Ottimizzazione Logica

Lorenzo Vaccarecci

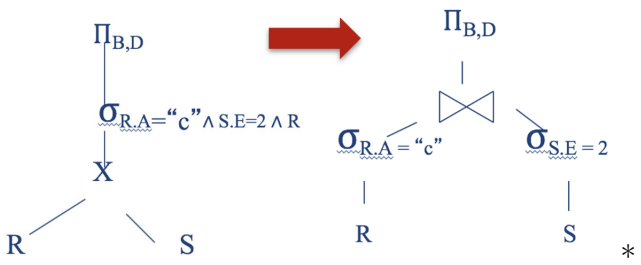
19 Aprile 2024

1 Introduzione

- **Input:** espressione algebrica
- **Output:** piano di esecuzione logico ottimizzato (LQP)

Piano di esecuzione logico: espressione algebrica per l'interrogazione rappresentata come albero

$$\Pi_{B,D}\,[\sigma_{R.A=\text{“c”}\,\wedge\,S.E=2\,\wedge\,R.C\,=\,S.C}\,(R\ X\ S)]$$



*

**L'esempio è incompleto ma rende l'idea*
Per ottimizzare l'albero, sfrutta le proprietà dell'algebra relazionale

2 Come avviene l'ottimizzazione

- si basa su equivalenze algebriche
  - si dicono equivalenti se e_1 e e_2 restituiscono lo **stesso risultato**
- queste equivalenze sono usate come regole di riscrittura, utilizzando opportune euristiche per passare da una espressione algebrica a un'altra

Regola di riscrittura: posso sostituire e_1 con e_2 se valgono determinate condizioni C.
Nell'ottimizzazione logica la regola verrà considerata se e_2 è "efficiente"

2.1 Esempi notevoli (da ricordare)

Altri esempi sulle slide, queste sono le principali

2.1.1 Selezione

$$\sigma_{P1}(\sigma_{P2}(e)) \equiv \sigma_{P2}(\sigma_{P1}(e)) \equiv \sigma_{P1 \wedge P2}(e)$$

2.1.2 Proiezione

$$\Pi_{A1,\ldots,An}(\Pi_{B1,\ldots,Bn}(e)) \equiv \Pi_{A1,\ldots,An}(e)$$

- Permette di gestire cascate di proiezioni
- vale se $\{A1, \ldots, An\} \subseteq \{B1, \ldots, Bn\}$

2.1.3 Commutazione di selezione e proiezione

$$\Pi_{A1,\ldots,An}(\sigma_P(e)) \equiv \sigma_P(\Pi_{A1,\ldots,An}(e))$$

Vale solo se $P \subseteq \{A1, \ldots, An\}$

2.1.4 Commutazione di selezione e prodotto cartesiano

$$\sigma_P(e_1 \times e_2) \equiv \sigma_P(e_1) \times e_2$$

Vale solo se P coinvolge gli attributi di e_1

2.1.5 Commutazione di proiezione e prodotto cartesiano

$$\Pi_{A1,\ldots,An}(e_1 \times e_2) \equiv \Pi_{B1,\ldots,Bn}(e_1) \times \Pi_{C1,\ldots,Cn}(e_2)$$

2.1.6 Selezioni, prodotto cartesiano e join

$$\sigma_P(e_1 \times e_2) \equiv e_1 \bowtie_P e_2$$

2.1.7 Prodotto cartesiano e join (Slide 34) Da completare

- **Commutatività**
  - $e_1 \bowtie_F e_2 \equiv e_2 \bowtie_F e_1$
  - $e_1 \bowtie e_2 \equiv e_2 \bowtie e_1$
  - $e_1 \times e_2 \equiv e_2 \times e_1$
- **Associatività**
  -

2.1.8 Unione e differenza (Slide 35) Da completare

- **Commutatività**
- **Associatività**

3 Euristiche

Le euristiche permettono di trasformare le equivalenze in regole di riscrittura. Va un po' a "buon senso", guarda quale delle riscritture ha l'input minore.

3.1 Euristiche fondamentali

- anticipare il più possibile **selezioni** e **proiezioni**
- fattorizzare condizioni di selezione complesse per aumentare la possibilità di applicare le regole di riscrittura

3.2 Euristiche principali

Nessuna regola di riscrittura si riferisce all'ordine con cui eseguire un insieme di join (verrà ottimizzato al livello fisico).

- **1A**: eseguire le operazioni di selezione il più presto possibile
- **1B**: eseguire le operazioni di proiezione il più presto possibile
- **2**: fattorizzare condizioni di selezione complesse per aumentare la possibilità di applicare le regole di riscrittura (come ad esempio 2.1.1)

3.2.1 Esempio

Relazioni: $R(ABC) \quad S(DEF)$
Query: $\Pi_{BE}(\sigma_{A>5 \wedge F>6 \wedge C=D}(R \times S))$
Ottimizzazione:

$\rightarrow^{1A} \Pi_{BE}(\sigma_{C=D}(\sigma_{A>5}(\sigma_{F>6}(R \times S))))$

$\rightarrow^{1A} \Pi_{BE}(\sigma_{C=D}(\sigma_{A>5}(R \times \sigma_{F>6}(S))))$

$\rightarrow^{1A} \Pi_{BE}(\sigma_{C=D}(\sigma_{A>5}(R) \times \sigma_{F>6}(S)))$

4 Output

L'output della fase di ottimizzazione logica è un singolo LQP ottimizzato. Si potrebbe anche guardare LQP alternativi ma non viene utilizzata perchè per ognuno bisognerebbe individuarne il piano fisico ottimale.