

## 6. Übungsblatt

Ausgabe: 21. November 2016 · Besprechung: Ab 28. November 2016

### 1 Dynamic Programming

Gegeben sei die folgende Anfrage:

```
SELECT A.name, F.airline, C.name
FROM Airport AS A, Flight AS F, Crew AS C
WHERE F.to = A.code
AND F.flightNo = C.flightNo
```

Auf dem Attribut *code* der Tabelle *Airport* existiere ein B-Baum. Zum Optimieren der Anfrage verwende das System den *Dynamic Programming*-Algorithmus wie in der Vorlesung besprochen. Dabei berücksichtige der Optimierer auch “interesting orders”. Das System verfüge ausschließlich über die Joinalgorithmen *Index Nested Loops Join* und *Block Nested Loops Join*, wobei wir annehmen, dass ersterer immer die günstigere Alternative sei.

Zeigen Sie, welche (Teil-)Pläne der *Dynamic Programming*-Optimierer in den einzelnen Optimierungphasen berücksichtigt, welche er verwirft und welche er beibehält.

### 2 Transaktionskontrolle

Gegeben sei der folgende Schedule:

$$S = \langle r_3(x), r_1(x), r_2(x), w_1(y), r_2(x), w_3(x) \rangle .$$

1. Zeichnen Sie den **Serialisierbarkeitsgraphen** für  $S$ .
2. Ist  $S$  **konflikt-serialisierbar**? Begründen Sie Ihre Antwort.
3. Könnte dieser Schedule erzeugt worden sein durch einen Scheduler, der **2-Phase-Locking** verwendet? Falls ja, zeigen Sie das durch Eintragen von *lock/unlock*-Aufrufen in den Schedule gemäß den 2PL-Protokoll. Falls nein, begründen Sie Ihre Antwort.
4. Zeigen oder widerlegen Sie:

Die Menge aller konflikt-serialisierbaren Schedules ist identisch mit der der aller Schedules, die von einem 2PL-Scheduler erzeugt werden können.