

Algorithmen und Datenstrukturen

Aufgabe 4

29

Problemstellung

- ◆ Gegeben sei eine Menge von Objekten (Schlüsseln), auf die mit
 - zeitlich veränderlichen
 - unbekanntenHäufigkeiten zugegriffen wird.
- ◆ Problem: Finde eine Speicherungsform, die die Zugriffskosten minimiert!
- ◆ Lösungsidee: Speichere Schlüssel in eine sich selbst anordnenden
 - linearen Liste! (Sequentiell gespeichert)
 - sortiertem Baum (nicht sequentiell gespeichert)

30

Selbstanordnende lineare Listen

- ♦ **MF-Regel (Move-to-front):** Mache ein Element zum ersten Element der Liste, nachdem auf das Element (als Ergebnis einer erfolgreichen Suche) zugegriffen wurde.
- ♦ **T-Regel (Transpose):** Vertausche ein Element mit dem unmittelbar vorangehenden, nachdem auf das Element zugegriffen wurde.
- ♦ **FC-Regel (Frequency Count):** Nach jedem Zugriff auf ein Element wird dessen Häufigkeitszähler um 1 erhöht. Ferner wird die Liste nach jedem Zugriff neu geordnet und zwar so, daß die Häufigkeitszähler der Elemente in absteigender Reihenfolge sind.

31

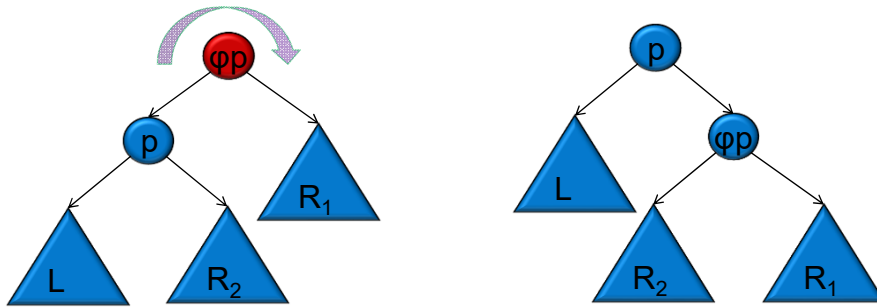
Splay-Bäume

- ♦ **Grundidee:** Move-to-root Strategie
nach jedem Zugriff wird ein Knoten durch Rotationen in Richtung der Wurzel bewegt. (Splay: engl. Verbreitern, spreizen).
- ♦ **Schritt 1:** Suche nach Element x im Baum T . Sei p der Knoten, bei dem die erfolgreiche Suche endet, falls x in T vorkommt und sei p der Vater des Blattes, bei dem eine erfolglose Suche nach x in T endet, sonst.
- ♦ **Schritt 2:** Wiederhole die folgenden Operationen zig, zig-zig und zig-zag beginnend bei p solange, bis sie nicht mehr ausführbar sind, weil p Wurzel geworden ist.

32

Fall 1: zig

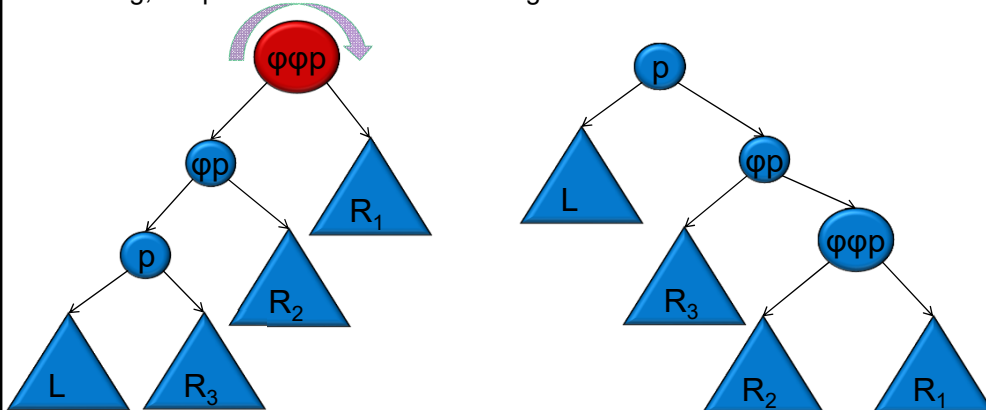
p hat Vater φp und φp ist Wurzel:
Führe Operation zig aus: Links bzw. Rechtsrotation um φp .



33

Fall 2: zig-zig

p hat Vater φp und Großvater $\varphi\varphi p$ und p und φp sind beides rechte oder beides linke Söhne:
Führe Operation zig-zig aus: Zwei aufeinanderfolgende Rotationen in dieselbe Richtung, die p zwei Niveaus hinaufbewegen.

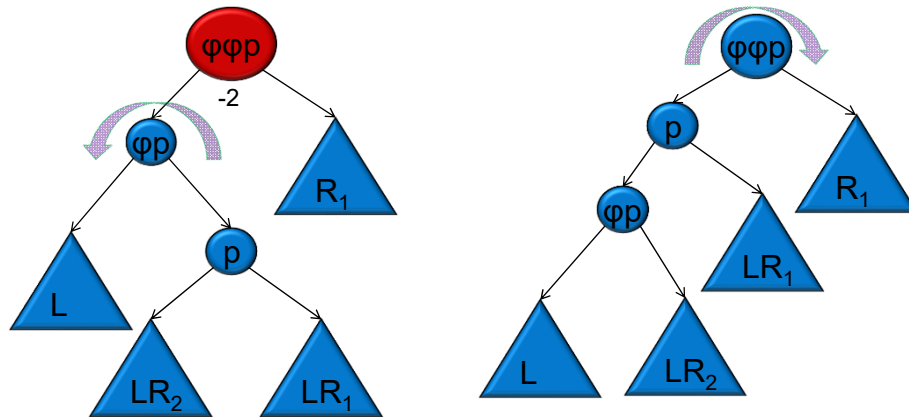


34

Fall 3: zig-zag

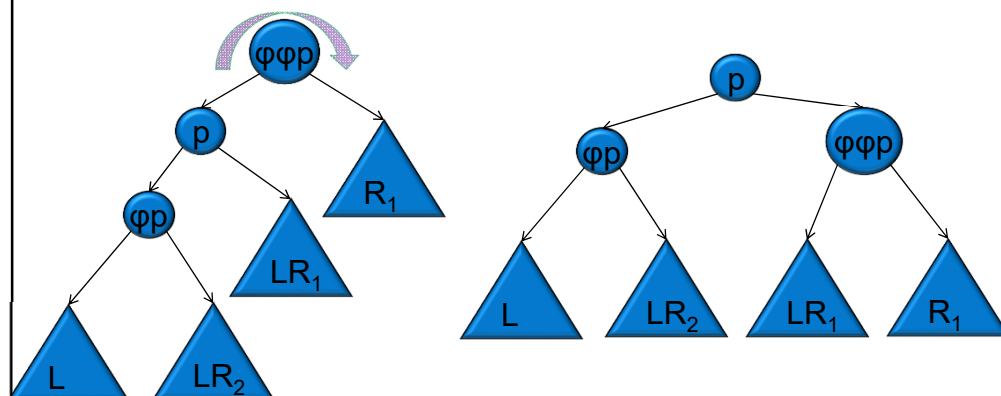
p hat Vater φp und Großvater $\varphi\varphi p$ und einer der beiden Knoten p und φp ist linker und der andere rechter Sohn:

Führe Operation zig-zag aus: zwei Rotationen in entgegengesetzte Richtungen, die p zwei Niveaus hinaufbewegen.



35

Fall 3: zig-zag



36

Transpose-Bäume

- ♦ **Grundidee:** Transpose Strategie
nach jedem Zugriff wird ein Knoten durch Rotationen in Richtung der Wurzel bewegt.
- ♦ **Schritt 1:** Suche nach Element x im Baum T . Sei p der Knoten, bei dem die erfolgreiche Suche endet, falls x in T vorkommt und sei p der Vater des Blattes, bei dem eine erfolglose Suche nach x in T endet, sonst.
- ♦ **Schritt 2:** Rotiere um den Vater ϕp von p so, dass p ein Niveaus hinaufbewegt wird. (Links- bzw. Rechtsrotation bzw. zig Operation)