B561 – Selected Solutions for Assignment 5

Dirk Van Gucht & Mathias Niepert

November 24, 2007

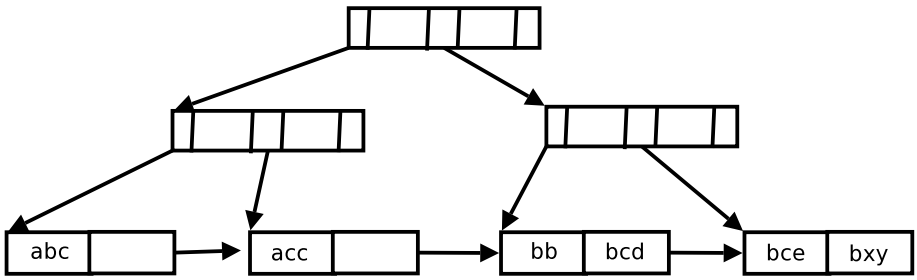

Figure 1: B+-tree

1. Consider the partially specified B+ tree in Figure 1.
   (a) See Figure 2.
   (b) See Figure 3.
   (c) See Figure 4.
2. See Figure 5.
3. (1) (a) Match, Sailors.sid < 50000.
      (a) Match, Sailors.sid $= 50000$.
   (2) (a) No Match.
      (b) Match, Sailors.sid $= 50000$.
   (3) (a) Match, Sailors.sid $< 50000 \wedge$ Sailors.age $= 21$.
      (b) Match, Sailors.sid=50000 $\wedge$ Sailors.age>21.
      (c) Match, Sailors.sid $= 50000$.
      (d) No match.

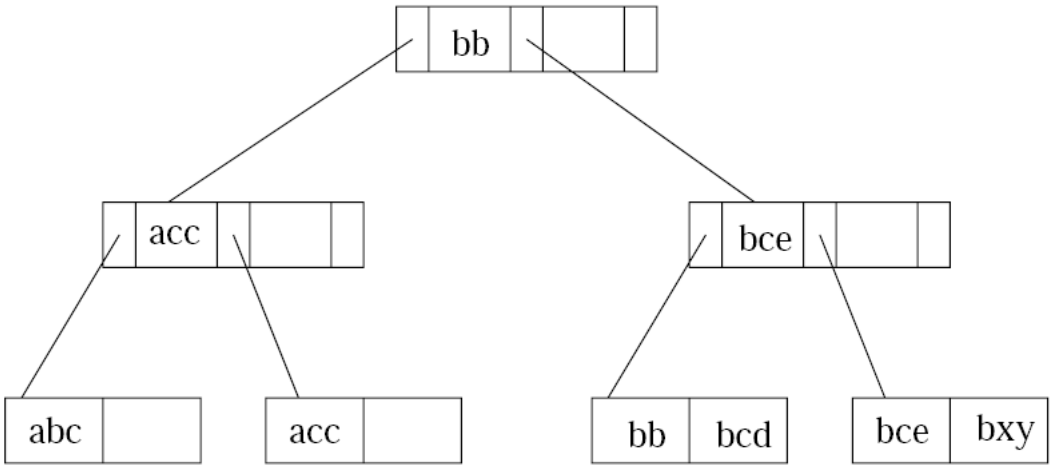

Figure 2: Solution to problem 1(a)

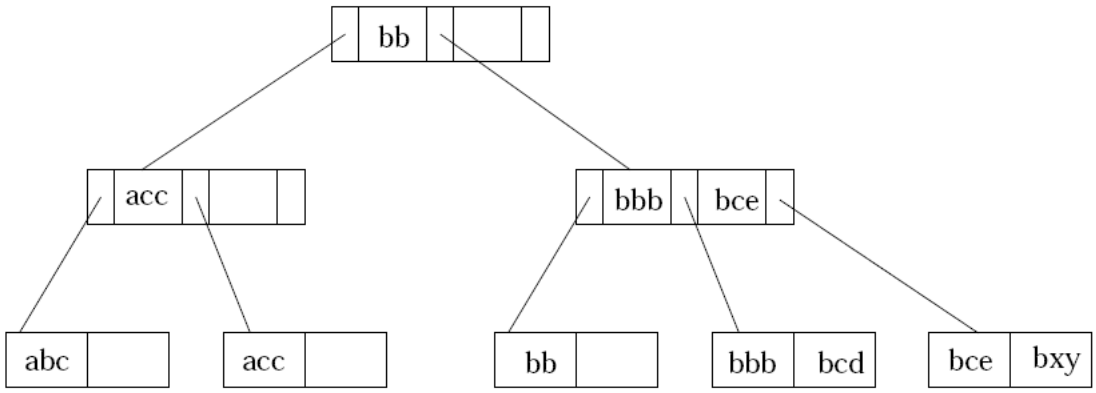

Figure 3: Solution to problem 1(b)

(4) This question can be understood in two ways:
(i) The textbook has a typo and there is only a hash-index on $\langle Sailors.id, Sailors.age \rangle$:
(a) Match, Sailors.sid = 50000 $\wedge$ Sailors.age = 21
(b) No match.

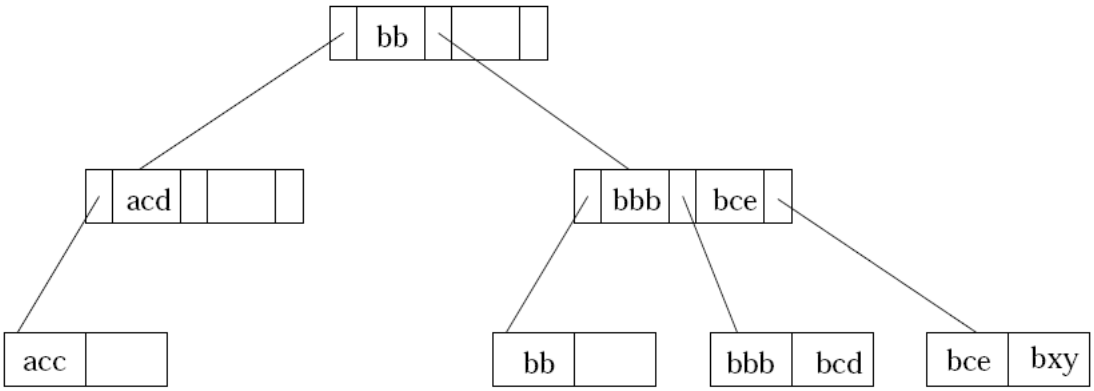

Figure 4: Solution to problem 1(c)

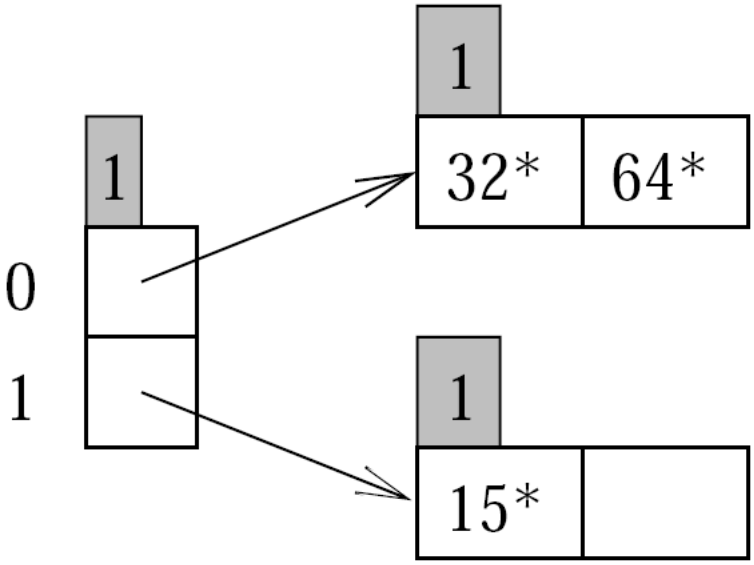

Figure 5: Possible solution to problem 2. Insertion of 128* causes recursive split of order 4.

(c) No match.

(d) No match.

(ii) There is a hash **and** a B+-tree index on $\langle Sailors.id, Sailors.age \rangle$:

(a) Match, Sailors.sid = 50000 $\wedge$ Sailors.age = 21 (Hash and B+-tree)

(b) Match, Sailors.sid = 50000 $\wedge$ Sailors.age > 21 (B+-tree)

(c) Match. Sailors.sid = 50000 (B+-tree)

(d) No match.

4. Consider the following SQL query

```
SELECT ROADID
FROM ROADS R, ZONES Z1, ZONES Z2
WHERE R.SRCZONE = Z1.ZONEID AND R.ENDZONE = Z2.ZONEID AND
      Z1.TYPE = 'R' AND Z2.TYPE = 'C' AND R.DIST < 10
```

(a) Please note the linebreak after ...$(ZONES)\times$!

$$\Pi_{ROADID}(\sigma_{\substack{SRCZONE=ZONEID_1 \wedge \\ ENDZONE=ZONEID_2 \\ TYPE_1='R' \wedge TYPE_2='C' \wedge DIST<10}}(ROADS \times \rho_{X \to X_1}(ZONES) \times \rho_{X \to X_2}(ZONES)))$$

(b) Our assumption is that the ZONE table is much smaller than the ROADS table. The query tree of the naive evaluation is depicted in Figure 6. In Figure 7 one can see the tree after the selections have been pushed down, and in Figure 8 after the Cartesian products have been rewritten as joins.

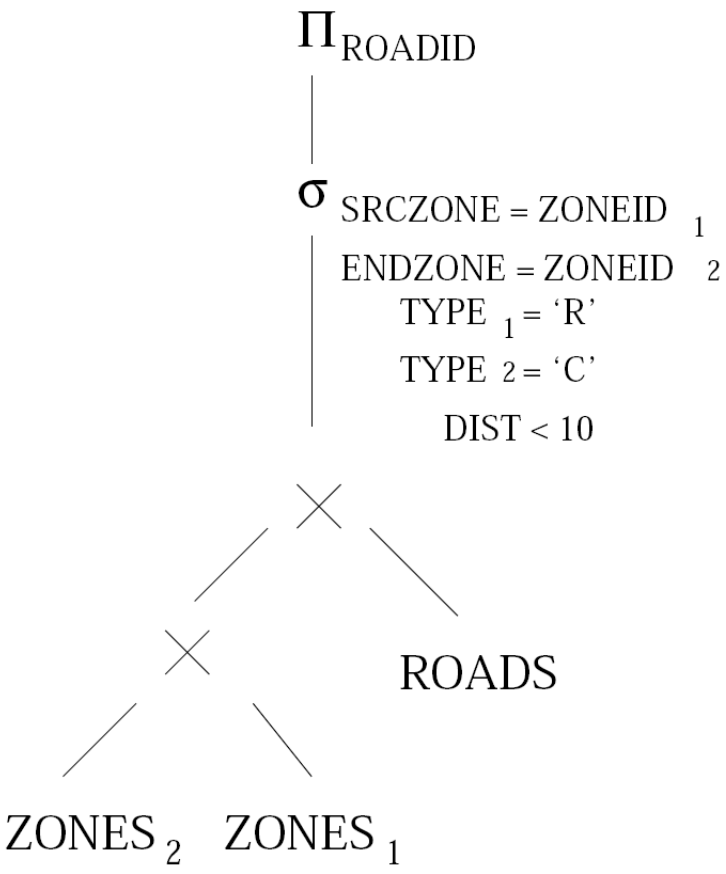

Figure 6: Query tree of naive evaluation.

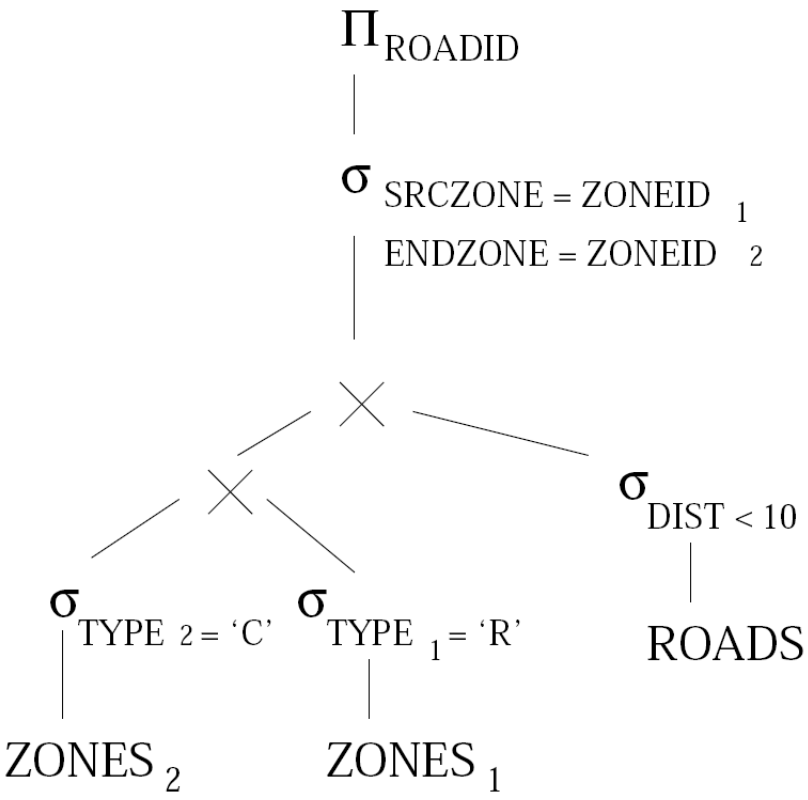

Figure 7: Query tree after the selections have been pushed down.

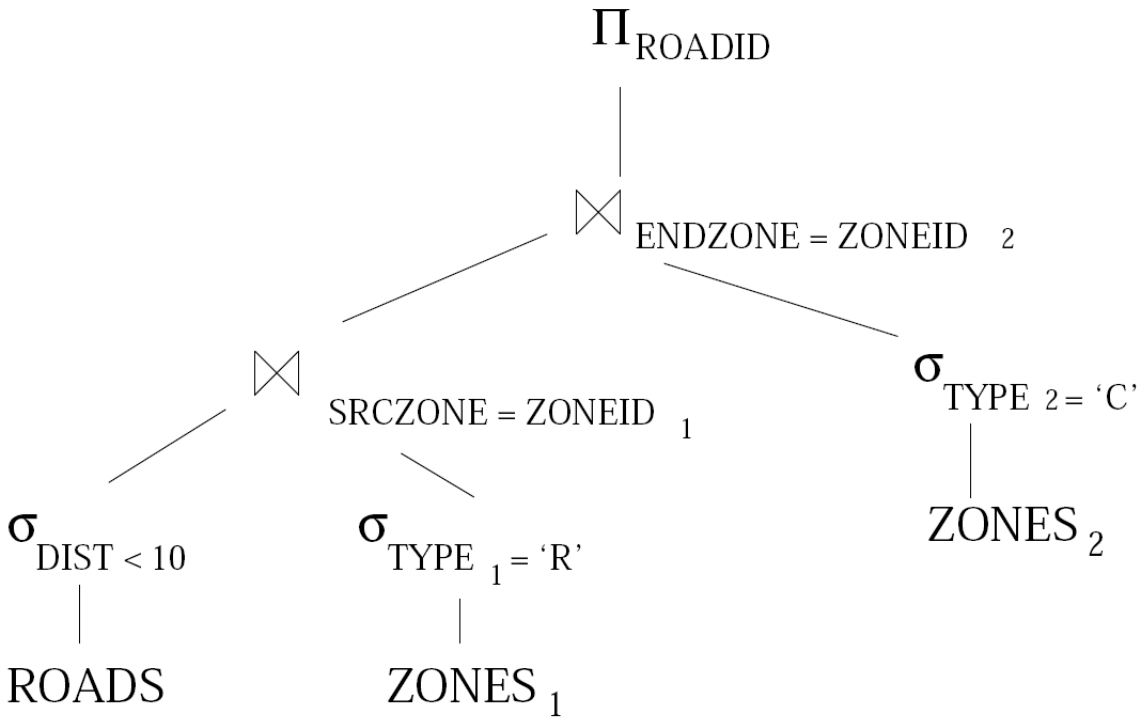

Figure 8: Query tree after making Cartesian products into joins.