

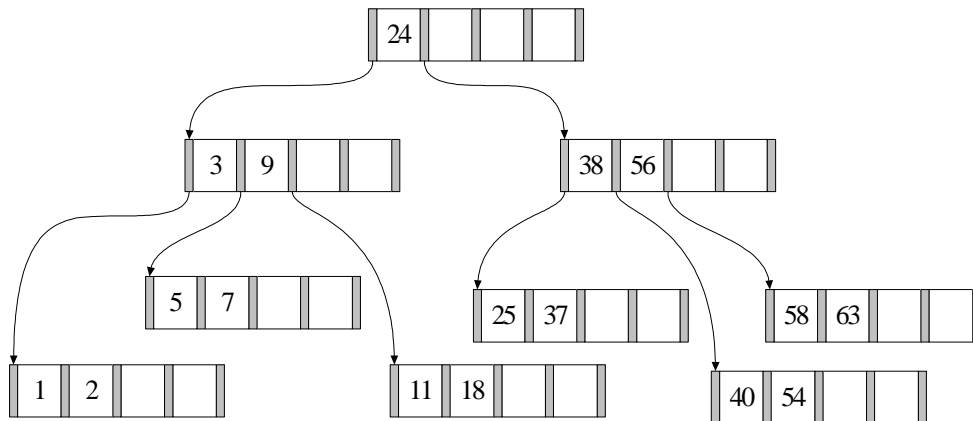


**Übung zur Vorlesung Grundlagen: Datenbanken im WS23/24**  
Christoph Anneser, Michael Jungmair, Stefan Lehner, Moritz Sichert, Lukas Vogel  
(gdb@in.tum.de)  
<https://db.in.tum.de/teaching/ws2324/grundlagen/>

**Blatt Nr. 10**

**Hausaufgabe 1**

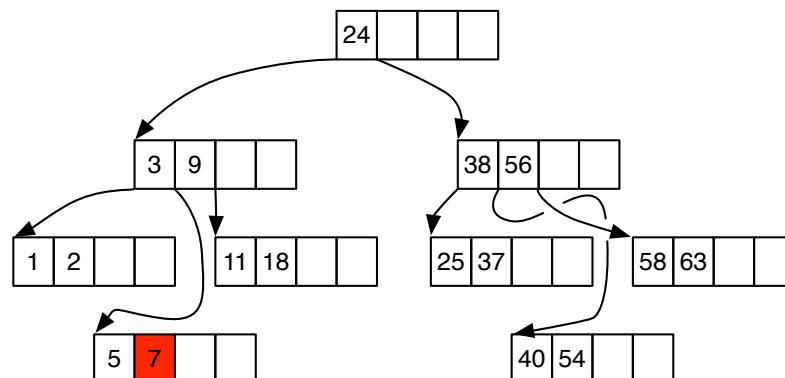
Folgende Ausprägung eines B-Baums mit  $k = 2$  sei gegeben:



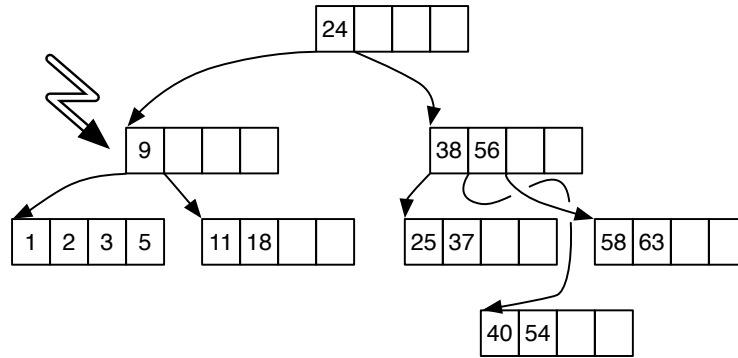
Geben Sie den Baum nach Löschen der 7 an.

**Lösung:**

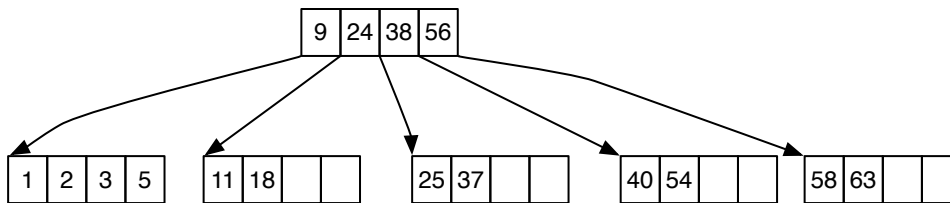
Original, 7 soll gelöscht werden.



Nach dem Entfernen der 7 und dem Merge mit ihrem Nachbarknoten. Anderer Knoten nun zu gering belegt.



Nach dem Merge mit dem Nachbarknoten und dem daraus resultierenden Wegfall der Wurzel.



## Hausaufgabe 2

Fügen Sie in einen anfangs leeren R-Baum mit Knotenkapazität 4 folgende Datenpunkte nacheinander ein:

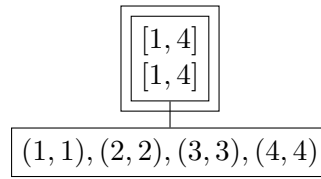
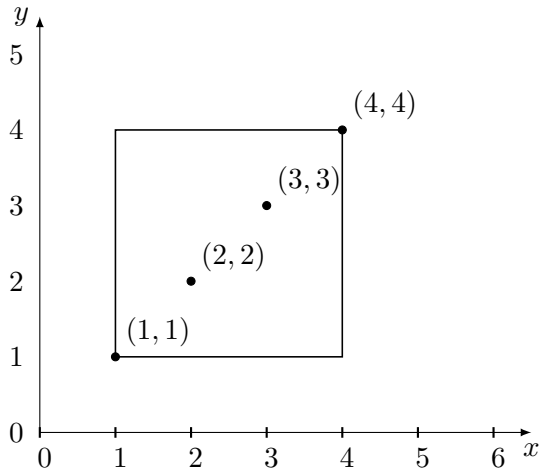
$$(1, 1), (2, 2), (3, 3), (4, 4), (5, 3), (5, 4), (6, 5), (4, 2)$$

Splitten Sie die Knoten dabei so, dass die summierte Fläche der durch den Split entstandenen Boxen möglichst klein ist.

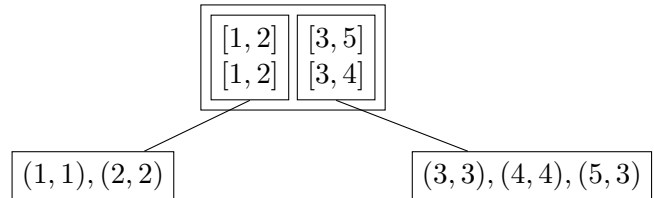
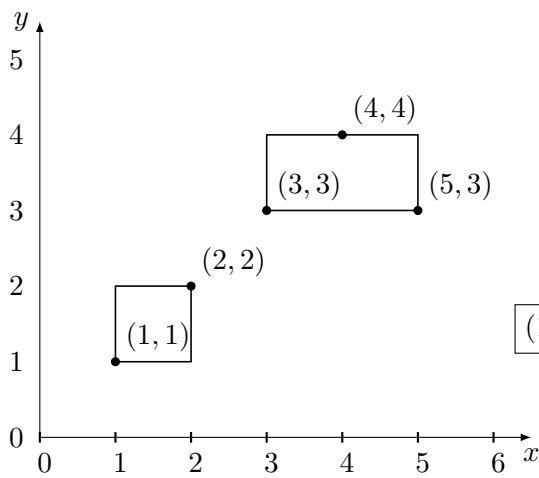
Illustrieren Sie die einzelnen Phasen im Aufbau des R-Baums. Zeichnen Sie hierzu den Baum und den Datenraum unmittelbar vor jedem Split und im Endzustand.

### Lösung:

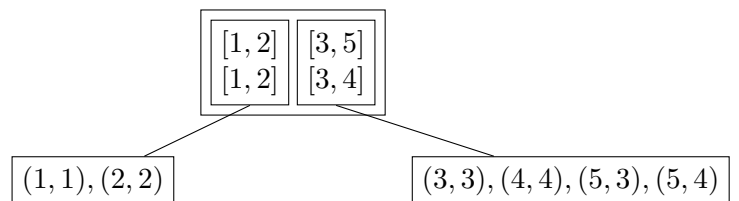
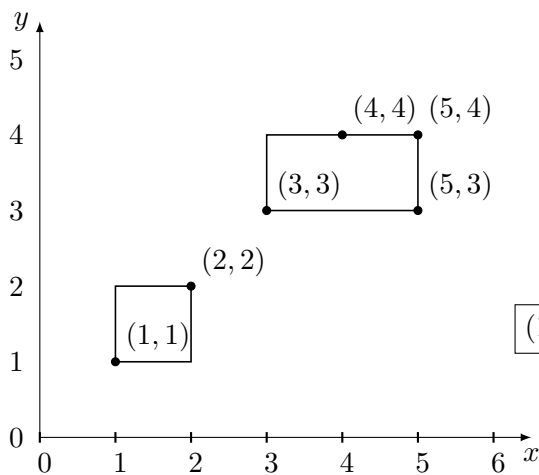
Wir fügen die Werte  $(1, 1)$ ,  $(2, 2)$ ,  $(3, 3)$  und  $(4, 4)$  in den ersten Knoten ein.



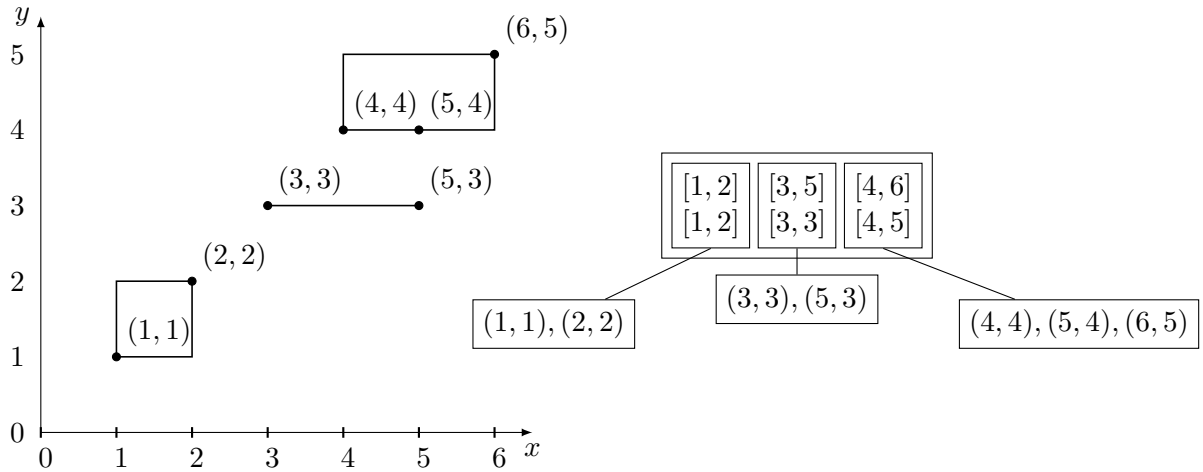
Da der Knoten nun voll ist, müssen wir ihn für den Wert  $(5, 3)$  aufspalten. Um die Gesamtfläche gering zu halten, fügen wir den neuen Wert dann in das zweite Blatt ein:



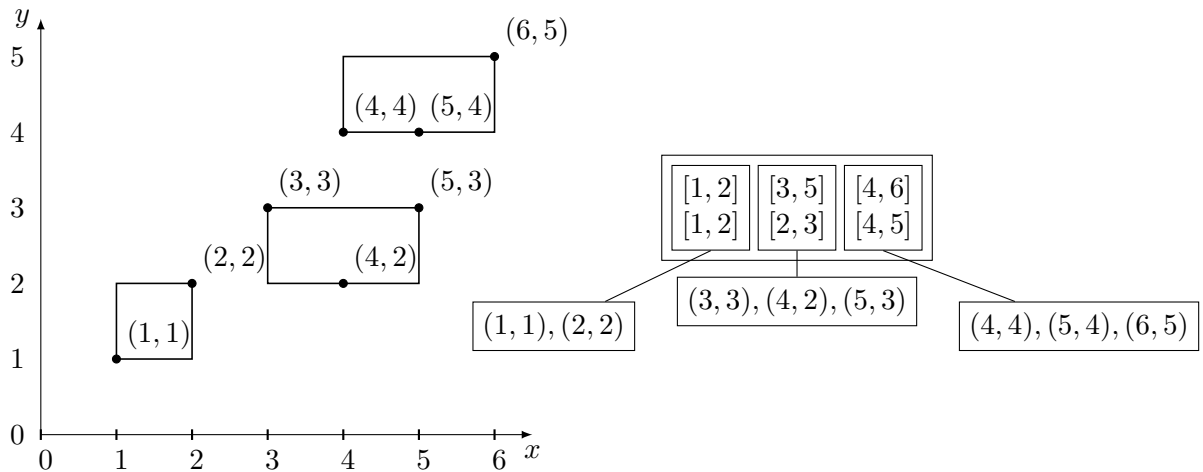
In dieses Blatt passt auch noch der Wert  $(5, 4)$ :



Vor Einfügen des Wertes  $(6, 5)$  müssen wir nun den zweiten Knoten splitten. Um die Größe nach dem Split minimal zu halten, ordnen wir die Werte  $(3, 3)$  und  $(5, 3)$ , sowie  $(4, 4)$  und  $(5, 4)$  jeweils einer Box mit der Fläche 0 zu und fügen schließlich  $(6, 5)$  in die obere Box ein.



Für den Wert  $(4, 2)$  gibt es zwei Möglichkeiten: Wir können den Wert entweder in die zweite Box mit der Größe 0, oder in die erste Box mit der Größe 1 einfügen. In beiden Fällen vergrößert sich die Gesamtfläche um 2. Wir entscheiden uns hier für die erste Option, die Alternative ist aber genauso richtig.



**NB:** Bei R-Bäumen ist es nicht immer eindeutig, wo neue Werte eingefügt werden müssen und wie man Knoten aufspaltet!

### Hausaufgabe 3

Fügen Sie die folgenden Tupel in eine anfangs leere erweiterbare Hashtabelle, welche 2 Einträge pro Bucket aufnehmen kann, ein. Dabei soll die Matrikelnummer als Suchschlüssel verwendet werden.

MatrNr	Name
2	Müller
8	Schmidt
19	Fischer
16	Huber
20	Bauer
34	Schneider
30	Wagner

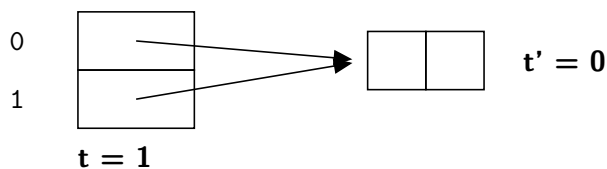
Verwenden Sie als Hashverfahren die inverse binäre Repräsentation der Matrikelnummer, wie in der Vorlesung beschrieben.

**Lösung:**

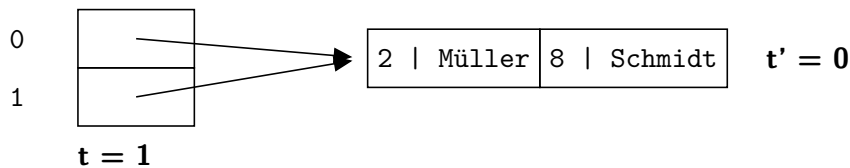
Zunächst errechnen wir den Hashwert für alle Studierende: Die inverse binäre Repräsentation der Matrikelnummer.

MatrNr	Name	Binär	Invers Binär
2	Müller	000010	010000
8	Schmidt	001000	000100
19	Fischer	010011	110010
16	Huber	010000	000010
20	Bauer	010100	001010
34	Schneider	100010	010001
30	Wagner	011110	011110

Zunächst eine leere erweiterbare Hashtabelle:



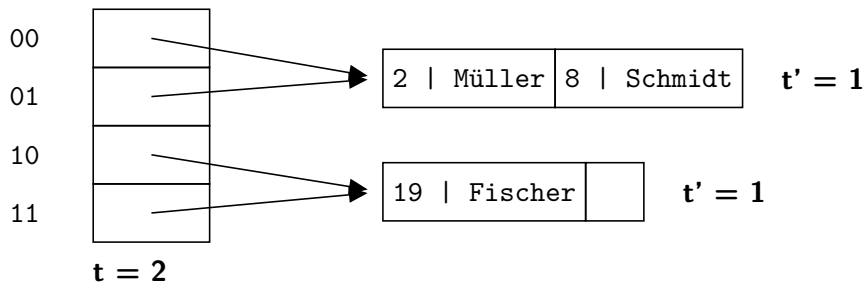
Wir fügen nun die ersten zwei Einträge ein, wonach die Hashtabelle wie folgt aussieht:



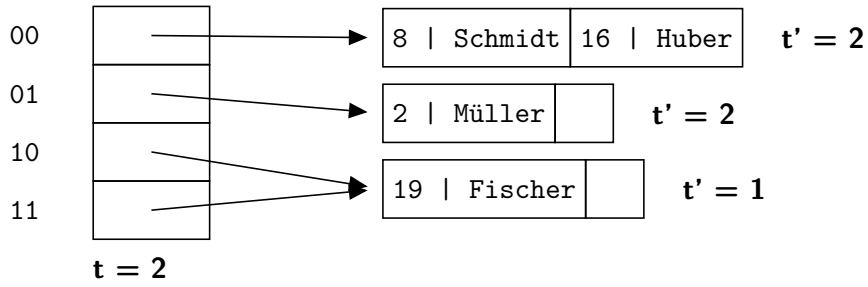
Der nächste Eintrag führt zu einem Überlauf. Da  $t' < t$ , können wir das Bucket teilen. Wir erhalten somit folgende Hashtabelle:



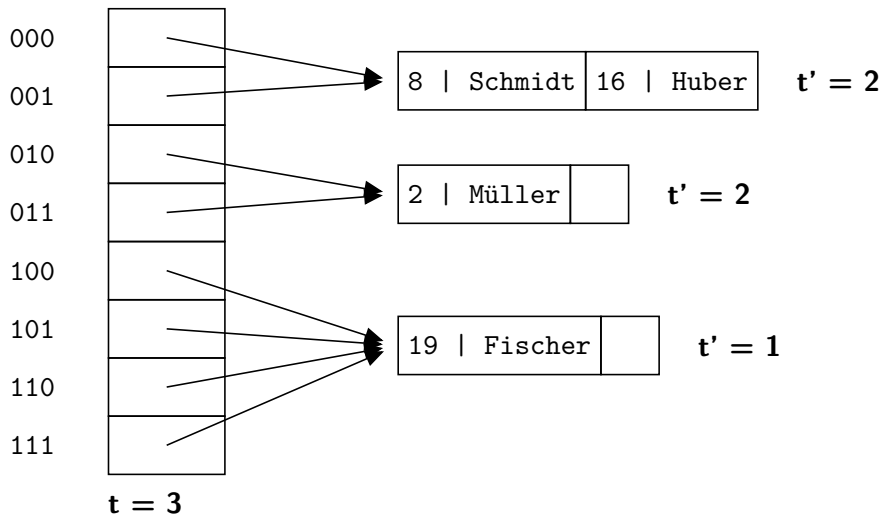
Der nächste Eintrag führt zu einem erneuten Überlauf! Leider in dem Bucket, für das  $t = t'$  gilt. Wir müssen also die globale Tiefe erhöhen und erhalten somit folgende Hashtabelle:



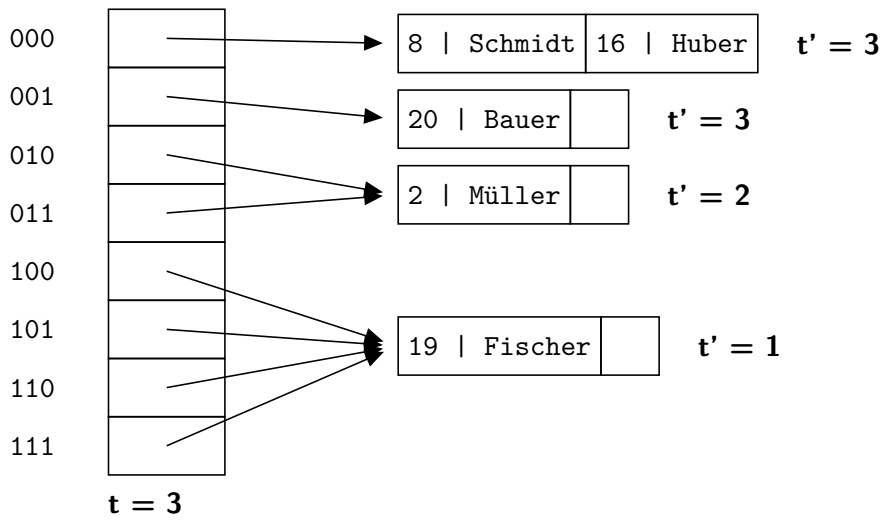
Nun können wir das erste Bucket splitten und unseren neuen Wert eintragen:



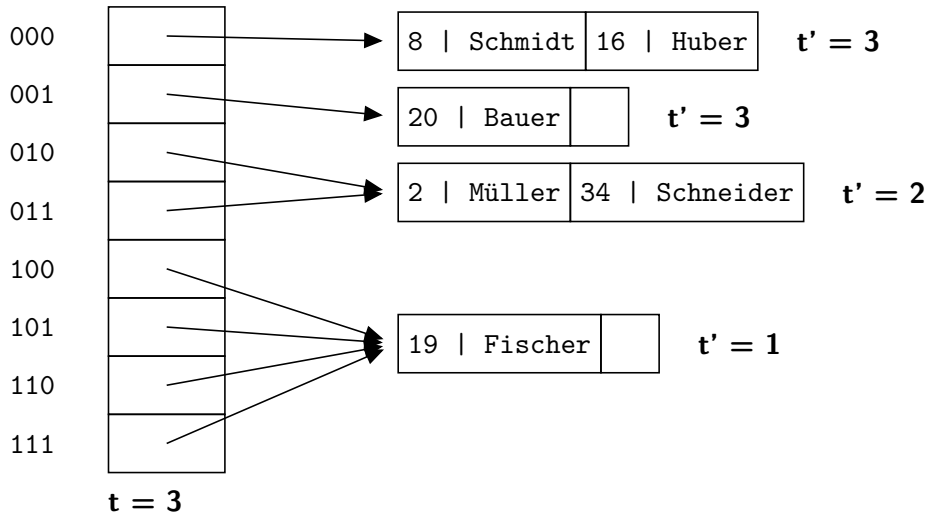
Bauer muss dem obersten Bucket hinzugefügt werden. Da dort kein Platz ist und  $t = t'$  gilt, müssen wir erneut zuerst die globale Tiefe erhöhen.



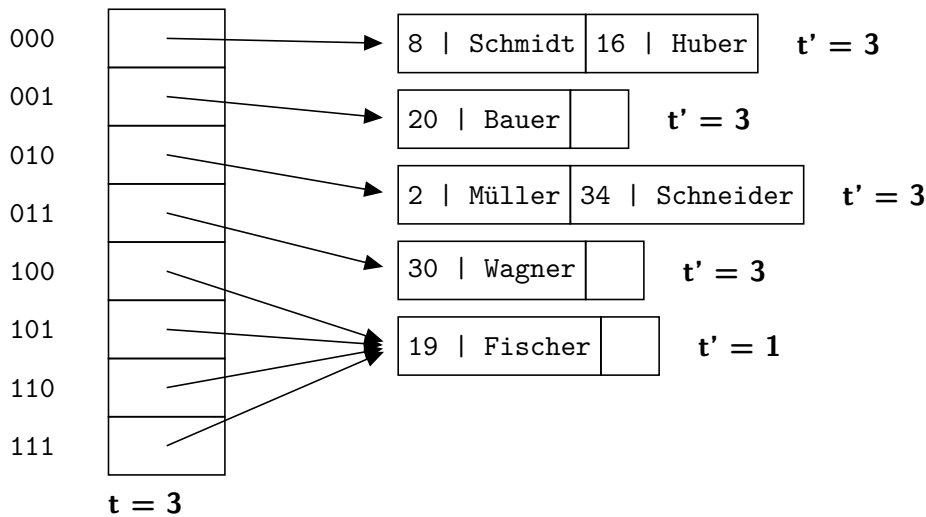
Wir können nun das oberste Bucket splitten, und Bauer hinzufügen:



Schneider passt in ein halbvolleres Bucket:



Für Wagner müssen wir zuletzt noch einmal die lokale Tiefe erhöhen.



#### Hausaufgabe 4

Gegeben sei eine erweiterbare Hashtabelle mit globaler Tiefe  $t$ . Wie viele Verweise zeigen vom Verzeichnis auf einen Behälter mit lokaler Tiefe  $t'$ ?

#### Lösung:

In dem Verzeichnis einer Hashtabelle mit globaler Tiefe  $t$  werden  $t$  Bits eines Hashwerts für die Identifizierung eines Verzeichniseintrags verwendet. Für einen Behälter mit lokaler Tiefe  $t'$  sind hingegen nur die ersten  $t'$  Bits dieses Bitmusters relevant.

Mit anderen Worten bedeutet dies, dass alle Einträge, die einen Behälter mit lokaler Tiefe  $t'$  referenzieren, in den ersten  $t'$  Bits übereinstimmen. Da alle Bitmuster bis zur Länge  $t$  in dem Directory aufgeführt sind, unterscheiden sich diese Einträge in den letzten  $t - t'$  Bits.

⇒ Es gibt somit  $2^{t-t'}$  Einträge im Verzeichnis, die auf denselben Behälter mit lokaler Tiefe  $t'$  verweisen.

#### Hausaufgabe 5

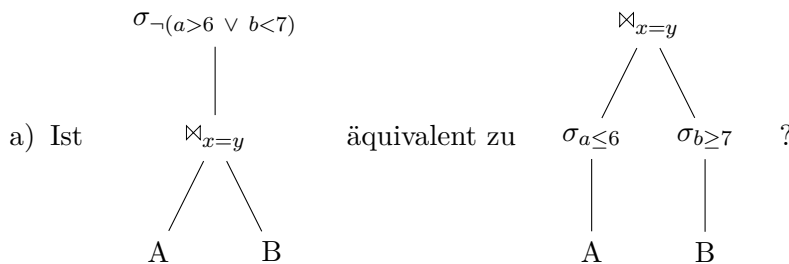
Gegeben seien die Relationen

$$A : \{[a, x]\}$$

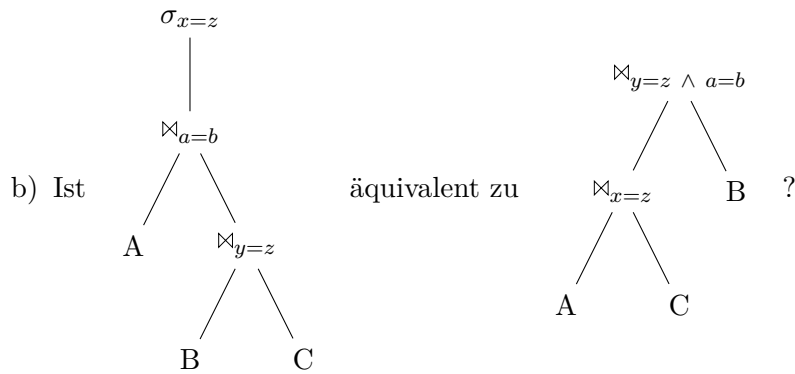
$$B : \{[b, y]\}$$

$$C : \{[c, z]\}$$

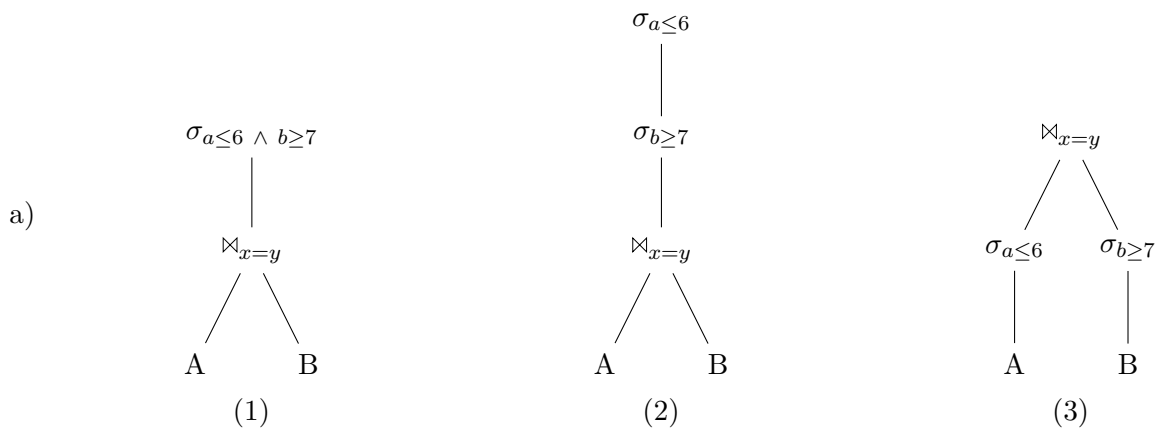
Im Folgenden sehen Sie jeweils zwei Operatorbäume der relationen Algebra. Sind diese äquivalent zueinander? Beweisen oder widerlegen Sie mithilfe der zwölf äquivalenzerhaltenden Transformationsregeln aus der Vorlesung.



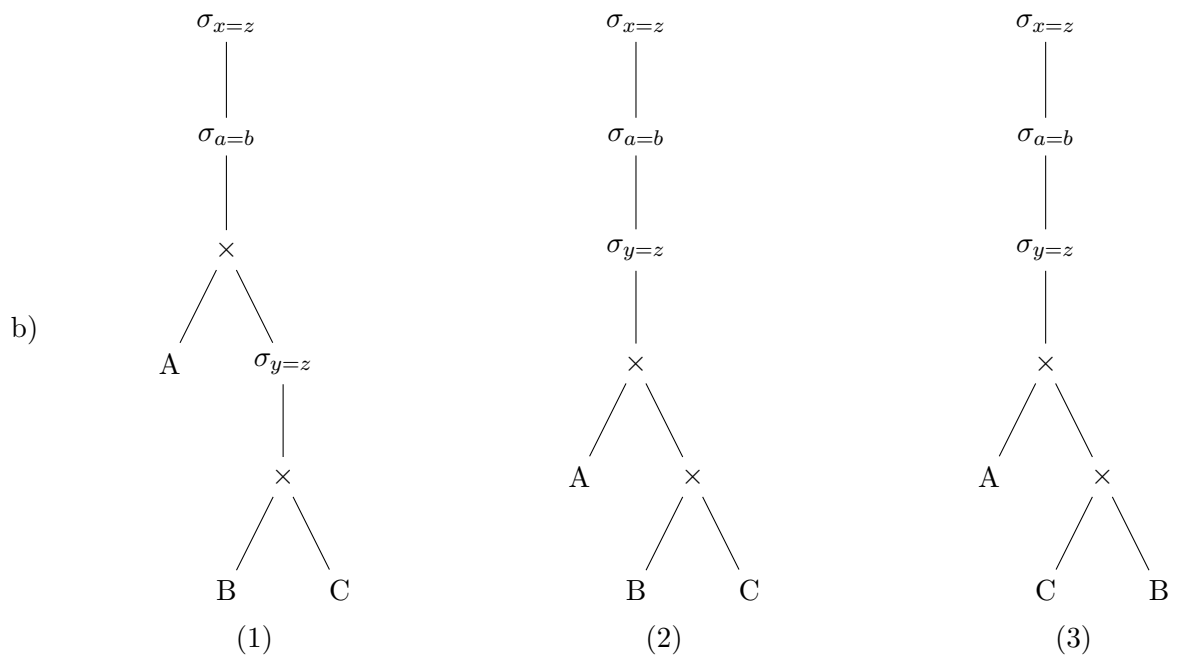


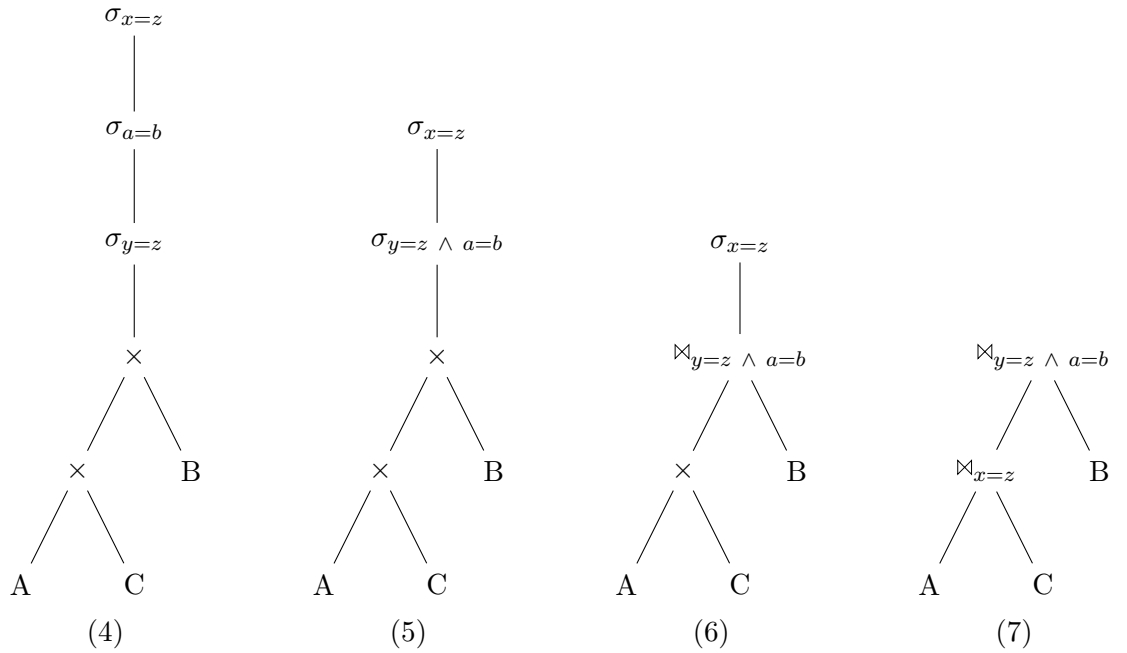


**Lösung:**



Das DeMorgansche Gesetz (Regel 11 im Foliensatz) erlaubt uns, die Disjunktion in der Selektion in eine Konjunktion umzuformen (1). Diese Konjunktion können wir nun nach Regel 1 im Foliensatz in zwei getrennte Selektionen aufbrechen (2). Diese Selektionen können wir nun nach unten propagieren (Regel 6 im Foliensatz), da  $a$  ein Attribut von  $A$  und  $b$  ein Attribut von  $B$  ist (3). Da wir lediglich Äquivalenzumformungen verwendet haben, ist dieser Operatorbaum äquivalent zum Ausgangsbaum.





Nach Regel 12 können wir einen Join in ein kartesisches Produkt und eine Selektion transformieren (1). Regel 6 erlaubt es uns dann, alle Selektionen nach oben zu propagieren (2). Wir können nun erst die Kommutativität (Regel 5) (3) und dann die Assoziativität (Regel 8) (4) des Kreuzproduktes verwenden. Nach Regel 1 können wir zwei Selektionen zusammenführen (5) und nach Regel 12 erneut in einen Join umwandeln (6). Ein erneutes Anwenden von Regeln 6 und 12 führt uns zum Ziel (7). Auch diese beiden Operatorbäume sind äquivalent.