POPL2 class 2020 (2020-05-07)

Logic and data bases

Database

- *query language* for retrieval of information in a database.
- Let *D*1;*D*2,…,*Dn* be collections of symbols called *domains*.
- A *database relation R* over the domains *D*1,…,*Dn* is a subset of *D*1×… × *Dn*.
- *relational database* is a finite number of such (finite) relations.
- *MALE* := *{adam; bill}*, *FEMALE* := *{anne; beth}* and *PERSON* := *MALE U FEMALE*

$$MALE \times PERSON = \left\{ \begin{array}{ll} \langle adam, adam \rangle & \langle bill, adam \rangle \\ \langle adam, bill \rangle & \langle bill, bill \rangle \\ \langle adam, anne \rangle & \langle bill, anne \rangle \\ \langle adam, beth \rangle & \langle bill, beth \rangle \end{array} \right\}$$

Relational database

- *FATHER*, *MOTHER* and *PARENT* be relations over the domains *MALE* × *PERSON*, *FEMALE* × *PERSON* and *PERSON* × *PERSON*

$$
\begin{aligned}
FATHER &:= \{\langle adam, bill\rangle, \langle adam, beth\rangle\} \\
MOTHER &:= \{\langle anne, bill\rangle, \langle anne, beth\rangle\} \\
PARENT &:= \{\langle adam, bill\rangle, \langle adam, beth\rangle, \langle anne, bill\rangle, \langle anne, beth\rangle\}
\end{aligned}
$$

$FATHER$:

C_1	C_2
$adam$	$bill$
$adam$	$beth$

$MOTHER$:

C_1	C_2
$anne$	$bill$
$anne$	$beth$

$PARENT$:

C_1	C_2
$adam$	$bill$
$adam$	$beth$
$anne$	$bill$
$anne$	$beth$

Relational Database

Logic Program

$$
\begin{aligned}
&father(adam, bill).\\
&father(adam, beth).\\
&mother(anne, bill).\\
&mother(anne, beth).\\
&parent(adam, bill).\\
&parent(adam, beth).\\
&parent(anne, bill).\\
&parent(anne, beth).
\end{aligned}
$$

Relational data base

- notation *R*(*A*1*;A*2*; : : :;An*) will be used to describe the name, *R*, and attributes, *A*1*;A*2*; : : : ; An*, of a database table (i.e. relation).
- *R*(*A*1*;A*2*; : : :;An*) is sometimes called a *relation scheme*.
- Type is important in relational database : restricted set of values which may occur in each column/argument-position due to domain

$$father(anne, adam).$$

- Any relational database can be represented as a logic program.

Deductive database

- Database can be represented using rules and variables
- part of a logic program which consists of rules and nonground facts is called the *intensional database* (IDB).
- Atomic formulas *deduced* from explicit facts, logic programs are often
- referred to as *deductive databases*.

$$parent(X, Y) \leftarrow father(X, Y).$$
$$parent(X, Y) \leftarrow mother(X, Y).$$
$$father(adam, bill).$$
$$father(adam, beth).$$
$$mother(anne, bill).$$
$$mother(anne, beth).$$

datalog programs

the program solely consists of constant symbols and variables.

Deductive database

$$grandparent(X, Z) \leftarrow parent(X, Y), parent(Y, Z).$$

$$parent(X, Y) \leftarrow father(X, Y).$$
$$parent(X, Y) \leftarrow mother(X, Y).$$

intensional part

$$father(adam, bill).$$
$$father(adam, beth).$$
$$father(bill, cathy).$$
$$father(donald, eric).$$

$$mother(anne, bill).$$
$$mother(anne, beth).$$
$$mother(cathy, donald).$$
$$mother(diana, eric).$$

$$female(anne).$$
$$female(beth).$$
$$female(cathy).$$
$$female(diana).$$

$$male(adam).$$
$$male(bill).$$
$$male(donald).$$
$$male(eric).$$

extensional part

type declarations

Deductive data base with type declaration

$$person(X) \leftarrow male(X).$$
$$person(X) \leftarrow female(X).$$

$$parent(X, Y) \leftarrow person(X), person(Y), father(X, Y).$$
$$parent(X, Y) \leftarrow person(X), person(Y), mother(X, Y).$$

$$father(adam, bill) \leftarrow male(adam), person(bill).$$
$$father(adam, beth) \leftarrow male(adam), person(beth).$$
$$\vdots$$

$$person(X) \leftarrow male(X).$$
$$person(X) \leftarrow female(X).$$

Relational algebra

- View is a relation which is not explicitly stored in the database, butis created by means of operations on existing database relations by using *query-language : relational algebra*
- all standard operations of relational algebra can be mimicked in logic programming
- primitive operations of relational algebra are *union*, *set dierence*, *cartesian product*, *projection* and *selection*.

Relational algebra

union of two relations $\{\langle x_1, \ldots, x_n \rangle \mid \langle x_1, \ldots, x_n \rangle \in R_1 \vee \langle x_1, \ldots, x_n \rangle \in R_2\}$

$r(X_1, \ldots, X_n) \leftarrow r_1(X_1, \ldots, X_n).$
$r(X_1, \ldots, X_n) \leftarrow r_2(X_1, \ldots, X_n).$

$parent(X, Y) \leftarrow father(X, Y).$
$parent(X, Y) \leftarrow mother(X, Y).$

difference $R_1 \setminus R_2$ of two relations R_1 and R_2

$$\{\langle x_1, \ldots, x_n \rangle \in R_1 \mid \langle x_1, \ldots, x_n \rangle \notin R_2\}$$

$$r(X_1, \ldots, X_n) \leftarrow r_1(X_1, \ldots, X_n), not\ r_2(X_1, \ldots, X_n).$$

$$father(X, Y) \leftarrow parent(X, Y), not\ mother(X, Y).$$

Relational algebra

cartesian product of two relations R_1 and R_2 (denoted $R_1 \times R_2$)

$$\{\langle x_1, \ldots, x_m, y_1, \ldots, y_n \rangle \mid \langle x_1, \ldots, x_m \rangle \in R_1 \wedge \langle y_1, \ldots, y_n \rangle \in R_2\}$$

$$r(X_1, \ldots, X_m, Y_1, \ldots, Y_n) \leftarrow r_1(X_1, \ldots, X_m), r_2(Y_1, \ldots, Y_n).$$

$$couple(X, Y) \leftarrow male(X), female(Y).$$

Projection can be seen as the deletion and/or rearrangement

$\pi_F(FATHER(F, C))$ $\{\langle x_1 \rangle \mid \langle x_1, x_2 \rangle \in FATHER\}$ $father(X) \leftarrow father(X, Y).$

Relational algebra

- Selection operation
- *selection* of a relation *R with formula F is* $\sigma_F(R)$

$$\langle x_1, \ldots, x_n \rangle \in R \text{ such that } \text{“}F \text{ is true for } \langle x_1, \ldots, x_n \rangle \text{”}.$$

$$\sigma_{Y \geq 1{,}000{,}000}\, INCOME(X, Y) \quad millionaire(X, Y) \leftarrow income(X, Y), Y \geq 1000000.$$

Logic as a Query-language retrieve information from the database.

$$\leftarrow parent(adam, X). \qquad \leftarrow mother(X, Y).$$

$$\begin{array}{ll} X = anne, & Y = bill \\ X = anne, & Y = beth \\ X = cathy, & Y = donald \\ X = diana, & Y = eric \end{array}$$

Relational algebra

R is said to be *reflexive* iff for all $x \in \mathcal{D}$, it holds that $\langle x, x \rangle \in R$;

$$r(X, X) \leftarrow t(X).$$

"every person looks like himself"

$$looks\_like(X, X) \leftarrow person(X).$$
$$person(bill).$$
$$person(kate).$$
$$abstract(love).$$

R is *symmetric* iff $\langle x, y \rangle \in R$ implies that $\langle y, x \rangle \in R$; $r(X, Y) \leftarrow r(Y, X).$

$\{sarah, diane, pamela, simon, david, peter\}$

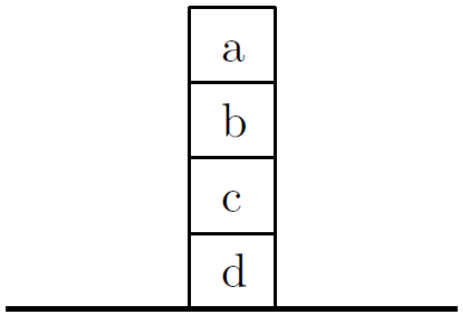
$$married(X, Y) \leftarrow married(Y, X).$$
$$married(sarah, simon).$$
$$married(diane, david).$$
$$married(pamela, peter).$$

Relational Algebra

R is *transitive* iff $\langle x, y\rangle \in R$ and $\langle y, z\rangle \in R$ implies that $\langle x, z\rangle \in R$;

$r(X, Z) \leftarrow r(X, Y), r(Y, Z).$

"... is positioned over ..."



extensional database:

$over(a, b).$ $over(a, c).$
$over(a, d).$ $over(b, c).$
$over(b, d).$ $over(c, d).$

deductive database:

$over(X, Z) \leftarrow over(X, Y), over(Y, Z).$
$over(a, b).$
$over(b, c).$
$over(c, d).$