

Optimizacija poizvedb

Hevristični postopek in primeri

Osnovne tehnike optimizacije proizvodb

- **Hevristična pravila** za pravilno razvrščanje operacij proizvodbe.
- **Primerjava relativnih stroškov različnih strategij**, kjer izberemo najcenejšo, t.j. predstavlja najmanjšo porabo sredstev.

Procesiranje poizvedbe ⁽¹⁾

- **Procesiranje poizvedbe** sestavlja:
 1. **Razčlenjevanje** oz. **dekompozicija**,
 - Cilj je pretvoriti poizvedbo iz visoko nivojskega jezika (SQL) v relacijsko algebro.
 2. **Optimizacija**,
 - Izbira najučinkovitejše izvedbene strategije.
 3. **Generiranje kode**,
 4. **Izvedba**.

Procesiranje proizvodbe (2)

Dekompozicija

- **Procesiranje proizvodbe** sestavlja:
 1. **Razčlenjevanje** oz. **dekompozicija**,
 - Analiza,
 - Normalizacija,
 - Semantična analiza,
 - Poenostavitev,
 - Reorganizacija proizvodbe.

Procesiranje proizvodbe (3)

Dekompozicija » Analiza (1)

- Analiza zajema **leksikalno** in **sintaktično analizo**.
- Dodatno je potrebno preveriti:
 - Ali so relacije in atributi zapisani v sistemskem katalogu?
 - Ali so operacije proizvodbe primerne glede na tip objekta, nad katerim se izvajajo?

Procesiranje poizvedbe (4)

Dekompozicija » Analiza (2)

- Navadno po opravljeni analizi poizvedbo pretvorimo v **drevo poizvedbe** za lažjo nadaljnjo obravnavo:
 - Osnovna relacija postane list drevesa.
 - Vmesna relacija, ki je rezultat operacij algebre, postane vozlišče drevesa.
 - Rezultat poizvedbe je v korenu drevesa.
 - Zaporedje operacij poteka od listov drevesa do korena.

Procesiranje poizvedbe (5)

Dekompozicija » Normalizacija

- Normalizacija zajema pretvorbo poizvedbe v normalizirano obliko, ki jo je lažje obdelovati.
- Pretvori se predikatni (WHERE) del poizvedbe.
- Obstajata dva načina:
 - **Konjunktivna NO**: Zaporedje konjunkcij (pogoji povezani z OR operatorjem) je povezanih z operatorjem AND.
 - **Disjunktivna NO**: Zaporedje disjunkcij (pogoji povezani z AND operatorjem) je povezanih z operatorjem OR.

Procesiranje poizvedbe (6)

Dekompozicija » Semantična analiza (1)

- Pri semantični analizi želimo **preprečiti napačno formulirane** (komponente ne prispevajo k ciljnemu rezultatu) ali **kontradiktorne poizvedbe** (predikatu ne zadošča nobena n-terica).
- Algoritmi za preverjanje semantične pravilnosti potrebujejo **množico poizvedb, ki ne vsebujejo disjunkcije in negacije!**

Procesiranje poizvedbe (7)

Dekompozicija » Semantična analiza (2)

- Metodi za ugotavljanje semantičnih nepravilnosti:
 - **Graf povezav relacij** → preverjanje napačnih formulacij
 - **Graf povezav normaliziranih atributov** → preverjanje kontradikcij

Procesiranje poizvedbe (8)

Dekompozicija » Semantična analiza (3)

- ◉ **Graf povezav relacij:**
 - ◉ Za vsako relacijo in za rezultat kreiramo vozlišče.
 - ◉ Za relacije v stiku kreiraj povezavo med vozlišči.
 - ◉ Za izvor v operacijah projekcije kreiraj povezave med vozlišči.
- ◉ Če graf ni povezan (t.j. obstajajo nepovezani deli), je poizvedba napačno formulirana!

Procesiranje poizvedbe (9)

Dekompozicija » Semantična analiza (4)

- ◉ **Graf normaliziranih atributov:**

- ◉ Za vsako referenco na atribut ali konstanto 0 kreiraj vozlišče.
- ◉ Med vozlišči, ki predstavljajo stik, kreiraj usmerjeno povezavo.
- ◉ Za vsako selekcijo kreiraj usmerjeno povezavo med vozliščema, ki predstavlja atribut selekcije in vozliščem, ki predstavlja konstanto 0.

Procesiranje poizvedbe (10)

Dekompozicija » Semantična analiza (5)

- **Graf normaliziranih atributov:**

- Povezave $a \rightarrow b$ uteži z vrednostjo c , če gre za pogoj neenakosti tipa $a \leq b + c$.
- Povezave $0 \rightarrow a$ uteži z vrednostjo $-c$, če gre za pogoj neenakosti tipa $a \geq c$.
- Če graf vsebuje cikel z negativnim seštevkom uteži, potem je poizvedba kontradiktorna!

Procesiranje proizvodbe (11)

Dekompozicija » Poenostavitev

- Pri poenostavitvi se ukvarjamo z:
 - detekcijo odvečnih kvalifikatorjev,
 - odpravo ponavljajočih sklopov in
 - transformacijo proizvodbe v semantično ekvivalentno, vendar enostavnejšo in za računanje učinkovitejšo obliko!
- V tej fazi se upoštevajo omejitve dostopa, definicije pogledov ter omejitve skladnosti, kar včasih posledično vpliva na pojav redundance.

Procesiranje proizvodbe (12)

Dekompozicija » Reorganizacija

- Pri reorganizaciji se proizvodba reorganizira z namenom zagotavljanja učinkovitejše implementacije.

Naloga #1

Ugotavljanje ali je poizvedba semantično pravilna?

Ugotavljanje semantične pravilnosti ⁽¹⁾

| | |
|---------|---|
| HOTEL | (<u>hotelNo</u> : number, hotelName : varchar2(30), city : varchar2(30)) |
| ROOM | (<u>roomNo</u> : number, <u>hotelNo</u> : number, type : varchar2(30), price : real) |
| BOOKING | (<u>hotelNo</u> : number, <u>guestNo</u> : number, <u>dateFrom</u> : date, <u>dateTo</u> : date, <u>roomNo</u> : number) |
| GUEST | (<u>guestNo</u> : number, guestName : varchar2(30), guestAddress : varchar2(30)) |

- Ugotovite, ali je naslednja poizvedba semantično pravilna?

```
SELECT r.type, r.price
FROM ROOM r, HOTEL h
WHERE r.hotel_number = h.hotel_number AND
      h.hotel_name = 'Habakuk' AND
      r.type > 100;
```


Ugotavljanje semantične pravilnosti (2)

- Potrebno je izvesti vse korake:
 - analiza,
 - normalizacija,
 - semantična analiza,
 - poenostavitve in
 - reorganizacija proizvodbe.

Ugotavljanje semantične pravilnosti (3)

| | |
|---------|---|
| HOTEL | (<u>hotelNo</u> : number, hotelName : varchar2(30), city : varchar2(30)) |
| ROOM | (<u>roomNo</u> : number, <u>hotelNo</u> : number, type : varchar2(30), price : real) |
| BOOKING | (<u>hotelNo</u> : number, <u>guestNo</u> : number, <u>dateFrom</u> : date, <u>dateTo</u> : date, <u>roomNo</u> : number) |
| GUEST | (<u>guestNo</u> : number, guestName : varchar2(30), guestAddress : varchar2(30)) |

```
SELECT r.type, r.price
FROM ROOM r, HOTEL h
WHERE r.hotel_number = h.hotel_number AND
      h.hotel_name = 'Habakuk' AND
      r.type > 100;
```

- Pri analizi najprej preverimo **leksikalno** in **sintaktično analizo** (podobno kot naredijo prevajalniki). ✓

Ugotavljanje semantične pravilnosti (4)

HOTEL (hotelNo: number, **hotelName**: varchar2(30), **city**: varchar2(30))
ROOM (roomNo: number, hotelNo: number, **type**: varchar2(30), **price**: real)
BOOKING (hotelNo: number, guestNo: number, dateFrom: date, dateTo: date, roomNo: number)
GUEST (guestNo: number, **guestName**: varchar2(30), **guestAddress**: varchar2(30))

```
SELECT r.type, r.price
FROM ROOM r, HOTEL h
WHERE r.hotel_number = h.hotel_number AND
h.hotel_name = 'Habakuk' AND
r.type > 100;
```

- Dodatno preveri ali so relacije in atributi zapisani v sistemskem katalogu in če so operacije primerne glede na tip objekta. **x**

Naloga #2

Ugotavljanje ali je poizvedba semantično pravilna?

Ugotavljanje semantične pravilnosti ⁽¹⁾

| | |
|---------|---|
| HOTEL | (<u>hotelNo</u> : number, hotelName : varchar2(30), city : varchar2(30)) |
| ROOM | (<u>roomNo</u> : number, <u>hotelNo</u> : number, type : varchar2(30), price : real) |
| BOOKING | (<u>hotelNo</u> : number, <u>guestNo</u> : number, <u>dateFrom</u> : date, <u>dateTo</u> : date, <u>roomNo</u> : number) |
| GUEST | (<u>guestNo</u> : number, guestName : varchar2(30), guestAddress : varchar2(30)) |

- Ugotovite, ali je poizvedba, ki za vsak hotel iz Pirana izpiše podatke o gostih in sobah, ki so jim pripadale, semantično pravilna?

| | |
|---------------|--|
| SELECT | g.guestName, g.guestAddress |
| FROM | ROOM r, HOTEL h, BOOKING b, GUEST g |
| WHERE | r.roomNo = b.roomNo AND b.hotelNo = h.hotelNo AND h.city = 'Piran' AND b.dateTo < '15.08.2006'; |

Ugotavljanje semantične pravilnosti (2)

| | |
|---------|---|
| HOTEL | (<u>hotelNo</u> : number, hotelName : varchar2(30), city : varchar2(30)) |
| ROOM | (<u>roomNo</u> : number, <u>hotelNo</u> : number, type : varchar2(30), price : real) |
| BOOKING | (<u>hotelNo</u> : number, <u>guestNo</u> : number, <u>dateFrom</u> : date, <u>dateTo</u> : date, <u>roomNo</u> : number) |
| GUEST | (<u>guestNo</u> : number, guestName : varchar2(30), guestAddress : varchar2(30)) |

| | |
|---------------|--|
| SELECT | g.guestName, g.guestAddress |
| FROM | ROOM r, HOTEL h, BOOKING b, GUEST g |
| WHERE | r.roomNo = b.roomNo AND b.hotelNo = h.hotelNo AND h.city = 'Piran' AND b.dateTo < '15.08.2006'; |

- Pri analizi najprej preverimo **leksikalno** in **sintaktično analizo** (podobno kot naredijo prevajalniki). ✓

Ugotavljanje semantične pravilnosti (3)

| | |
|---------|---|
| HOTEL | (<u>hotelNo</u> : number, hotelName : varchar2(30), city : varchar2(30)) |
| ROOM | (<u>roomNo</u> : number, <u>hotelNo</u> : number, type : varchar2(30), price : real) |
| BOOKING | (<u>hotelNo</u> : number, <u>guestNo</u> : number, <u>dateFrom</u> : date, <u>dateTo</u> : date, <u>roomNo</u> : number) |
| GUEST | (<u>guestNo</u> : number, guestName : varchar2(30), guestAddress : varchar2(30)) |

| | |
|---------------|--|
| SELECT | g.guestName, g.guestAddress |
| FROM | ROOM r, HOTEL h, BOOKING b, GUEST g |
| WHERE | r.roomNo = b.roomNo AND b.hotelNo = h.hotelNo AND h.city = 'Piran' AND b.dateTo < '15.08.2006'; |

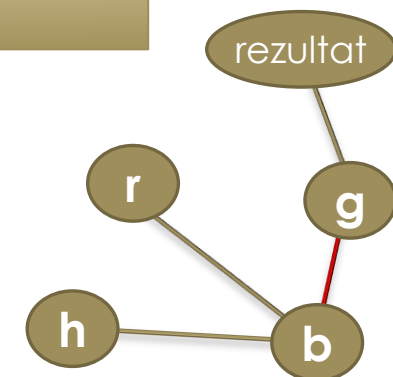
- Dodatno preveri ali so relacije in atributi zapisani v sistemskem katalogu in če so operacije primerne glede na tip objekta. ✓

Ugotavljanje semantične pravilnosti (4)

HOTEL (hotelNo: number, **hotelName**: varchar2(30), **city**: varchar2(30))
ROOM (roomNo: number, hotelNo: number, **type**: varchar2(30), **price**: real)
BOOKING (hotelNo: number, guestNo: number, dateFrom: date, dateTo: date, roomNo: number)
GUEST (guestNo: number, **guestName**: varchar2(30), **guestAddress**: varchar2(30))

SELECT g.guestName, g.guestAddress
FROM ROOM r, HOTEL h, BOOKING b, GUEST g
WHERE r.roomNo = b.roomNo **AND** b.hotelNo = h.hotelNo **AND** h.city = 'Piran' **AND** b.dateTo < '15.08.2006';

- Pri semantični analizi preverimo graf povezav relacij. **x**



Procesiranje poizvedbe (13)

Optimizacija (1)

- **Procesiranje poizvedbe** sestavlja:

- 2. **Optimizacija**

- Izbira najučinkovitejše izvedbene strategije za procesiranje poizvedbe, kjer obstaja dva pogleda:
 - doseči želimo najkrajši čas izvajanja poizvedbe,
 - želimo čim bolj izkoristiti razpoložljiva sredstva (npr. vzporedne operacije).

Procesiranje poizvedbe (14)

Optimizacija (2)

- Spoznali bomo hevristično optimizacijo poizvedb, kjer uporabljamo hevristična pravila.
- Na ta način izraz v relacijski algebri pretvorimo, kjer je končni rezultat učinkovitejši.

Procesiranje poizvedbe (15)

Optimizacija (3)

Transformacijsko pravilo 1

• Kaskada selekcije: $\sigma_{p \wedge q \wedge r}(R) = \sigma_p(\sigma_q(\sigma_r(R)))$

• Primer: $\sigma_{\text{branchNo} = 'B003' \wedge \text{salary} > 15000}(\text{Staff}) =$
 $\sigma_{\text{branchNo} = 'B003'}(\sigma_{\text{salary} > 15000}(\text{Staff}))$

Transformacijsko pravilo 2

• Komutativnost oper. selek.: $\sigma_p(\sigma_q(R)) = \sigma_q(\sigma_p(R))$

• Primer: $\sigma_{\text{branchNo} = 'B003'}(\sigma_{\text{salary} > 15000}(\text{Staff})) =$
 $\sigma_{\text{salary} > 15000}(\sigma_{\text{branchNo} = 'B003'}(\text{Staff}))$

Procesiranje poizvedbe (16)

Optimizacija (4)

Transformacijsko pravilo 3

- Pri zaporedju operacij projekcije je pomembna le zadnja: $\Pi_L \Pi_M \dots \Pi_N(R) = \Pi_L(R)$

- Primer: $\Pi_{\text{IName}} \Pi_{\text{branchNo, IName}}(\text{Staff}) = \Pi_{\text{IName}}(\text{Staff})$

Transformacijsko pravilo 4

- Komutativnost operacij selekcije in projekcije:

$$\Pi_{A_i, \dots, A_m}(\sigma_p(R)) = \sigma_p(\Pi_{A_i, \dots, A_m}(R)), \text{ kjer } p \in \{A_1, A_2, \dots, A_m\}$$

- Primer: $\Pi_{\text{fName, IName}}(\sigma_{\text{IName}='Beech'}(\text{Staff})) = \sigma_{\text{IName}='Beech'}(\Pi_{\text{fName, IName}}(\text{Staff}))$

Procesiranje poizvedbe (17)

Optimizacija (6)

Transformacijsko pravilo 5

- Komutativnost theta stika, kartezijskega produkta, equijoin in naravni stik: $R \bowtie_p S = S \bowtie_p R$

- Primer: $Staff \bowtie_{staff.branchNo = branch.branchNo} Branch =$
 $Branch \bowtie_{staff.branchNo = branch.branchNo} Staff$

Procesiranje poizvedbe (18)

Optimizacija (7)

Transformacijsko pravilo 6

- Komutativnost selekcije in theta stika (ali kartezijskega produkta): $\sigma_p(R \bowtie_r S) = (\sigma_p(R)) \bowtie_r S$,
kjer $p \in \{A_1, A_2, \dots, A_n\}$ in

$$\sigma_{p \wedge q}(R \bowtie_r S) = (\sigma_p(R)) \bowtie_r (\sigma_q(S))$$

- Primer: $\sigma_{\text{position} = \text{'Manager'} \wedge \text{city} = \text{'London'}}(\text{Staff} \bowtie_{\text{Staff.branchNo}} \text{Branch}) =$
 $\text{Branch.branchNo} (\sigma_{\text{position} = \text{'Manager'}}(\text{Staff})) \bowtie_{\text{Staff.branchNo}} =$
 $\text{Branch.branchNo} (\sigma_{\text{city} = \text{'London'}}(\text{Branch}))$

Procesiranje poizvedbe (19)

Optimizacija (8)

Transformacijsko pravilo 7

- Komutativnost projekcije in theta stika (ali kartezijskega produkta): $\Pi_{L_1 \cup L_2}(R \bowtie_r S) = (\Pi_{L_1}(R)) \bowtie_r (\Pi_{L_2}(S))$, kjer je $L = L_1 \cup L_2$ in L_1 atributi relacije R in L_2 atributi relacije S

- Primer: $\Pi_{\text{position, city, branchNo}}(\text{Staff} \bowtie_{\text{Staff.branchNo} = \text{Branch.branchNo}} \text{Branch}) =$
 $(\Pi_{\text{position, branchNo}}(\text{Staff})) \bowtie_{\text{Staff.branchNo} = \text{Branch.branchNo}} (\Pi_{\text{city, branchNo}}(\text{Branch}))$

Procesiranje poizvedbe (20)

Optimizacija (9)

Transformacijsko pravilo 7 (dodatek)

- Če pogoj stika vsebuje attribute, ki jih ni v L ($M = M_1 \cup M_2$ in M_1 attribute relacije R in M_2 attribute relacije S): $\Pi_{L_1 \cup L_2}(R \bowtie_r S) =$

$$\Pi_{L_1 \cup L_2}(\Pi_{L_1 \cup M_1}(R)) \bowtie_r (\Pi_{L_2 \cup M_2}(S))$$

- Primer: $\Pi_{\text{position, city}}(\text{Staff} \bowtie_{\text{Staff.branchNo} =$

Branch.branchNo Branch) =

$\Pi_{\text{position, city}}((\Pi_{\text{position, branchNo}} \text{Staff})) \bowtie_{\text{Staff.branchNo} =$

$\text{Branch.branchNo} (\Pi_{\text{city, branchNo}} (\text{Branch}))$

Procesiranje poizvedbe (21)

Optimizacija (10)

- **Transformacijsko pravilo 8**

- Komutativnost operacij unije in preseka:

$$R \cup S = S \cup R \text{ in } R \cap S = S \cap R$$

- **Transformacijsko pravilo 9**

- Komutativnost operacij selekcije in operacij množic:

$$\sigma_p(R \cup S) = \sigma_p(S) \cup \sigma_p(R)$$

$$\sigma_p(R \cap S) = \sigma_p(S) \cap \sigma_p(R)$$

$$\sigma_p(R - S) = \sigma_p(S) - \sigma_p(R)$$

Procesiranje poizvedbe (22)

Optimizacija (11)

Transformacijsko pravilo 10

- Komutativnost projekcije in unije:

$$\Pi_L(R \cup S) = \Pi_L(S) \cup \Pi_L(R)$$

Transformacijsko pravilo 11

- Asociativnost theta stika in kartezijskega produkta: $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$

- Primer: $(\text{Staff} \bowtie_{\text{Staff.staffNo} = \text{PropertyForRent.staffNo}} \text{PropertyForRent}) \bowtie_{\text{ownerNo} = \text{Owner.ownerNo} \wedge \text{staff.lName} = \text{Owner.lName}} \text{Owner} =$
 $\text{Staff} \bowtie_{\text{staff.staffNo} = \text{PropertyForRent.staffNo} \wedge \text{staff.lName} = \text{Owner.lName}} (\text{PropertyForRent} \bowtie_{\text{ownerNo}} \text{Owner})$

Procesiranje poizvedbe (23)

Optimizacija (12)

• Transformacijsko pravilo 12

- Asociativnost unije in preseka:

$$(R \cup S) \cup T = S \cup (R \cup T) \text{ in}$$

$$(R \cap S) \cap T = S \cap (R \cap T)$$

Procesiranje poizvedbe (24)

Optimizacija (13)

- Poznamo **5 strategij hevrističnega procesiranja**:
 - **Strategija 1**
 - Selekcijo izvedi čim prej (zmanjša se kardinalnost)
 - Selekcije nad isto relacijo naj bodo skupaj
 - Uporabi transformacije 2, 4, 6 in 9
 - **Strategija 2**
 - Združi kartezijski produkt s selekcijami, ki so hkrati stični pogoj

Procesiranje poizvedbe (25)

Optimizacija (14)

◉ Strategija 3

- ◉ Relacije čim bolj zreduciramo, preden uporabimo binarne operacije – najbolj omejevalne selekcije se upoštevajo na začetku
- ◉ Podobno asociativnost uporabimo na theta stiku, kjer se manjši stiki izvedejo prej

◉ Strategija 4

- ◉ Projekcije uredimo kaskadno (pravilo 3) in projekcijo premaknemo bliže listom drevesa (pravila 4, 7 in 10)
- ◉ Projekcijo izvedemo na samem začetku
- ◉ Projekcije nad istimi relacijami naj bodo skupaj

Procesiranje poizvedbe (26)

Optimizacija (15)

- **Strategija 5**

- Če se rezultat pojavi večkrat in ni prevelik, ga shrani za nadaljnjo uporabo
- Če je možno se rezultat shrani v primarni pomnilnik, sicer se izračuna kaj stane več: branje iz sekundarnega pomnilnika ali ponovni izračun izraza

Naloga #3

Optimizirajte proizvodbo z uporabo hevrističnih pravil

Optimizacije poizvedbe (1)

| | |
|---------|---|
| HOTEL | (<u>hotelNo</u> : number, hotelName : varchar2(30), city : varchar2(30)) |
| ROOM | (<u>roomNo</u> : number, <u>hotelNo</u> : number, type : varchar2(30), price : real) |
| BOOKING | (<u>hotelNo</u> : number, <u>guestNo</u> : number, <u>dateFrom</u> : date, <u>dateTo</u> : date, <u>roomNo</u> : number) |
| GUEST | (<u>guestNo</u> : number, guestName : varchar2(30), guestAddress : varchar2(30)) |

- Z uporabo hevrističnih pravil optimizirajte naslednjo poizvedbo!

| | |
|---------------|---|
| SELECT | r.roomNo, r.type, r.price |
| FROM | ROOM r, BOOKING b, HOTEL h |
| WHERE | r.roomNo = b.roomNo AND b.hotelNo = h.hotelNo AND h.hotelName = 'Habakuk' AND r.price > 100; |

Optimizacija poizvedbe (2)

- Najprej SQL poizvedbo pretvorimo v relacijsko algebro:

```
SELECT    r.roomNo, r.type, r.price  
FROM      ROOM r, BOOKING b, HOTEL h  
WHERE     r.roomNo = b.roomNo AND b.hotelNo = h.hotelNo AND  
           h.hotelName = 'Habakuk' AND r.price > 100;
```

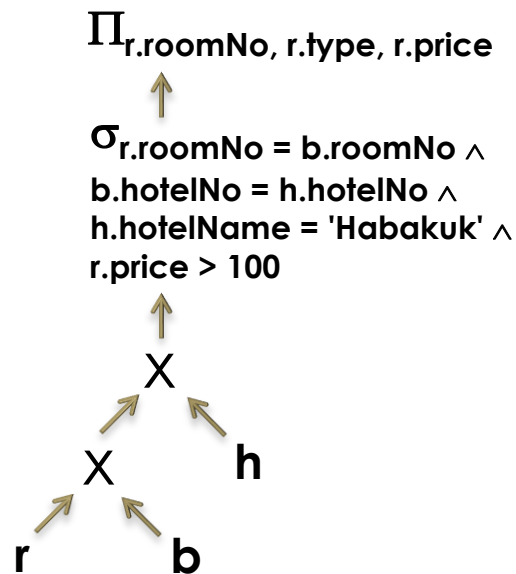

$$\Pi_{r.roomNo, r.type, r.price}(\sigma_{r.roomNo = b.roomNo \wedge b.hotelNo = h.hotelNo \wedge h.hotelName = 'Habakuk' \wedge r.price > 100}((r \times b) \times h))$$

Optimizacija poizvedbe (3)

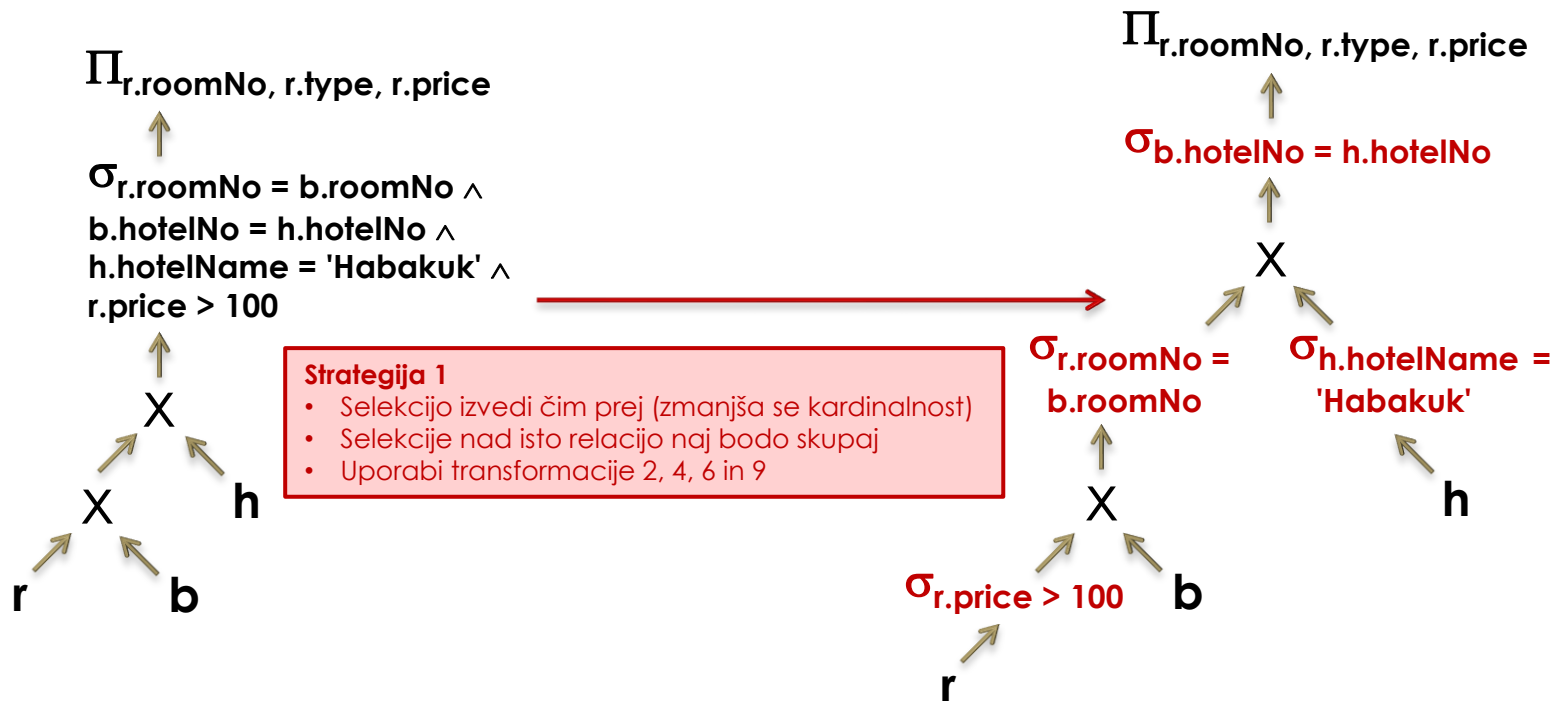
- Nato poizvedbo v relacijski algebri predstavimo kot kanonično drevo relacijske algebre:
 - Osnovne relacije postanejo listi drevesa,
 - Vmesne relacije so vozlišča drevesa,
 - Rezultat je v korenu drevesa,
 - Zaporedje operacij si sledi od listov do korena.

Optimizacija poizvedbe (4)

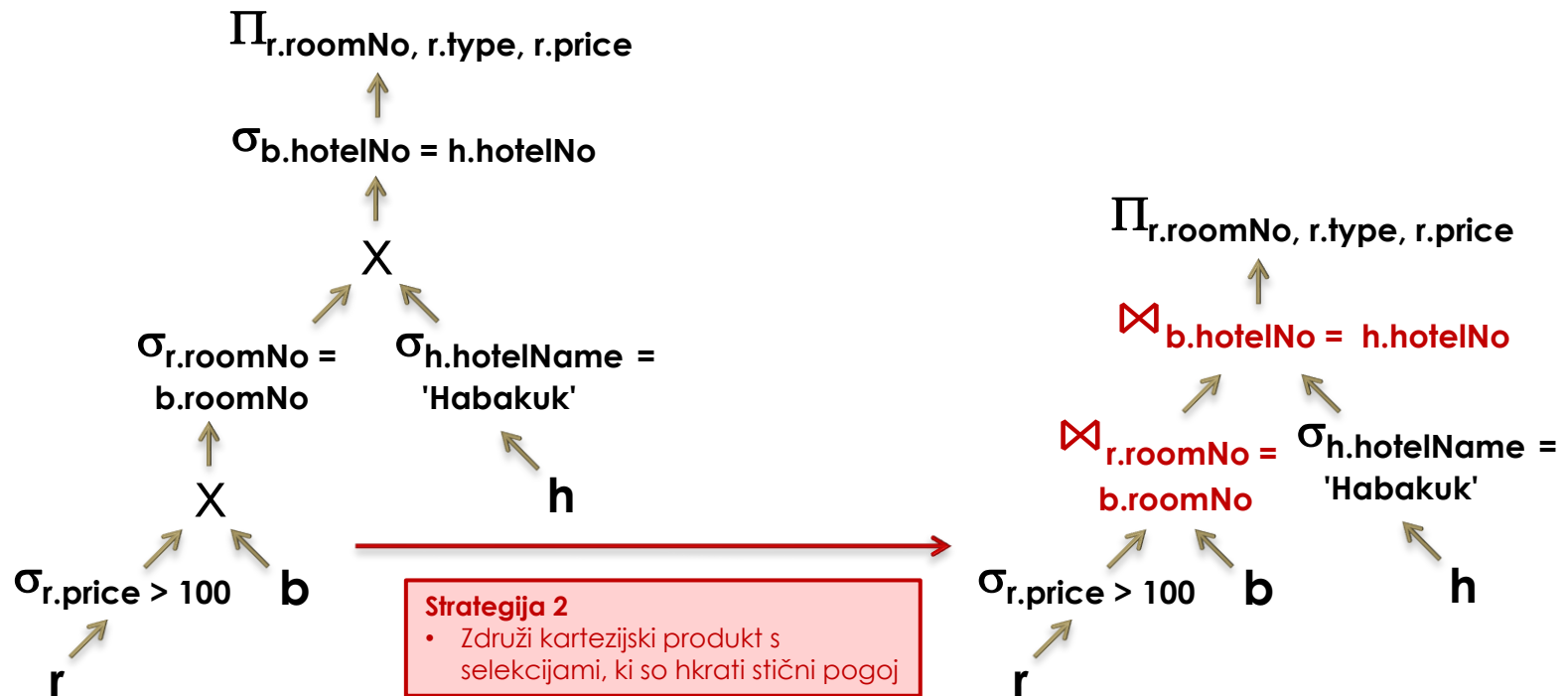
$\Pi_{r.roomNo, r.type, r.price}(\sigma_{r.roomNo = b.roomNo \wedge b.hotelNo = h.hotelNo \wedge h.hotelName = 'Habakuk' \wedge r.price > 100}((r \times b) \times h))$



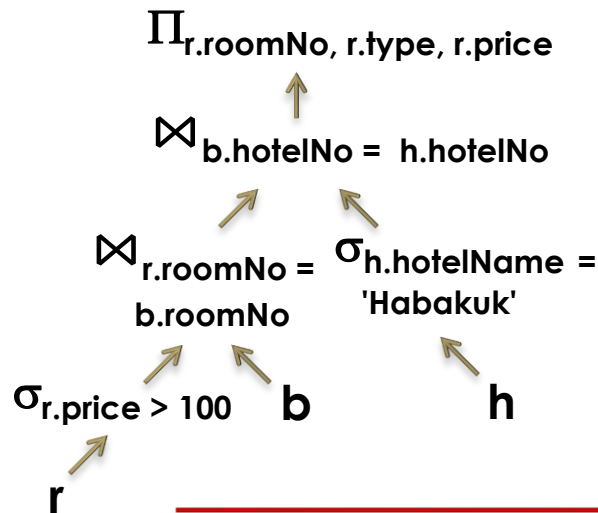
Optimizacija poizvedbe (5)



Optimizacija poizvedbe (6)

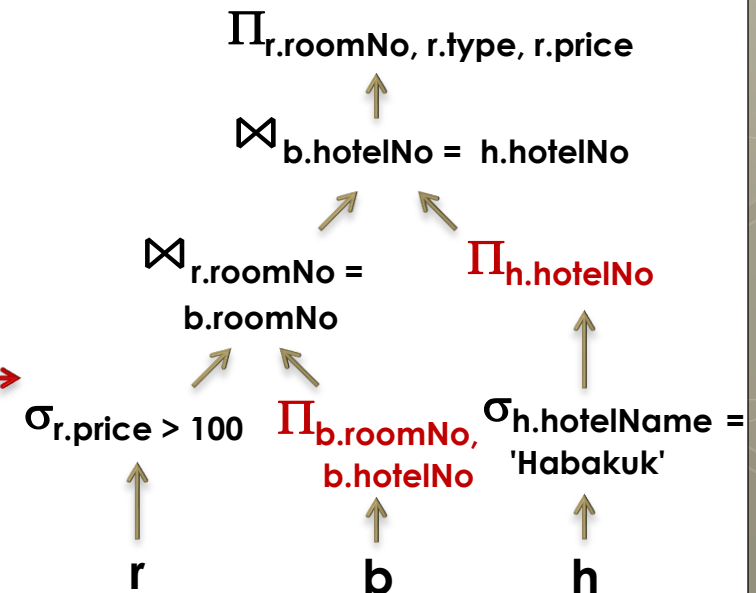


Optimizacija poizvedbe (7)

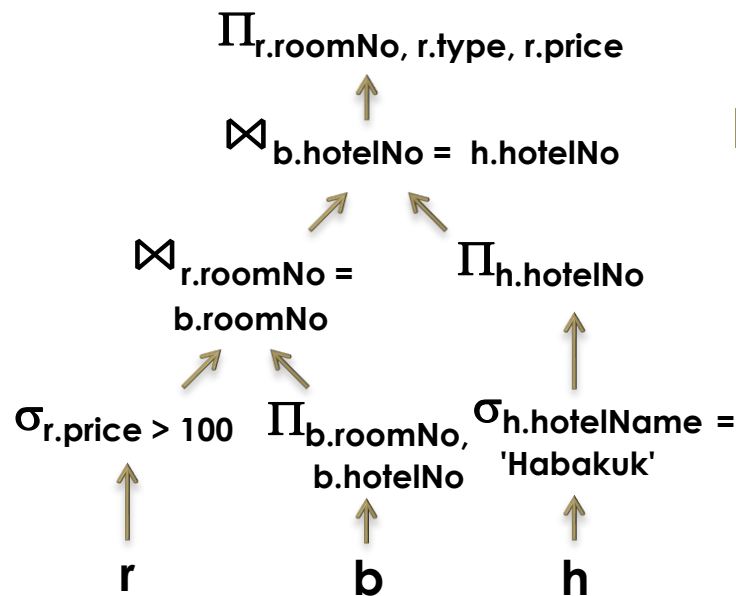


Strategija 4

- Projekcije uredimo kaskadno (pravilo 3) in projekcijo premaknemo bliže listom drevesa (pravila 4, 7 in 10)
- Projekcijo izvedemo na samem začetku
- Projekcije nad istimi relacijami naj bodo skupaj



Optimizacija poizvedbe (8)



Optimizirana rešitev

```

Πr.roomNo, r.type, r.price (
  ( σr.price > 100(r)
    ⋈r.roomNo = b.roomNo
      ( Πr.roomNo, b.roomNo(b) )
    ⋈b.roomNo = h.roomNo
      ( Πh.roomNo(σh.roomNo = 'Habakuk'(h)) )
  )

```

Prvotna rešitev

```

Πr.roomNo, r.type, r.price (
  σr.roomNo = b.roomNo ∧
    b.roomNo = h.roomNo ∧
    h.roomNo = 'Habakuk' ∧ r.price > 100
    ( r x b ) x h
  )

```