Databases

L2 sciences et technologies, mention informatique

conjunctive algebra

how to query this database (with an algebraic language)?

movies	title	director	year
	starwars	lucas	1977
	nikita	besson	1990
	locataires	ki-duk	2005
	dune	lynch	1984

directors	name	nationality
	lucas	american
	lynch	american
	besson	french
	ki-duk	korean

patrick.marcel@univ-tours.fr
http://celene.univ-tours.fr/course/view.php?id=3131

in what follows...

consider the following database instance:

movies	title	director	year
	starwars1	lucas	1977
	nikita	besson	1990
	locataires	ki-duk	2005
	dune	lynch	1984
	starwars4	lucas	1999
	starwars5	lucas	2002
	the island	ki-duk	2000
	lucy	besson	2014
	eraserhead	lynch	1976
	...		

directors	name	nationality
	lucas	american
	lynch	american
	besson	french
	ki-duk	korean
	...	

actors	name	movie
	ford	starwars1
	ford	indiana jones
	willis	5th element
	...	

algebraic language

definition of unary and binary operators over relation instances

two approaches:

- unnamed approach
  SPC algebra
- named approach
  SPJR algebra

SPC algebra

SPC algebra

attribute names are not used in the operator definitions

S :	Selection	σ
P :	Projection	π
C :	Cartesian product	$\times$

example

how to *construct* the tuples of the answer to the query?
"list movies directed by americans"

4 steps:

1. select tuples of *directors* corresponding to american directors
2. combine these tuples with that of *movies* using Cartesian product
3. restrict to tuples where positions for directors are the same
4. output the movie titles only

step 1

$$I_1 := \sigma_{2=''american''}(\text{directors})$$

$$I_1 = \{(\text{lucas,american}),(\text{lynch,american}), \ldots\}$$

step 2

$I_2 := I_1 \times$ movies

$I_2 = \{$(lucas,american,starwars1,lucas,1977),
(lucas,american,dune,lynch,1984),
(lucas,american,nikita,besson,1990),
(lynch,american,dune,lynch,1984),
(lynch,american,starwars1,lucas,1977), $\ldots\}$

step 3

$I_3 := \sigma_{1=4}(I_2)$

$I_3 = \{$(lucas,american,starwars1,lucas,1977),
(lynch,american,dune,lynch,1984),
(lucas,american,starwars4,lucas,1999),
(lucas,american,starwars5,lucas,2002),
(lucas,american,starwars6,lucas,2005),
(lynch,american,eraserhead,lynch,1976), $\ldots\}$

step 4

$$I_4 := \pi_3(I_3)$$

$I_4 = \{$(starwars1), (dune), (starwars4), (starwars5), (starwars6), (eraserhead), $\ldots\}$

step 4

$I_4 := \pi_3(I_3)$

$I_4 = \{$(starwars1), (dune), (starwars4), (starwars5), (starwars6), (eraserhead), $\ldots\}$

all together: $I_4 = \pi_3(\sigma_{1=4}(\sigma_{2=''american''}(\text{directors}) \times \text{movies}))$

Selection

let $j, k \in \mathbb{N}$ and $a \in \mathbf{dom}$, I a relation instance, such that $max(j, k) \leq \text{arity}(I)$

$$\sigma_{j=a}(I) = \{t \in I | t(j) = a\}$$

$$\sigma_{j=k}(I) = \{t \in I | t(j) = t(k)\}$$

Generalized selection

the generalized form of selection:

$$\sigma_\varphi$$

where φ is a conjunctive selection formula:

- $\varphi = \gamma_1 \wedge \ldots \wedge \gamma_n$
- and the γ_i are of the form $j = a$ or $j = k$

σ_φ is equivalent to $\sigma_{\gamma_1}(\ldots(\sigma_{\gamma_n}(I)))$

Projection

let $j_1, \ldots, j_n \in \mathbb{N}$ and I a relation instance, such that $max(j_1, \ldots, j_n) \leq \text{arity}(I)$

$$\pi_{j_1,\ldots,j_n}(I) = \{(t(j_1), \ldots, t(j_n)) | t \in I\}$$

Cartesian product

let I and J be two relation instances such that arity(I)$= n$ and arity(J) $= m$

$$I \times J = \{(t(1), \ldots, t(n), s(1), \ldots, s(m)) | t \in I, s \in J\}$$

Cartesian product

let I and J be two relation instances such that arity(I)$= n$ and arity$(J) = m$

$$I \times J = \{(t(1), \ldots, t(n), s(1), \ldots, s(m)) | t \in I, s \in J\}$$

associative, non commutative, with relation $\{()\}$ for neutral element

syntax of SPC queries

for a database schema D, an SPC query q is:

syntax of SPC queries

for a database schema D, an SPC query q is:

- R, if $R \in D$, $\mathrm{arity}(q) = \mathrm{arity}(R)$

syntax of SPC queries

for a database schema D, an SPC query q is:

- R, if $R \in D$, $\text{arity}(q) = \text{arity}(R)$
- $\{(a)\}$ if $a \in \textbf{dom}$, $\text{arity}(q) = 1$

syntax of SPC queries

for a database schema D, an SPC query q is:

- R, if $R \in D$, $\mathrm{arity}(q) = \mathrm{arity}(R)$
- $\{(a)\}$ if $a \in \mathbf{dom}$, $\mathrm{arity}(q) = 1$
- $\sigma_\varphi(q')$ if q' is a query, $\mathrm{arity}(q) = \mathrm{arity}(q')$

syntax of SPC queries

for a database schema D, an SPC query q is:

- R, if $R \in D$, $\text{arity}(q) = \text{arity}(R)$
- $\{(a)\}$ if $a \in \textbf{dom}$, $\text{arity}(q) = 1$
- $\sigma_\varphi(q')$ if q' is a query, $\text{arity}(q) = \text{arity}(q')$
- $\pi_{j_1,\ldots,j_n}(q')$ if q' is a query, $\text{arity}(q) = n$

syntax of SPC queries

for a database schema D, an SPC query q is:

- R, if $R \in D$, $\mathrm{arity}(q) = \mathrm{arity}(R)$
- $\{(a)\}$ if $a \in \mathbf{dom}$, $\mathrm{arity}(q) = 1$
- $\sigma_\varphi(q')$ if q' is a query, $\mathrm{arity}(q) = \mathrm{arity}(q')$
- $\pi_{j_1,\ldots,j_n}(q')$ if q' is a query, $\mathrm{arity}(q) = n$
- $q_1 \times q_2$ if q_1 and q_2 is a query, $\mathrm{arity}(q) = \mathrm{arity}(q_1) + \mathrm{arity}(q_2)$

syntax of SPC queries

for a database schema D, an SPC query q is:

- R, if $R \in D$, $\text{arity}(q) = \text{arity}(R)$
- $\{(a)\}$ if $a \in \mathbf{dom}$, $\text{arity}(q) = 1$
- $\sigma_{\varphi}(q')$ if q' is a query, $\text{arity}(q) = \text{arity}(q')$
- $\pi_{j_1,\ldots,j_n}(q')$ if q' is a query, $\text{arity}(q) = n$
- $q_1 \times q_2$ if q_1 and q_2 is a query, $\text{arity}(q) = \text{arity}(q_1) + \text{arity}(q_2)$

for an instance I and a query q, we note $q(I)$ the image of I by q

examples

what is the release year of “nikita”?

$$\pi_3(\sigma_{1=''nikita''}(movies))$$

examples

what is the release year of “nikita”?

$\pi_3(\sigma_{1=''nikita''}(movies))$

what is the nationality of the director of “locataires” ?

$\pi_5(\sigma_{2=4}(\sigma_{1=''locataires''}(movies) \times directors)$

remark

some SPC queries are not satisfiables...

remark

some SPC queries are not satisfiables...

$\sigma_{1=a}(\sigma_{1=b}(I))$ with arity$(I) \geq 1$ and $a \neq b$

intersection

consider the intersection $\cap$ of 2 instances

let I and J be two instances of the same arity

$$I \cap J = \{t | t \in I \text{ and } t \in J\}$$

$\cap$ can be simulated with operators $\sigma, \pi, \times$

equi-join

for two instances I and J, the equi-join $\bowtie_\varphi$ is defined by:

let $\varphi = (j_1 = k_1) \wedge \ldots \wedge (j_n = k_n)$ be a formula such that $j_i \in [1, arity(I)]$ and $k_i \in [1, arity(J)]$

$$I \bowtie_\varphi J = \sigma_{\varphi'}(I \times J)$$

with $\varphi' = (j_1 = k_1 + arity(I)) \wedge \ldots \wedge (j_n = k_n + arity(I))$

example

the query "list movies directed by americans"

can be written:

$\pi_3(\sigma_{2=''american''}(\text{directors}) \bowtie_{1=2} \text{movies}))$

normal form

any SPC query can be written:

$$\pi_{j_1,\ldots,j_n}(\{(a_1)\} \times \ldots \times \{(a_m)\} \times \sigma_\varphi(R_1 \times \ldots \times R_k))$$

where

- $a_1, \ldots, a_m \in$ **dom**,
- the R_i are relation names,
- φ is a conjunctive selection formula

example

$$q = \pi_3(\sigma_{1=4}(\sigma_{2=''american''}(\text{directors}) \times \text{movies}))$$

can be rewritten

$$q = \pi_2(\sigma_{2=4 \wedge 5=''american''}(\text{movies} \times \text{directors}))$$

SPJR algebra

SPJR algebra

attribute names are used in the operator definitions

S :	Selection	σ
P :	Projection	π
J :	natural Join	$\bowtie$
R :	Renaming	ρ

example

the query "list the movies directed by americans"

can be written

$$\pi_{title}(\sigma_{nationality=''american''}(\text{directors}) \bowtie \rho_{title,director,year \to title,name,year}(\text{movies})))$$

Selection

let $c \in \mathbf{dom}$, I be an instance with $A, B \in sort(I)$

$$\sigma_{A=c}(I) = \{t \in I | t(A) = c\}$$
$$\sigma_{A=B}(I) = \{t \in I | t(A) = t(B)\}$$

Selection

let $c \in \mathbf{dom}$, I be an instance with $A, B \in sort(I)$

$$\sigma_{A=c}(I) = \{t \in I | t(A) = c\}$$
$$\sigma_{A=B}(I) = \{t \in I | t(A) = t(B)\}$$

the generalized form σ_φ with φ a conjunctive selection formula

Projection

let I be an instance and $A_1, \ldots, A_n \in sort(I)$

$$\pi_{A_1,\ldots,A_n}(I) = \{(A_1 : t(A_1), \ldots, A_n : t(A_n)) | t \in I\}$$

or simply: $\pi_{A_1,\ldots,A_n}(I) = \{t|_{\{A_1\ldots,A_n\}} \; |t \in I\}$

natural Join

let I and J be two instances

$$
\begin{aligned}
I \bowtie J = \{t \text{ over } sort(I) \cup sort(J) | \\
&\exists v \in I \text{ and } w \in J,\ t|_{sort(I)} = v \\
&\text{and } t|_{sort(J)} = w\}
\end{aligned}
$$

natural Join

let I and J be two instances

$$I \bowtie J = \{t \text{ over } sort(I) \cup sort(J) | \\ \exists v \in I \text{ and } w \in J,\ t|_{sort(I)} = v \\ \text{and } t|_{sort(J)} = w\}$$

associative and commutative operation with relation $\{()\}$ as neutral element

example of natural Join

consider the relation instances:

I

A	B
1	2
4	2

J

B	C
2	3
2	5

K

A	B
1	2
2	3

L

A	D
1	2
3	4

example of natural Join

consider the relation instances:

I

A	B
1	2
4	2

J

B	C
2	3
2	5

K

A	B
1	2
2	3

L

A	D
1	2
3	4

$I \bowtie J$

A	B	C
1	2	3
1	2	5
4	2	3
4	2	5

$I \bowtie K$

A	B
1	2

$J \bowtie L$

A	B	C	D
1	2	3	2
3	2	3	4
1	2	5	2
3	2	5	4

Renaming function

let U be an attribute set

a Renaming of the attributes of U is a function f

- from U to **att**
- written $A_1, \ldots, A_n \to B_1, \ldots, B_n$
- $f(A_i) = B_i$ for $i \in [1, n]$
- such that all the B_i's are pairwise distinct

Renaming operation

let I be an instance, f be a Renaming function from $sort(I)$ to **att**

$$\rho_f(I) = \{t \text{ over } f[sort(I)] | \text{ for } u \in I, \\ t(f(A)) = u(A) \text{ for all } A \in sort(I)\}$$

syntax and normal form

SPJR syntax is defined analogously as SPC syntax

syntax and normal form

SPJR syntax is defined analogously as SPC syntax

any SPJR query can be written under the form:

$$\pi_{B_1,\dots,B_n}(\{(A_1 : a_1)\} \bowtie \dots \bowtie \{(A_m : a_m)\} \bowtie \sigma_{\varphi}(\rho_{f_1}(R_1) \bowtie \dots \bowtie \rho_{f_k}(R_k)))$$

normal form

in this normal form:

$$\pi_{B_1,\ldots,B_n}(\{(A_1 : a_1)\} \bowtie \ldots \bowtie \{(A_m : a_m)\} \bowtie \sigma_\varphi(\rho_{f_1}(R_1) \bowtie \ldots \bowtie \rho_{f_k}(R_k)))$$

- $a_1, \ldots, a_m \in \mathbf{dom}$,
- the R_i are relation names,
- φ is a conjunctive selection formula,
- the A_i are distinct and appear in the B_j
- the f_j are Renamings over $\mathit{sorte}(R_j)$
- the A_i do not appear among $\rho_{f_j}(R_j)$
- the sorts of the $\rho_{f_j}(R_j)$ are pairwise disjoint

equivalences

theorem:

the SPC algebra and the SPJR algebra are equivalent

q is an SPC query iff q is an SPJR query

equivalences

we must how that:

1. any SPC query can be written as an SPJR query
2. any SPJR query can be written as an SPC query

sketch of the proof for 1

any SPC query q can be put under normal form

$$q = \pi_{j_1,\ldots,j_n}(\{(a_1)\} \times \ldots \times \{(a_m)\} \times \sigma_\varphi(R_1 \times \ldots \times R_k))$$

let's write an SPJR query q' equivalent to q

as any SPJR query, q' can be written under normal form:

$$q' = \pi_{B_1,\ldots,B_n}(\{(A_1 : a_1)\} \bowtie \ldots \bowtie \{(A_m : a_m)\} \bowtie \sigma_{\varphi'}(\rho_{f_1}(R_1) \bowtie \ldots \bowtie \rho_{f_k}(R_k)))$$

sketch of the proof for 1

what is the problem?

we have

$(R_1 \times \ldots \times R_k)$ that must be expressed by $\rho_{f_1}(R_1) \bowtie \ldots \bowtie \rho_{f_k}(R_k)$

and $\sigma_\varphi(R_1 \times \ldots \times R_k)$ to be expressed by
$\sigma_{\varphi'}(\rho_{f_1}(R_1) \bowtie \ldots \bowtie \rho_{f_k}(R_k))$

with φ a formula over the positions of $(R_1 \times \ldots \times R_k)$

sketch of the proof for 1

for $t \in [0, k]$, define $\beta(t) = m + \sum_{s=1}^{t} \text{arity}(R_s)$

let $A_{m+1}, \ldots, A_{\beta(k)}$ be new attributes

for $t \in [1, k]$, define Renaming functions f_t such that, for R_t, $f_t(B_i) = A_{\beta(t-1)+i}$ if B_i is the ith attribute of R_t

translate $(R_1 \times \ldots \times R_k)$ by $\rho_{f_1}(R_1) \bowtie \ldots \bowtie \rho_{f_k}(R_k)$

checking on an example (1)

assume $R_1[A, B, C]$ and $R_2[C, D]$ with $A_1, \ldots, A_5$ new attributes

le q be the query $q = \sigma_{3=4}(R_1 \times R_2)$

R_1 of arity 3, R_2 of arity 2 therefore $\beta(0) = 0, \beta(1) = 3, \beta(2) = 5$

$f_1(A) = A_1, f_1(B) = A_2, f_1(C) = A_3, f_2(C) = A_4, f_2(D) = A_5$

translate $(R_1 \times R_2)$ by $\rho_{f_1}(R_1) \bowtie \rho_{f_2}(R_2)$

sketch of the proof for 1

assume function γ from $[1, \beta(k)] \cup \mathbf{dom}$ to $\{A_{m+1}, \ldots, A_{\beta(k)}\} \cup \mathbf{dom}$

defined by

- $\gamma(j) = A_{m+j}$ if $j \in [1, \beta(k)]$
- $\gamma(a) = a$ if $a \in \mathbf{dom}$
- $\gamma(\alpha_1 = \alpha_2) = (\gamma(\alpha_1) = \gamma(\alpha_2))$
- $\gamma(\alpha_1 \wedge \alpha_2) = \gamma(\alpha_1) \wedge \gamma(\alpha_2)$

replace σ_φ by $\sigma_{\gamma(\varphi)}$

checking on an example (2)

for $q = \sigma_{3=4}(R_1 \times R_2)$,

translate $(R_1 \times R_2)$ by $\rho_{f_1}(R_1) \bowtie \rho_{f_2}(R_2)$

define γ such that
$\gamma(1) = A_1, \gamma(2) = A_2, \gamma(3) = A_3, \gamma(4) = A_4, \gamma(5) = A_5$

translate $\sigma_{3=4}$ by $\sigma_{A_3=A_4}$

sketch of the proof for 1

translating the projection is straightforward

Quod Erat Demonstrandum

equivalences

for satisfiable conjunctive queries, the following languages are equivalent

1. SPC algebra
2. SPJR algebra
3. rule-base language
4. conjunctive calculus

sketch of the proof

we already know that the rule-based language and the conjunctive calculus are equivalent

sketch of the proof

we already know that the rule-based language and the conjunctive calculus are equivalent

we have to show that any rule

$$résultat(\overrightarrow{x}) \leftarrow R_1(\overrightarrow{x_1}), \ldots, R_k(\overrightarrow{x_k})$$

can be written

$$\pi_{j_1,\ldots,j_n}(\{(a_1)\} \times \ldots \times \{(a_m)\} \times \sigma_\varphi(R_1 \times \ldots \times R_k))$$

sketch of the proof

it is sufficient to

1. do the cartesian product $R_1 \times \ldots \times R_k$
2. do the selection of
   - the constantes appearing in $\overrightarrow{x_1}, \ldots, \overrightarrow{x_k}$
   - the variables repeated in $\overrightarrow{x_1}, \ldots, \overrightarrow{x_k}$
3. do the cartesian product for the constants appearing in $\overrightarrow{x}$
4. project on positions corresponding $\overrightarrow{x}$

checking on an example

$$answer(x, y, 1) \leftarrow R_1(x, y), R_2(y, 1, z)$$

1. the Cartesian product : $R_1 \times R_2$
2. the Selection : $\sigma_{2=3 \wedge 4=''1''}(R_1 \times R_2)$
3. the Cartesian product with constants :
   $\{''1''\} \times \sigma_{2=3 \wedge 4=''1''}(R_1 \times R_2)$
4. the Projection : $\pi_{2,3,1}(\{''1''\} \times \sigma_{2=3 \wedge 4=''1''}(R_1 \times R_2))$