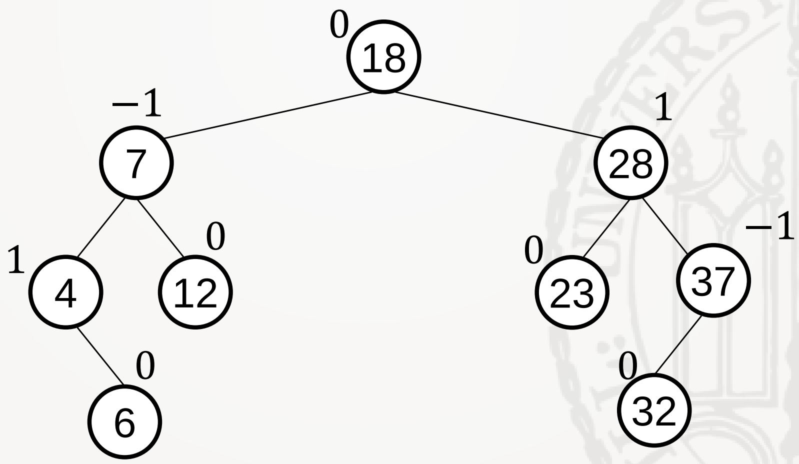
AVL-Bäume: Einfügen

- Zuerst normales Einfügen wie bei binären Bäumen.
- Beim Einfügen kann sich nur die Balance b von Knoten ändern, die auf dem Suchpfad liegen.
- Wird das AVL-Kriterium verletzt, gehe den Suchpfad zurück und aktualisiere die Balance.

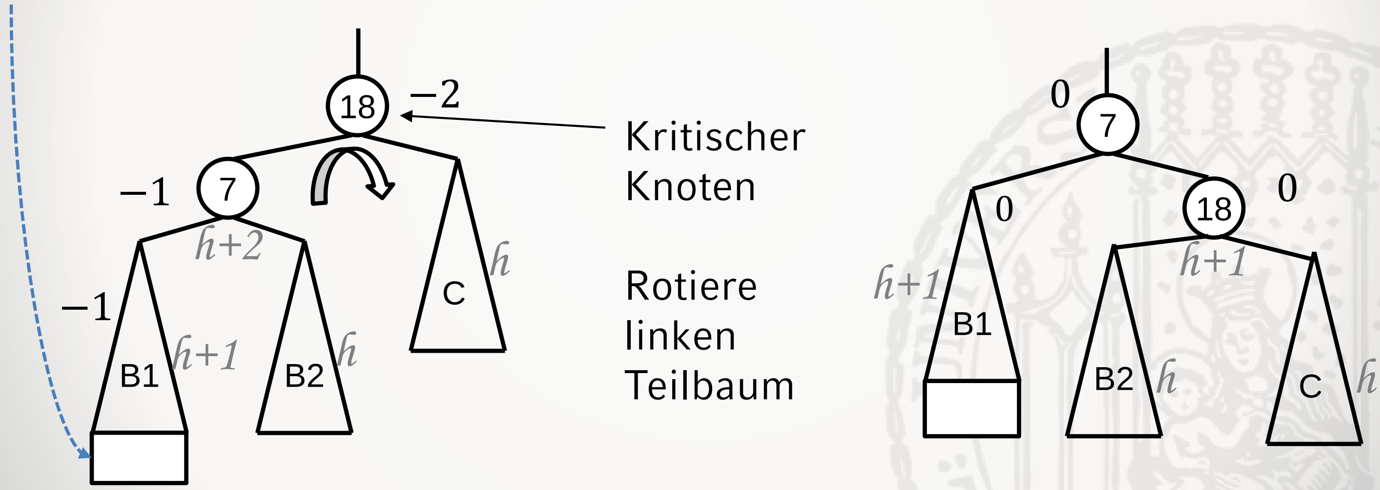


AVL-Bäume: Einfachrotation

- Rechtsrotation (rechts-rechts)

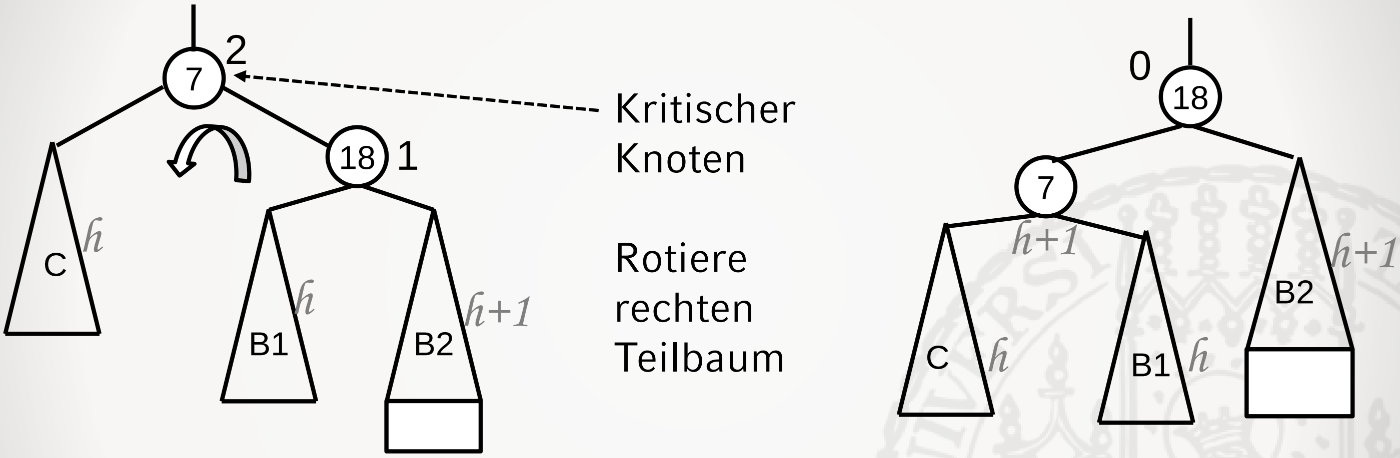
Beispiel: Einfügung war in Teilbaum „links links“ (Balance = −2)

Baum ist nach der Rotation wieder balanciert



AVL-Bäume: Einfachrotation

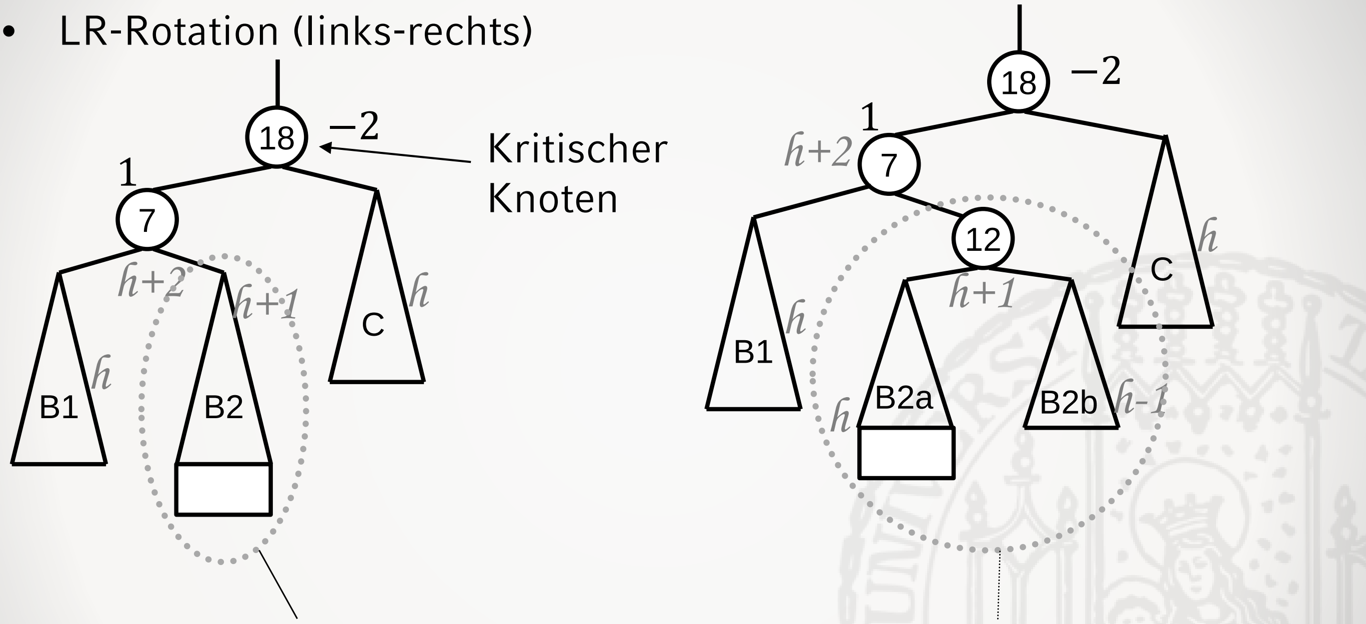
- Linksrotation (links-links)



Symmetrisch zur Rechtsrotation

AVL-Bäume: Doppelrotation

- LR-Rotation (links-rechts)

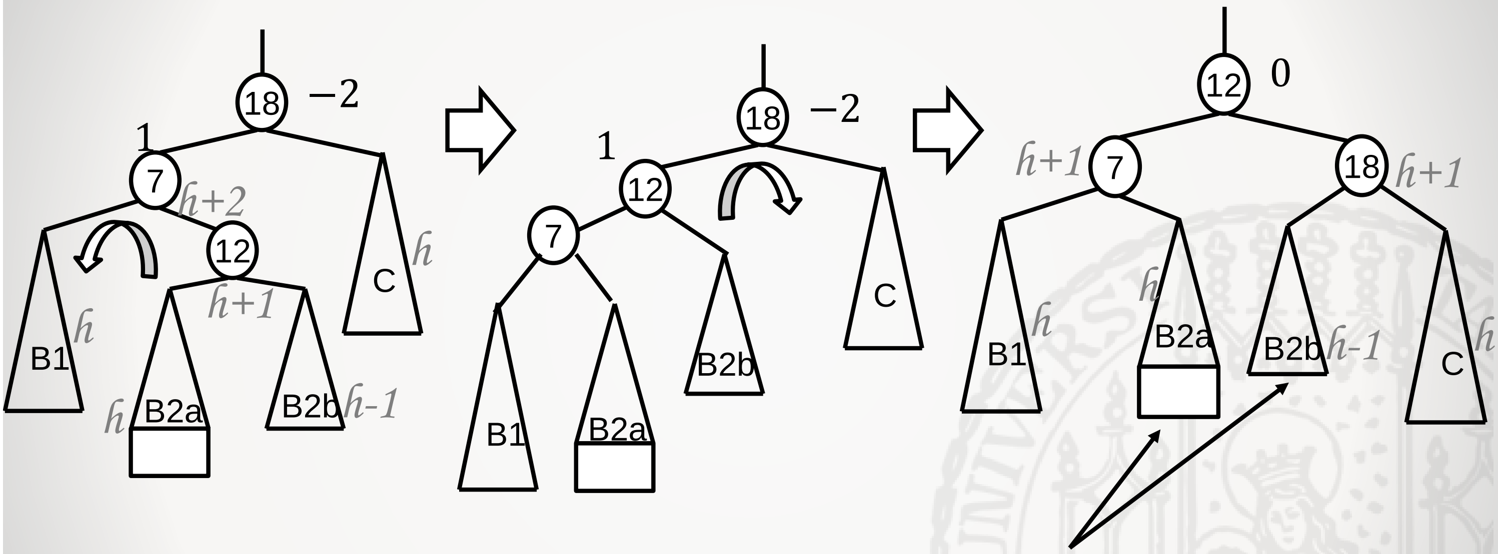


Eine einfache Rotation ist nicht mehr ausreichend, da der problematische Baum innen liegt

→ der Baum B2 muss näher betrachtet werden

AVL-Bäume: Doppelrotation

- LR-Rotation (links-rechts)



Wie man sieht, ist es dabei egal, ob der neue Knoten im Teilbaum B2a oder B2b eingefügt wurde

Die RL-Rotation geht analog zur LR-Rotation (symmetrischer Fall)

AVL-Bäume: Komplexität beim Einfügen

- Die Rotationen stellen das AVL-Kriterium im rebalancierten Unterbaum wieder her und sie bewahren die Sortierreihenfolge

- Wenn ein Baum rebalanciert wird, ist der entsprechende Unterbaum danach immer genauso hoch wie vor dem Einfügen.

  → der restliche Baum bleibt konstant und muss nicht überprüft werden

  → beim Einfügen eines Knotens benötigt man höchstens eine Rotation zur Rebalancierung.

Aufwand:

$$\begin{array}{cccccc} \text{Einfügen} & + & \text{Rotieren} & & \\ O(h) & + & O(1) & = & O(\log(n)) \end{array}$$

AVL-Bäume: Löschen

Vorgehensweise
- Zuerst „normales“ Löschen wie bei binären Bäumen

- Nur für Knoten auf diesem Pfad kann das AVL-Kriterium verletzt werden (wie beim Einfügen)

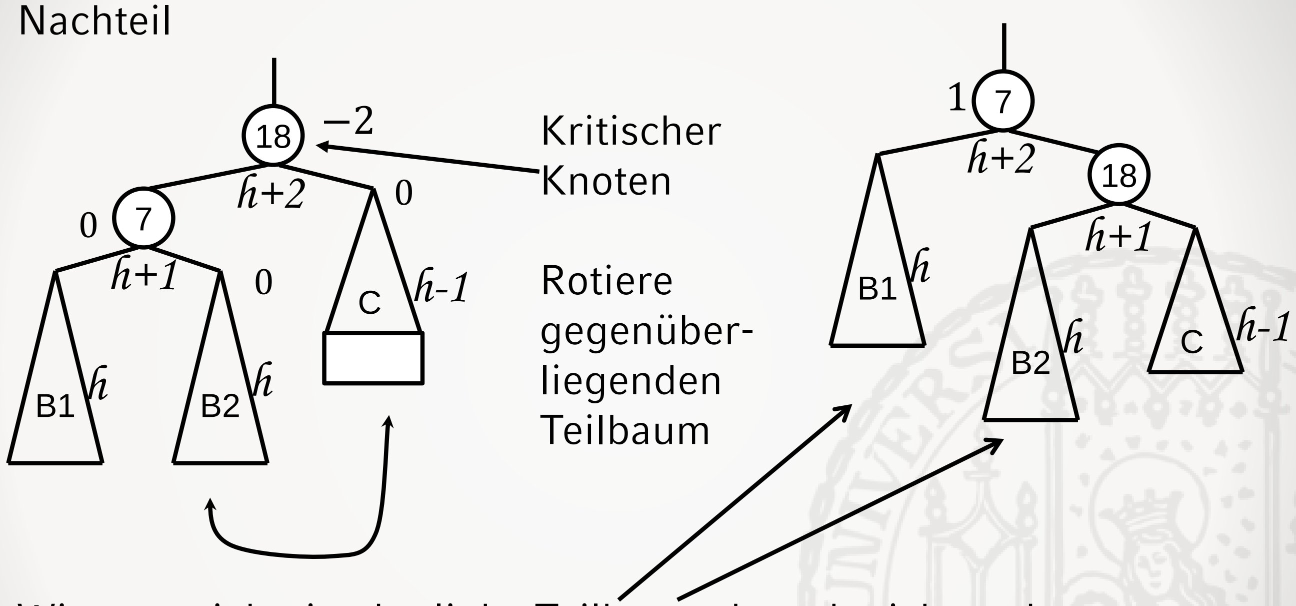
Ablauf:
- Nach dem „normalen“ Löschen den kritischen Knoten bestimmen (nächster Vorgänger zum tatsächlich entfernten Knoten mit Balance $b = \pm 2$)

- Dieser ist Ausgangspunkt der Reorganisation (hier Rotation genannt)

- Rotationstyp wird bestimmt, als ob im gegenüberliegenden Unterbaum ein Knoten eingefügt worden wäre

AVL-Bäume: Löschen

- Nachteil



- Wie man sieht, ist der linke Teilbaum danach nicht mehr vollkommen ausbalanciert
- D.h., AVL-Balance wird zum Teil durch Abnahme von vollkommenen Teilbaumbalancen erkauft.

AVL-Bäume: Komplexität beim Löschen

- Beim Löschen eines Knotens wird
  - das AVL-Kriterium wiederhergestellt, die Sortierreihenfolge bleibt erhalten
  - kann es vorkommen, dass der re-balancierte Unterbaum nicht die gleiche Höhe wie vor dem Löschen besitzt

  $\rightarrow$ auf dem weiteren Pfad zur Wurzel kann es zu weiteren Re-Balancierungen (des obigen Typs, also immer im anderen Unterbaum) kommen

  $\rightarrow$ beim Löschen werden maximal h Rotationen benötigt

Aufwand:

Entfernen	+	Rotieren	
$O(h)$	+	$O(h)$	$= O(\log(n))$