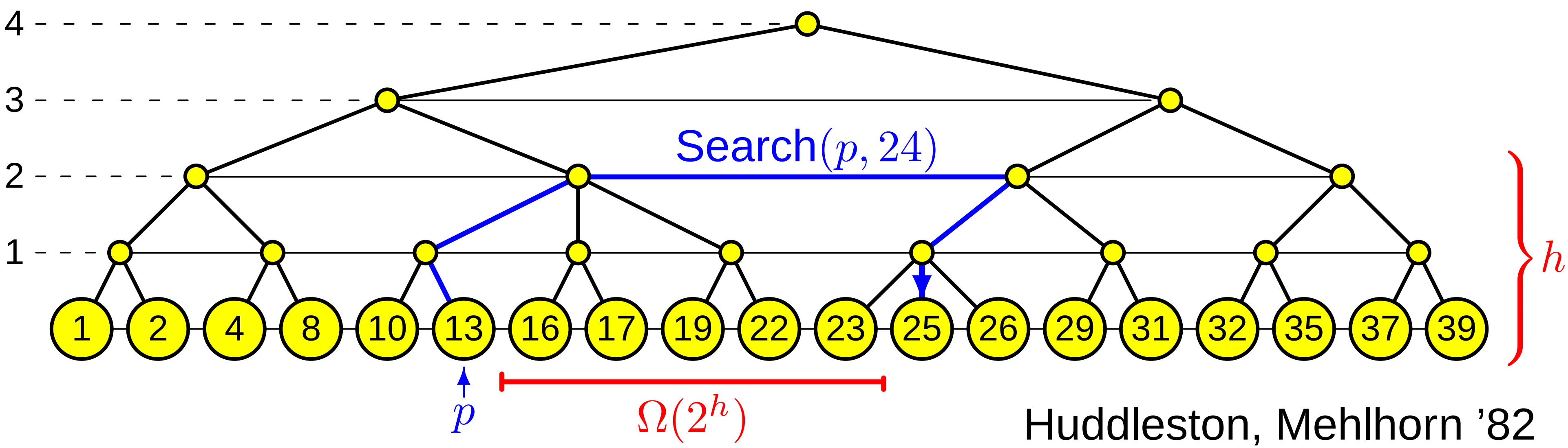
Huddleston, Mehlhorn '82

Finger Search Trees



Insert, Delete	$O(1)$
Search	$O(\log d)$

Amortized



Huddleston, Mehlhorn ’82

History

<table>
<tr><th></th><th>Insert</th><th>Delete</th><th>Search</th></tr>
<tr><td>AVL-trees, (2,3)-trees [a)]</td><td colspan="2">$\log n$</td><td>$\log n$</td></tr>
<tr><td>Red-black-trees, (2,4)-trees [a)]</td><td colspan="2">1[b)]</td><td>$\log n$</td></tr>
<tr><td>Levcopoulos, Overmars ’88 [a)]</td><td colspan="2">1</td><td>$\log n$</td></tr>
<tr><td>Guibas et al. ’77, Tsakalidis ’85 [a)]</td><td colspan="2">1[c)]</td><td>$\log d$</td></tr>
<tr><td>Harel, Lucker ’79 [a)]</td><td colspan="2">$\log^* n$</td><td>$\log d$</td></tr>
<tr><td>Huddleston, Mehlhorn ’82 [a)]</td><td colspan="2">1[b)]</td><td>$\log d$</td></tr>
<tr><td>Brodal ’98 [a)]</td><td>1</td><td>$\log^* n$</td><td>$\log d$</td></tr>
<tr><td>This talk [a)]</td><td colspan="2">1</td><td>$\log d$</td></tr>
<tr><td>Dietz, Raman ’94 [d)]</td><td colspan="2">1</td><td>$\log d$</td></tr>
<tr><td>Andersson, Thorup ’00 [e)]</td><td colspan="2">1</td><td>$\sqrt{\frac{\log d}{\log \log d}}$</td></tr>
</table>

[a)] Pointer machine [b)] Amortized [c)] $O(1)$ movable fingers

[d)] Comparison RAM [e)] Word RAM

This Talk

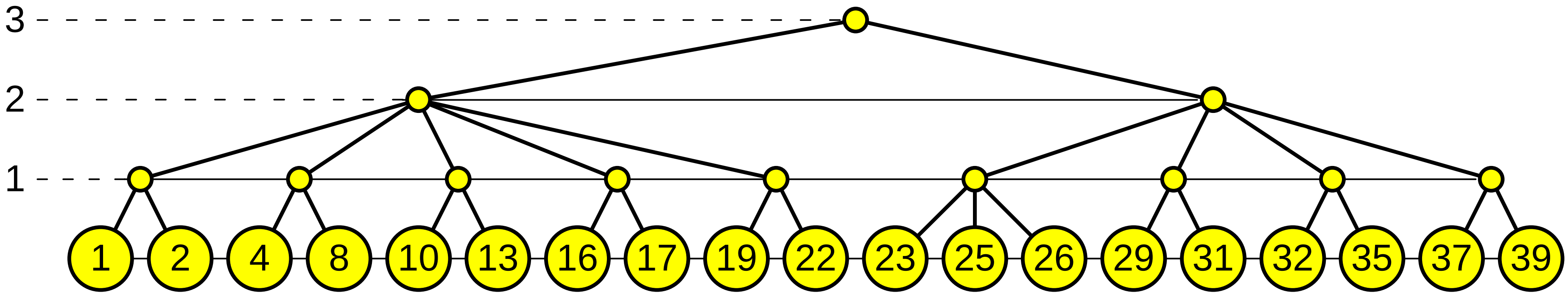
Insert	$O(1)$
Delete	$O(1)$
Search	$O(\log d)$

Worst-case

Pointer machine

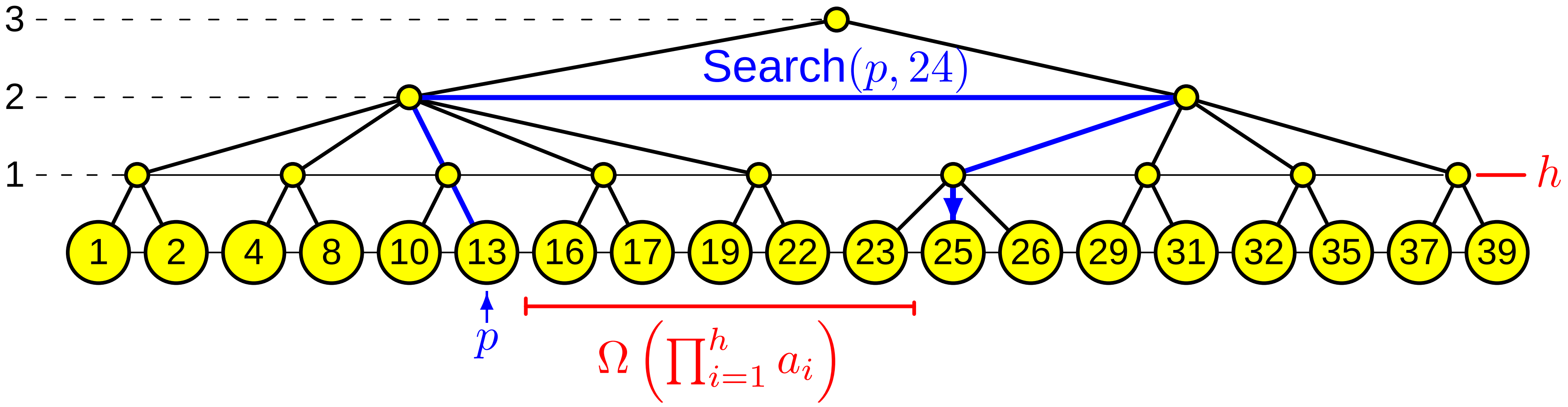
- Level dependent degrees (Andersson '96)
- Components (Brodal '96)
- Multi-split and multi-join
- Incremental (pre)processing

Level Dependent Degrees



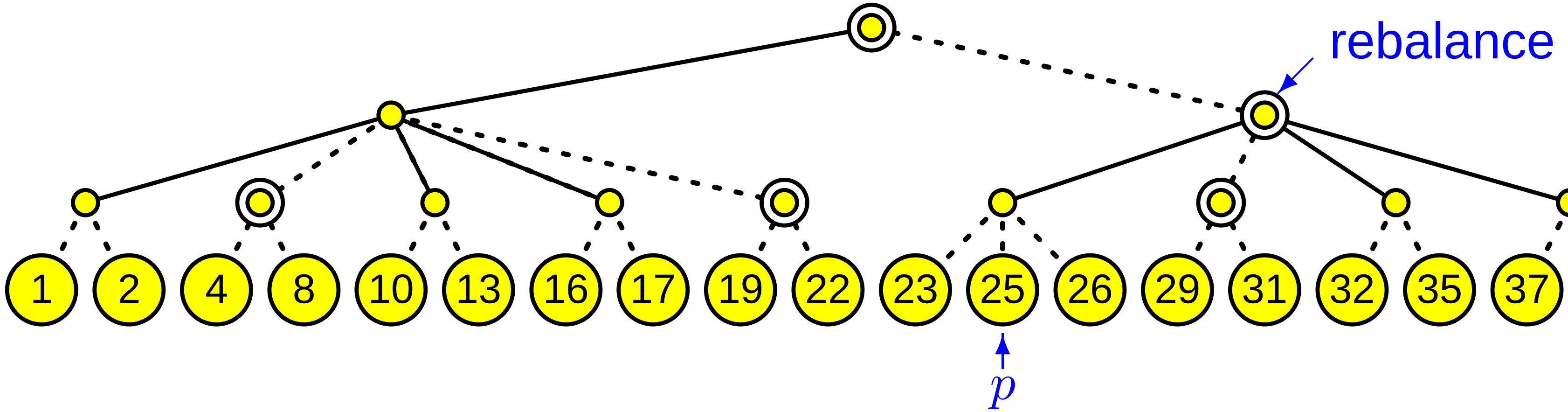
- Level i nodes (except root) have degree $[\, a_i : {a_i}^3 \,]$
- Each node represented by Levcopoulos, Overmars ’88

Level Dependent Degrees



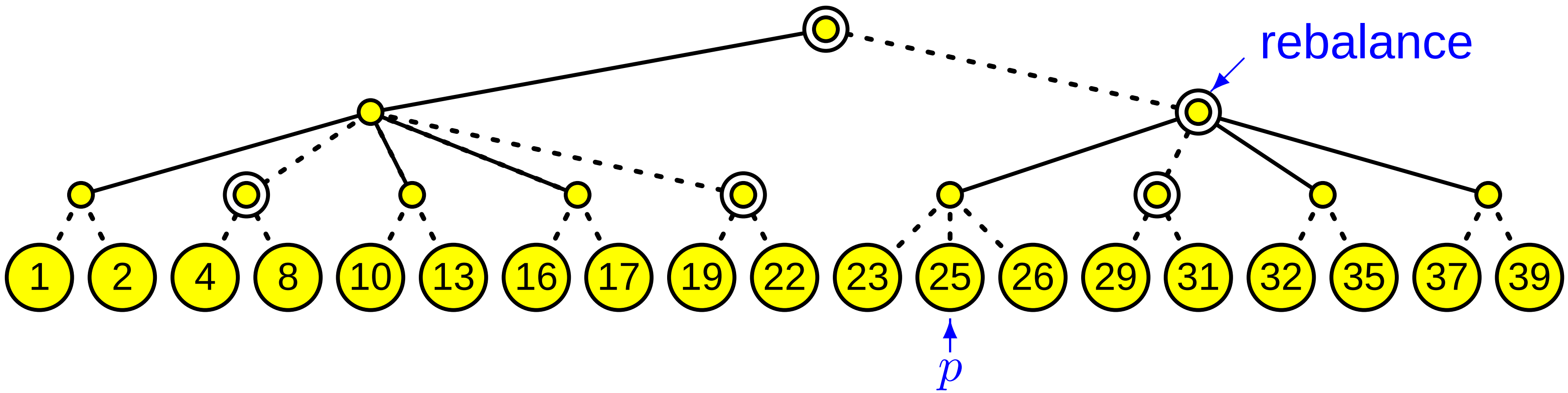
- Level i nodes (except root) have degree $[\, a_i : {a_i}^3 \,]$
- Each node represented by Levcopoulos, Overmars '88
- Search $O(\sum_{i=1}^{h+1} \log a_i) = O(\log d)$ if $a_{h+1} = a_h^{O(1)}$

Components

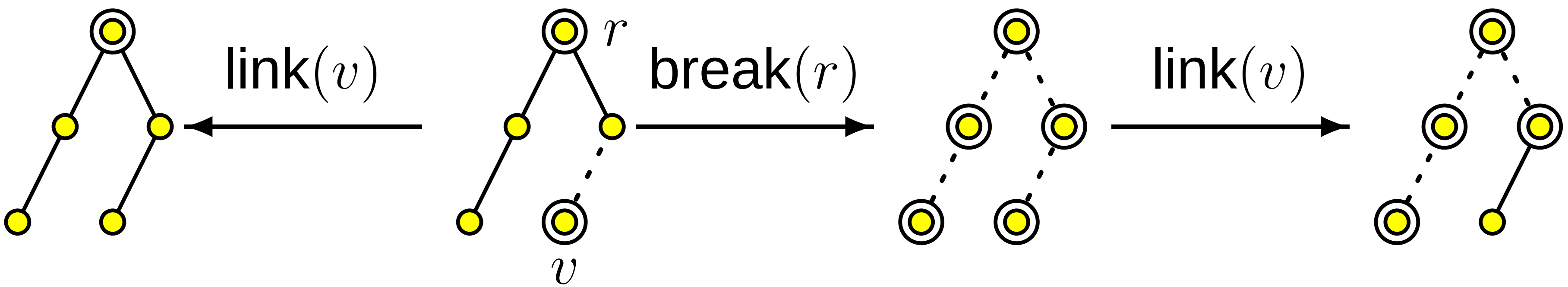


- Mechanisme to identify where to rebalance
- Components = partition of internal nodes into subtrees

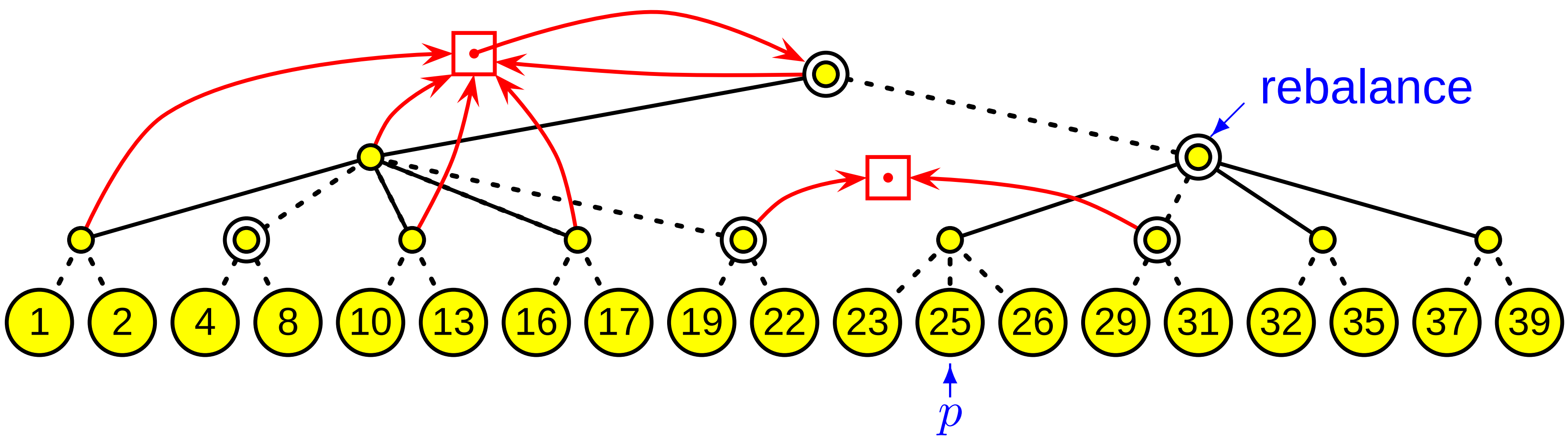
Components



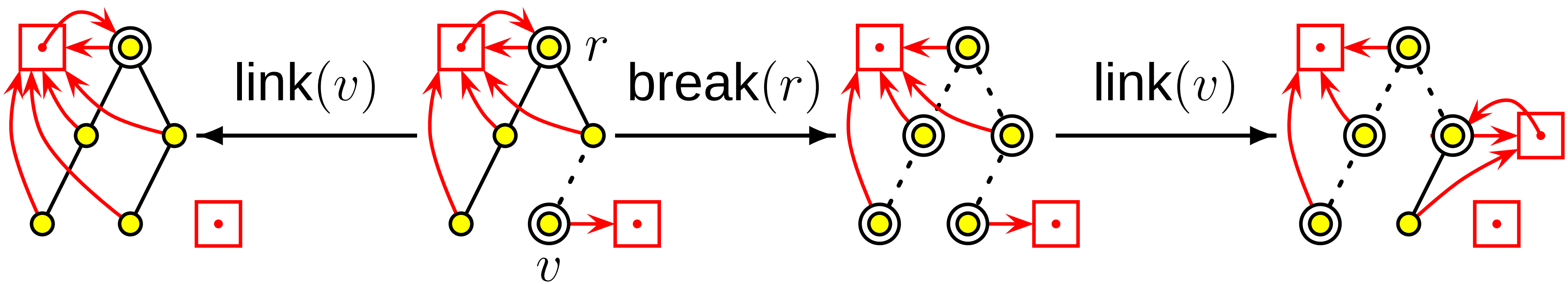
- Mechanisme to identify where to rebalance
- Components = partition of internal nodes into subtrees



Components



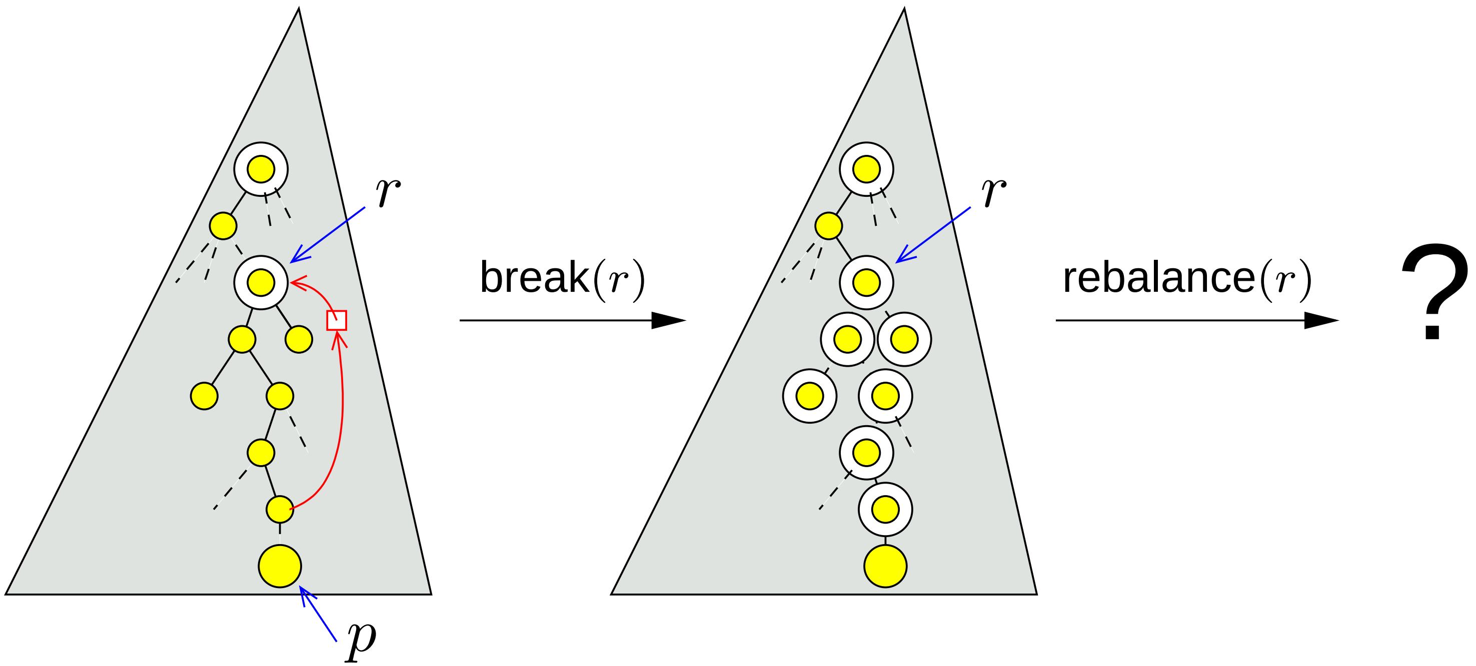
- Mechanisme to identify where to rebalance
- Components = partition of internal nodes into subtrees
- component records with root pointers (nil if singleton)



Updates

Insert(p, e)
r = component root
create-leaf(p, e)
break(r)
rebalance(r)

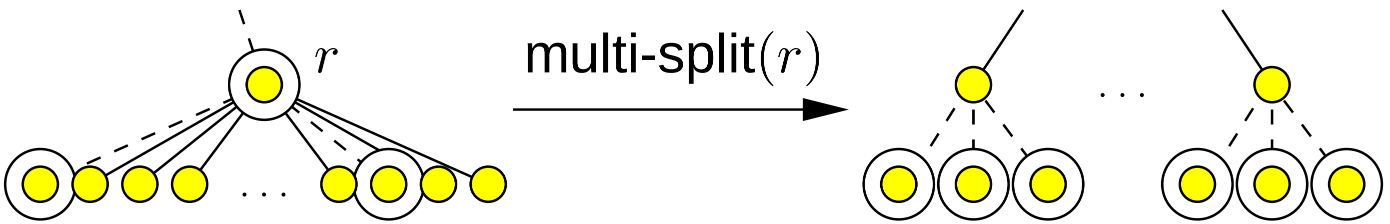
Delete(p)
r = component root
remove-leaf(p)
break(r)
rebalance(r)



rebalance(r)

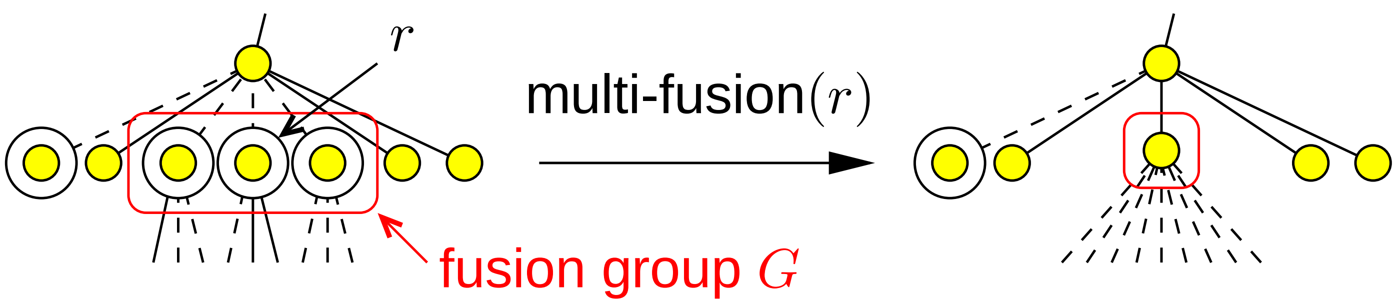
degree$(r) > 3{a_i}^2$

- link(r)
- split r into at most a_i nodes of degree $[{a_i}^2 : 3{a_i}^2]$



degree$(r) < {a_i}^2$

- $\forall v \in G$: break(v), G total degree $[{a_i}^2 : 3{a_i}^2]$
- fusion r with all nodes in G
- link(r)

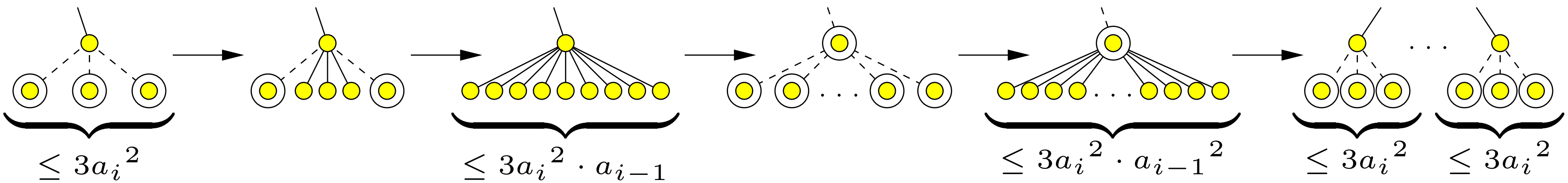


Proof of Degree $[a_i : {a_i}^3]$

Assumption : Nodes at level $i-1$ have degree $[a_{i-1} : {a_{i-1}}^3]$

$\leq {a_i}^3$

- split at level $i-1$ generates $\leq a_{i-1}$ nodes
- degree at level $i \leq 3{a_i}^2 \cdot {a_{i-1}}^2 \leq {a_i}^3$

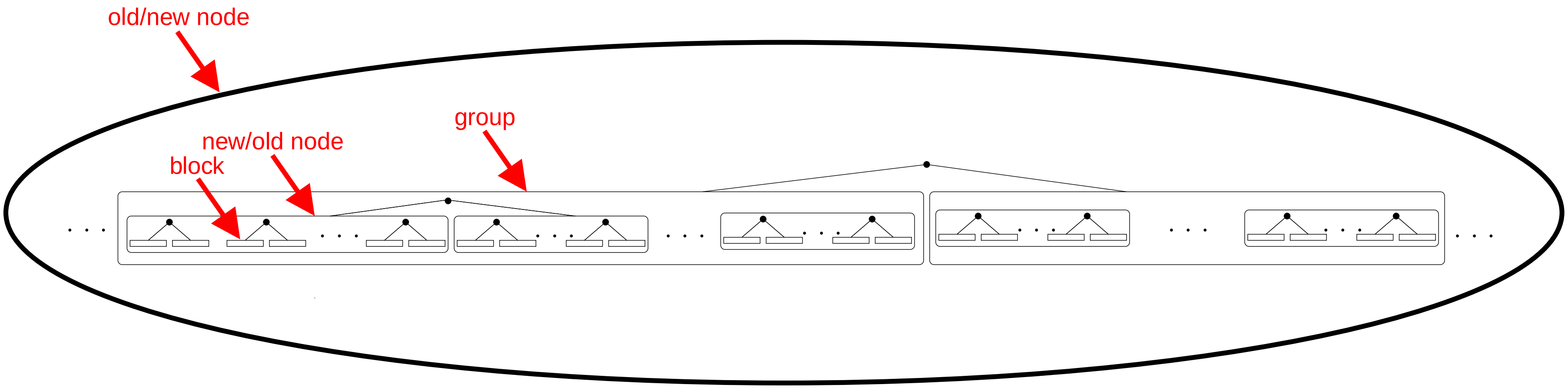


$\geq a_i$

- fusion at level $i-1$ fusions $\leq 3a_{i-1}$ nodes
- degree at level $i \geq {a_i}^2/(3a_{i-1})^2 = a_i$

$\boxed{a_i = 9 \cdot {a_{i-1}}^2}$

Incremental Processing



Internal representation of a node

- Blocks (pairs of)
- Nodes (pairs of)
- Splitting groups
- Fusion groups (pairs of)
- Fusion/splitting group records

Conclusion and Open Problems

Insert	$O(1)$
Delete	$O(1)$
Search	$O(\log d)$

Worst-case

Pointer machine

- Simpler construction ?
- Direct binary tree solution ?
- (a, b)-trees with worst-case update time ?
- Components usefull for deamortization of other structures ?