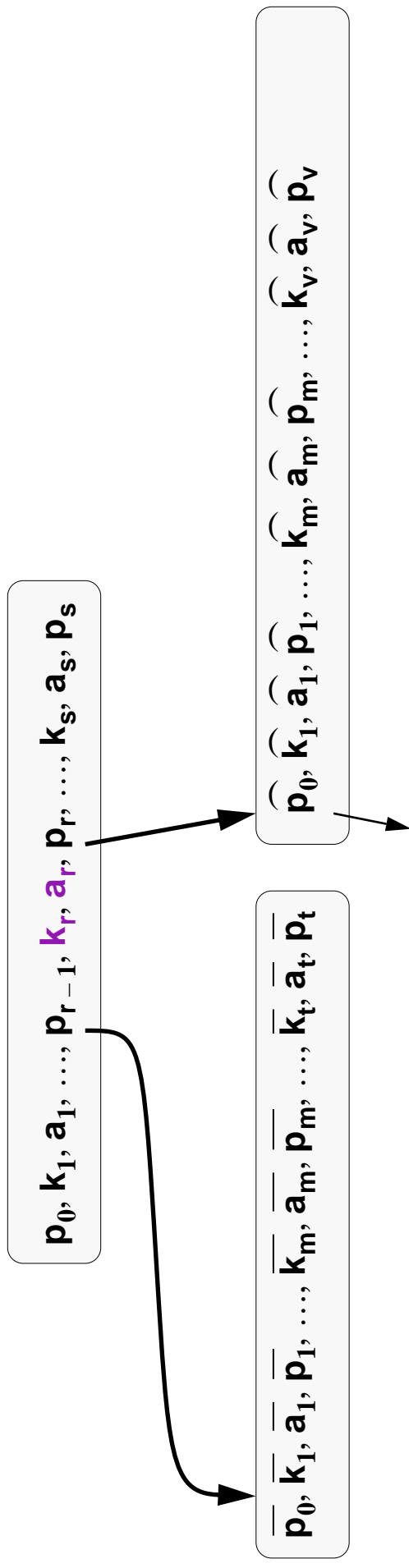


LÖSCH-ALGORITHMUS

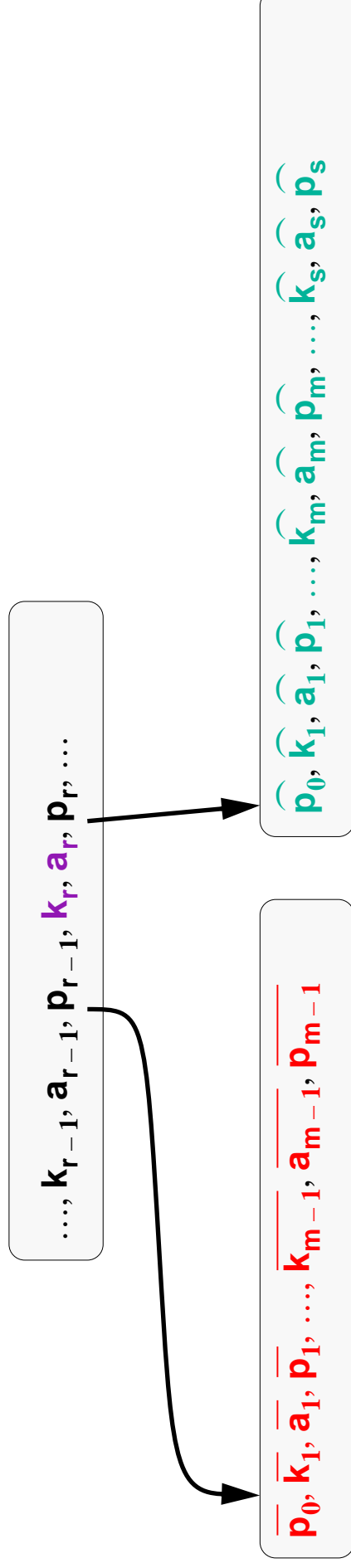
- Vorgehen
 - ➔ Suchen der Lösch-Stelle
 - ➔ Entfernen des Elements
 - ... In Blatt, o.k.
 - ... In innerem Knoten: Ersetzen durch Inorder-Nachfolger
 - ➔ Sicherstellen, daß weiterhin mindestens m Elemente in dem Knoten enthalten sind.
- Falls der Knoten jetzt m-1 Element enthält (Underflow), Ausbalancieren oder Verschmelzen des Knotens mit einem Geschwister-Knoten



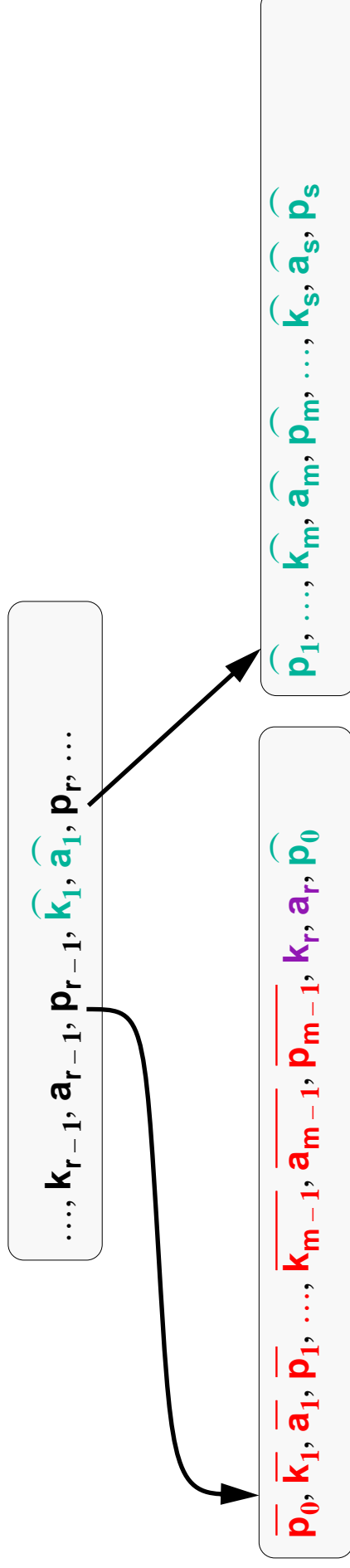
LÖSCH-ALGORITHMUS (FORTS.)

- Underflow-Behandlung

Balancierung (rechtes Geschwister hat mehr als m Elemente)



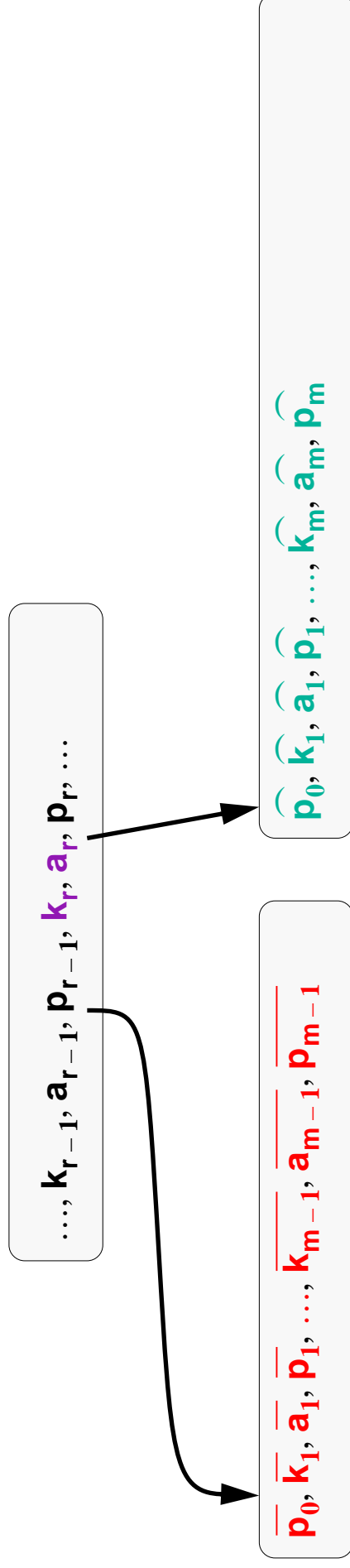
Übernehmen eines Elements vom rechten Geschwister



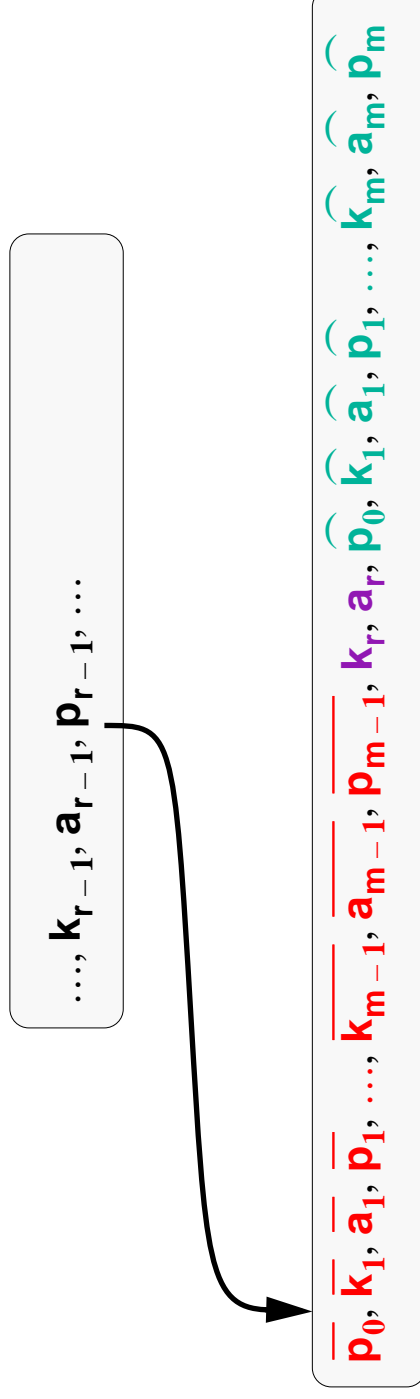
LÖSCH-ALGORITHMUS (FORTS.)

- Underflow-Behandlung

Verschmelzen (rechtes Geschwister hat genau m Elemente)



Mergen der Elemente (inkl. Trenn-Element aus dem Vorgänger)



KOMPLEXITÄT DES EINFÜGE-/LÖSCH-ALGORITHMUS

- Einfügen

Einfügen immer in einem Blatt

schlimmster Fall: Overflow in jedem Knoten auf dem Pfad

→ Komplexität von der Ordnung der Höhe

- Löschen

Löschen in einem Knoten bewirkt Löschen des Inorder-Nachfolgers (auf einem Blatt)

schlimmster Fall: Underflow in jedem Knoten auf dem Pfad

→ Komplexität von der Ordnung der Höhe

B-Baum stellt eine Realisierung eines DDs auf einem peripheren Speicher, die alle Operationen mit $O(\log_{m+1} n)$ Zeit erlaubt.