

Fondamenti Logici dell'Informatica

Corso di Laurea Magistrale in Informatica

Logica e Basi di Dati

Ugo Dal Lago



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

inria informatiques mathématiques

Anno Accademico 2017-2018

Il Modello Relazionale

- ▶ È il modello logico di gran lunga più comune tra quelli utilizzati per organizzare le basi di dati.
- ▶ Nonostante esistano indizi già negli anni Sessanta, la sua vera e propria introduzione si deve a Codd, in un celebre articolo del 1970.
- ▶ Nel modello relazionale, come in tutti i modelli logici, un concetto centrale è quello di *interrogazione*, o *query*.
- ▶ In questa parte, ci prefiggiamo di studiare le varie tecniche per definire interrogazioni, e di indagare il loro potere espressivo.
- ▶ Particolare attenzione verrà data agli aspetti di natura logica.

Relazione Ordinate

- ▶ Supponiamo di lavorare con dei **domini** come i seguenti:
 - ▶ L'insieme dei numeri naturali, \mathbb{N} ;
 - ▶ L'insieme delle stringhe in un alfabeto Σ , ovvero Σ^* ;
 - ▶ L'insieme dei valori booleani $\mathbb{T} = \{0, 1\}$.

Indichiamo un generico dominio con D .

Relazione Ordinate

- ▶ Supponiamo di lavorare con dei **domini** come i seguenti:
 - ▶ L'insieme dei numeri naturali, \mathbb{N} ;
 - ▶ L'insieme delle stringhe in un alfabeto Σ , ovvero Σ^* ;
 - ▶ L'insieme dei valori booleani $\mathbb{T} = \{0, 1\}$.

Indichiamo un generico dominio con D .

- ▶ Una **relazione** può prima di tutto essere vista come un sottoinsieme *finito* \mathcal{R} di $D_1 \times \dots \times D_n$ dove i D_i sono *domini*.
 - ▶ \mathcal{R} , in altre parole, *vive* nell'insieme delle parti finite di $D_1 \times \dots \times D_n$:

$$\mathbf{P}_{fn}\left(\prod_{1 \leq i \leq n} D_i\right) = \mathbf{P}_{fn}(D_1 \times \dots \times D_n)$$

- ▶ Una relazione, in altre parole, è un insieme finito di n -uple nella forma (d_1, \dots, d_n) dove $d_i \in D_i$ per ogni $1 \leq i \leq n$.
- ▶ Per quello che diremo in questa parte del corso possiamo tranquillamente supporre che esista un unico E tale che $E = D_i$ per ogni $1 \leq i \leq n$.

Relazioni Ordinate — Esempio

<i>ASCII</i> *	<i>ASCII</i> *	N	T
Rossi	Mario	1973	0
Verdi	Carlo	1978	1
Gialli	Luca	1980	1
Bianchi	Andrea	1971	0

Relazioni Ordinate — Esempio

E	E	E	E
Rossi	Mario	1973	0
Verdi	Carlo	1978	1
Gialli	Luca	1980	1
Bianchi	Andrea	1971	0

$$E = ASCII^* \uplus \mathbb{N} \uplus \mathbb{T}.$$

Relazioni Non Ordinate

- ▶ Spesso conviene dare alle “colonne” di una relazione un *nome*, e non solo una *posizione*.
- ▶ Una relazione diventerebbe quindi:
 - ▶ Un insieme finito $\{f_1, \dots, f_n\}$;
 - ▶ dove $f_i : C \rightarrow D$;
 - ▶ e C è un insieme finito di campi.

Relazioni Non Ordinate

- ▶ Spesso conviene dare alle “colonne” di una relazione un *nome*, e non solo una *posizione*.
- ▶ Una relazione diventerebbe quindi:
 - ▶ Un insieme finito $\{f_1, \dots, f_n\}$;
 - ▶ dove $f_i : C \rightarrow D$;
 - ▶ e C è un insieme finito di campi.
- ▶ La nostra relazione d'esempio diventerebbe $\{f_1, f_2, f_3, f_4\}$ dove l'insieme C è, per esempio, $\{COGNOME, NOME, ANNO, SOCIO\}$ e il valore delle quattro funzioni f_1, \dots, f_4 è descritto dalla seguente tabella:

	<i>COGNOME</i>	<i>NOME</i>	<i>ANNO</i>	<i>SOCIO</i>
$f_1(\cdot)$	Rossi	Mario	1973	0
$f_2(\cdot)$	Verdi	Carlo	1978	1
$f_3(\cdot)$	Gialli	Luca	1980	1
$f_4(\cdot)$	Bianchi	Andrea	1971	0

Equivalenza tra le Due Nozioni

- ▶ Ogni relazione **ordinata** può essere trasformata in una relazione **non ordinata**.
 - ▶ Data $\mathcal{R} \subseteq D^n = D \times \dots \times D$, è sufficiente *dare un nome* a ciascun intero compreso tra 1 ed n .

Equivalenza tra le Due Nozioni

- ▶ Ogni relazione **ordinata** può essere trasformata in una relazione **non ordinata**.
 - ▶ Data $\mathcal{R} \subseteq D^n = D \times \dots \times D$, è sufficiente *dare un nome* a ciascun intero compreso tra 1 ed n .
- ▶ Ogni relazione **non ordinata** può essere trasformata in una relazione **ordinata**.
 - ▶ Data una relazione non ordinata su un insieme finito di campi C , basterà fissare un *ordine totale* su C .

Equivalenza tra le Due Nozioni

- ▶ Ogni relazione **ordinata** può essere trasformata in una relazione **non ordinata**.
 - ▶ Data $\mathcal{R} \subseteq D^n = D \times \dots \times D$, è sufficiente *dare un nome* a ciascun intero compreso tra 1 ed n .
- ▶ Ogni relazione **non ordinata** può essere trasformata in una relazione **ordinata**.
 - ▶ Data una relazione non ordinata su un insieme finito di campi C , basterà fissare un *ordine totale* su C .
- ▶ I dettagli sono un piacevole *esercizio*.

Equivalenza tra le Due Nozioni

- ▶ Ogni relazione **ordinata** può essere trasformata in una relazione **non ordinata**.
 - ▶ Data $\mathcal{R} \subseteq D^n = D \times \dots \times D$, è sufficiente *dare un nome* a ciascun intero compreso tra 1 ed n .
- ▶ Ogni relazione **non ordinata** può essere trasformata in una relazione **ordinata**.
 - ▶ Data una relazione non ordinata su un insieme finito di campi C , basterà fissare un *ordine totale* su C .
- ▶ I dettagli sono un piacevole *esercizio*.
- ▶ Passeremo da una rappresentazione all'altra molto liberamente, sapendo che sono assolutamente equivalenti.

Query Come Funzioni

- ▶ Che funzione *calcola* una query?

Query Come Funzioni

- ▶ Che funzione *calcola* una query?
- ▶ **Idea**
 - ▶ La base di dati su cui la query opera è vista come una sequenza di relazioni.
 - ▶ Il risultato della query deve essere anch'esso una relazione.

Query Come Funzioni

- ▶ Che funzione *calcola* una query?
- ▶ **Idea**
 - ▶ La base di dati su cui la query opera è vista come una sequenza di relazioni.
 - ▶ Il risultato della query deve essere anch'esso una relazione.
- ▶ In altre parole, la funzione $\llbracket Q \rrbracket$ calcolata da una query Q dovrebbe avere la forma seguente:

$$\llbracket Q \rrbracket : \mathbf{P}_{fin}(D^{n_1}) \times \dots \times \mathbf{P}_{fin}(D^{n_k}) \rightarrow \mathbf{P}_{fin}(D^m)$$

Query Come Funzioni

- ▶ Che funzione *calcola* una query?
- ▶ **Idea**
 - ▶ La base di dati su cui la query opera è vista come una sequenza di relazioni.
 - ▶ Il risultato della query deve essere anch'esso una relazione.
- ▶ In altre parole, la funzione $\llbracket Q \rrbracket$ calcolata da una query Q dovrebbe avere la forma seguente:

$$\llbracket Q \rrbracket : \mathbf{P}_{fin}(D^{n_1}) \times \dots \times \mathbf{P}_{fin}(D^{n_k}) \rightarrow \mathbf{P}_{fin}(D^m)$$

- ▶ Nel momento in cui progettiamo un linguaggio per scrivere query, quindi, dovremmo essere sicuri che, almeno, le query possano avere questa semantica.

Query Come Funzioni

- ▶ Che funzione *calcola* una query?
- ▶ **Idea**
 - ▶ La base di dati su cui la query opera è vista come una sequenza di relazioni.
 - ▶ Il risultato della query deve essere anch'esso una relazione.
- ▶ In altre parole, la funzione $\llbracket Q \rrbracket$ calcolata da una query Q dovrebbe avere la forma seguente:

$$\llbracket Q \rrbracket : \mathbf{P}_{fin}(D^{n_1}) \times \dots \times \mathbf{P}_{fin}(D^{n_k}) \rightarrow \mathbf{P}_{fin}(D^m)$$

- ▶ Nel momento in cui progettiamo un linguaggio per scrivere query, quindi, dovremmo essere sicuri che, almeno, le query possano avere questa semantica.
- ▶ Domanda importante: *che funzioni vogliamo* che il nostro linguaggio per interrogazioni calcoli? In altre parole, che *espressività* dovrebbe avere tale linguaggio?

Algebra Relazionale — Sintassi

- ▶ La sintassi dell'algebra relazionale sull'insieme di simboli relazionali $\{R_1, \dots, R_k\}$ è la seguente:

$$Q ::= R_i \mid Q \cup P \mid Q - P \mid Q \times P \mid \pi_\ell(Q) \mid \sigma_c(Q)$$

$$c ::= i \leq j \mid i = j \mid \neg c \mid c \wedge d \mid c \vee d$$

dove:

- ▶ i e j sono numeri naturali positivi.
- ▶ ℓ è una sequenza di numeri naturali positivi, ossia un elemento di \mathbb{N}_+^* .

Algebra Relazionale — Sintassi

- ▶ La sintassi dell'algebra relazionale sull'insieme di simboli relazionali $\{R_1, \dots, R_k\}$ è la seguente:

$$\begin{aligned} Q &::= R_i \mid Q \cup P \mid Q - P \mid Q \times P \mid \pi_\ell(Q) \mid \sigma_c(Q) \\ c &::= i \leq j \mid i = j \mid \neg c \mid c \wedge d \mid c \vee d \end{aligned}$$

dove:

- ▶ i e j sono numeri naturali positivi.
- ▶ ℓ è una sequenza di numeri naturali positivi, ossia un elemento di \mathbb{N}_+^* .
- ▶ Un'interrogazione Q soddisfa i **vincoli di integrità** se i numeri interi che occorrono in essa sono coerenti con la relazioni cui si riferiscono. Ad esempio:
 - ▶ Nella query $\pi_\ell(Q)$, tutti i numeri in ℓ devono essere in accordo con l'arietà di Q .
 - ▶ Nella query $Q \cup P$, le arietà di Q e P devono essere identiche.
- ▶ Nel seguito, si supporrà che tutte le interrogazioni con cui lavoreremo soddisfino i vincoli di integrità.

Algebra Relazionale — Semantica

- ▶ Data una query Q su $\{R_1, \dots, R_k\}$, vogliamo definire la sua semantica:

$$\llbracket Q \rrbracket : \mathbf{P}_{fin}(D^{n_1}) \times \dots \times \mathbf{P}_{fin}(D^{n_k}) \rightarrow \mathbf{P}_{fin}(D^m)$$

dove n_1, \dots, n_k sono le arietà di R_1, \dots, R_k e m è l'arietà di Q .

Algebra Relazionale — Semantica

- ▶ Data una query Q su $\{R_1, \dots, R_k\}$, vogliamo definire la sua semantica:

$$\llbracket Q \rrbracket : \mathbf{P}_{fin}(D^{n_1}) \times \dots \times \mathbf{P}_{fin}(D^{n_k}) \rightarrow \mathbf{P}_{fin}(D^m)$$

dove n_1, \dots, n_k sono le arietà di R_1, \dots, R_k e m è l'arietà di Q .

- ▶ La funzione $\llbracket Q \rrbracket$ è definita per induzione sulla struttura di Q .
- ▶ Se Q è semplicemente R_i , allora $\llbracket Q \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k)$ è semplicemente \mathcal{R}_i .
- ▶ Gli operatori \cup , $-$ e \times hanno un'interpretazione insiemistica naturale. Ad esempio:

$$\llbracket Q \cup P \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k) = \llbracket Q \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k) \cup \llbracket P \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k)$$

- ▶ Rimane solo da vedere cosa succede per proiezione e selezione.

- ▶ Nell'operatore di proiezione, interviene una lista di interi ℓ , che indica semplicemente quali campi considerare nella proiezione.
- ▶ Formalmente: $\llbracket \pi_{i_1, \dots, i_s}(\mathbf{Q}) \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k)$ sarà l'insieme

$$\{(d_{i_1}, \dots, d_{i_s}) \mid (d_1, \dots, d_n) \in \llbracket \mathbf{Q} \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k)\}$$

- ▶ Da osservare come i vincoli di integrità risultino in questa fase assolutamente cruciali.

- ▶ Nell'operatore di selezione, interviene invece una condizione c , che indica quali *tuple* considerare nella selezione.
- ▶ Data una tupla di valori $t = (d_1, \dots, d_n)$ e una condizione c , possiamo definire quando quest'ultima è *soddisfatta* in (d_1, \dots, d_n) , per induzione. Ad esempio:
 - ▶ $(d_1, \dots, d_n) \vdash i = j$ sse $d_i = d_j$;
 - ▶ $t \vdash c \wedge d$ sse $t \vdash c$ e $t \vdash d$.
- ▶ A questo punto definire la semantica dell'operatore di selezione è semplice. La relazione $\llbracket \sigma_c(Q) \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k)$ sarà

$$\{t \mid t \in \llbracket Q \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k) \wedge t \vdash c\}$$

- ▶ Di nuovo, i vincoli di integrità risultano in questa fase importanti.

Algebra Relazionale — Un Esempio

\mathcal{R}_1

0012	Rossi	Mario	1973	0
1492	Verdi	Carlo	1978	1
9834	Gialli	Luca	1980	1
7511	Bianchi	Andrea	1971	0

Algebra Relazionale — Un Esempio

 \mathcal{R}_1

0012	Rossi	Mario	1973	0
1492	Verdi	Carlo	1978	1
9834	Gialli	Luca	1980	1
7511	Bianchi	Andrea	1971	0

 \mathcal{R}_2

0012	1492	3	2
1492	7511	1	3
9834	7511	0	3

Algebra Relazionale — Un Esempio

 \mathcal{R}_1

0012	Rossi	Mario	1973	0
1492	Verdi	Carlo	1978	1
9834	Gialli	Luca	1980	1
7511	Bianchi	Andrea	1971	0

 \mathcal{R}_2

0012	1492	3	2
1492	7511	1	3
9834	7511	0	3

- ▶ Le relazioni di cui sopra sono la base di dati di una società di tennis. La prima tiene traccia dei soci, la seconda dei risultati delle partite.
- ▶ Vogliamo costruire una query che determini gli anni di nascita dei *soci* che hanno vinto almeno una partita.

Algebra Relazionale — Un Esempio

 \mathcal{R}_1

0012	Rossi	Mario	1973	0
1492	Verdi	Carlo	1978	1
9834	Gialli	Luca	1980	1
7511	Bianchi	Andrea	1971	0

 \mathcal{R}_2

0012	1492	3	2
1492	7511	1	3
9834	7511	0	3

- ▶ Le relazioni di cui sopra sono la base di dati di una società di tennis. La prima tiene traccia dei soci, la seconda dei risultati delle partite.
- ▶ Vogliamo costruire una query che determini gli anni di nascita dei *soci* che hanno vinto almeno una partita.
- ▶ Basta la query:

$$\pi_4(\sigma_{(1=6) \wedge (8 > 9)}(R_1 \times R_2)) \cup \pi_4(\sigma_{(1=7) \wedge (9 > 8)}(R_1 \times R_2))$$

- ▶ Osserviamo come, ad esempio, la condizione $8 > 9$ non sia esprimibile direttamente, ma debba essere scritta nella forma $8 \geq 9 \wedge \neg(8 = 9)$.

Algebra Relazionale — Potere Espressivo

- Possiamo a questo punto definire l'insieme delle funzioni che l'algebra relazionale ci permette di catturare:

$$\mathcal{AR} = \{ \llbracket Q \rrbracket \mid Q \text{ è una query ben formata} \}$$

Algebra Relazionale — Potere Espressivo

- ▶ Possiamo a questo punto definire l'insieme delle funzioni che l'algebra relazionale ci permette di catturare:

$$\mathcal{AR} = \{ \llbracket Q \rrbracket \mid Q \text{ è una query ben formata} \}$$

- ▶ Ci chiediamo: quanto grande è \mathcal{AR} ? Corrisponde magari all'insieme delle funzioni calcolabili con certi limitazioni di tempo o spazio?

Algebra Relazionale — Potere Espressivo

- ▶ Possiamo a questo punto definire l'insieme delle funzioni che l'algebra relazionale ci permette di catturare:

$$\mathcal{AR} = \{ \llbracket Q \rrbracket \mid Q \text{ è una query ben formata} \}$$

- ▶ Ci chiediamo: quanto grande è \mathcal{AR} ? Corrisponde magari all'insieme delle funzioni calcolabili con certe limitazioni di tempo o spazio?
- ▶ Ma ci interessa catturare una classe di complessità ampia?

Algebra Relazionale — Potere Espressivo

- ▶ Possiamo a questo punto definire l'insieme delle funzioni che l'algebra relazionale ci permette di catturare:

$$\mathcal{AR} = \{ \llbracket Q \rrbracket \mid Q \text{ è una query ben formata} \}$$

- ▶ Ci chiediamo: quanto grande è \mathcal{AR} ? Corrisponde magari all'insieme delle funzioni calcolabili con certe limitazioni di tempo o spazio?
- ▶ Ma ci interessa catturare una classe di complessità ampia?
- ▶ Vogliamo veramente fare in modo che le query possano, anche solo ipoteticamente, richiedere, per esempio, spazio lineare? O tempo più che lineare?

Calcolo Relazionale — Sintassi

- ▶ Un modo naturale per lavorare con le relazioni è certamente la *logica predicativa*, che conosciamo bene.
- ▶ Supponiamo di voler:
 - ▶ **interrogare** una base di dati che consti delle relazioni $\mathcal{R}_1, \dots, \mathcal{R}_k$ aventi arietà n_1, \dots, n_k .
 - ▶ ottenendo come **risultato** una relazione Q di arietà m .

Calcolo Relazionale — Sintassi

- ▶ Un modo naturale per lavorare con le relazioni è certamente la *logica predicativa*, che conosciamo bene.
- ▶ Supponiamo di voler:
 - ▶ **interrogare** una base di dati che consti delle relazioni $\mathcal{R}_1, \dots, \mathcal{R}_k$ aventi arietà n_1, \dots, n_k .
 - ▶ ottenendo come **risultato** una relazione \mathcal{Q} di arietà m .
- ▶ Basterà costruire una formula predicativa F nel modo seguente:
 - ▶ Gli unici simboli funzionali sono delle *costanti* che indicano gli elementi di D , mentre i simboli predicativi saranno $\mathcal{R}_1, \dots, \mathcal{R}_k$, più i simboli \leq e $=$, questi ultimi preinterpretati.
 - ▶ Le variabili che occorrono libere in F dovranno essere incluse nell'insieme $\{f_1, \dots, f_m\}$ e corrispondere ciascuna ad un campo della relazione \mathcal{Q} .

Calcolo Relazionale — Sintassi

- ▶ Un modo naturale per lavorare con le relazioni è certamente la *logica predicativa*, che conosciamo bene.
- ▶ Supponiamo di voler:
 - ▶ **interrogare** una base di dati che consti delle relazioni $\mathcal{R}_1, \dots, \mathcal{R}_k$ aventi arietà n_1, \dots, n_k .
 - ▶ ottenendo come **risultato** una relazione Q di arietà m .
- ▶ Basterà costruire una formula predicativa F nel modo seguente:
 - ▶ Gli unici simboli funzionali sono delle *costanti* che indicano gli elementi di D , mentre i simboli predicativi saranno R_1, \dots, R_k , più i simboli \leq e $=$, questi ultimi preinterpretati.
 - ▶ Le variabili che occorrono libere in F dovranno essere incluse nell'insieme $\{f_1, \dots, f_m\}$ e corrispondere ciascuna ad un campo della relazione Q .
- ▶ Abbiamo appena definito il **calcolo relazionale**, ossia un frammento della logica predicativa.

Calcolo Relazionale — Un Esempio

\mathcal{R}_1

0012	Rossi	Mario	1973	0
1492	Verdi	Carlo	1978	1
9834	Gialli	Luca	1980	1
7511	Bianchi	Andrea	1971	0

Calcolo Relazionale — Un Esempio

 \mathcal{R}_1

0012	Rossi	Mario	1973	0
1492	Verdi	Carlo	1978	1
9834	Gialli	Luca	1980	1
7511	Bianchi	Andrea	1971	0

 \mathcal{R}_2

0012	1492	3	2
1492	7511	1	3
9834	7511	0	3

Calcolo Relazionale — Un Esempio

 \mathcal{R}_1

0012	Rossi	Mario	1973	0
1492	Verdi	Carlo	1978	1
9834	Gialli	Luca	1980	1
7511	Bianchi	Andrea	1971	0

 \mathcal{R}_2

0012	1492	3	2
1492	7511	1	3
9834	7511	0	3

- Vogliamo costruire una formula del calcolo relazionale che catturi gli anni di nascita dei vincitori.

Calcolo Relazionale — Un Esempio

 \mathcal{R}_1

0012	Rossi	Mario	1973	0
1492	Verdi	Carlo	1978	1
9834	Gialli	Luca	1980	1
7511	Bianchi	Andrea	1971	0

 \mathcal{R}_2

0012	1492	3	2
1492	7511	1	3
9834	7511	0	3

- ▶ Vogliamo costruire una formula del calcolo relazionale che catturi gli anni di nascita dei vincitori.
- ▶ Un esempio potrebbe essere:

$$\begin{aligned} & \exists p. \exists s. \exists c. \exists n. \exists o. \exists pp. \exists ps. R_1(p, c, n, f, o) \wedge R_2(p, s, pp, ps) \wedge (pp > ps) \\ & \quad \vee \\ & \exists p. \exists s. \exists c. \exists n. \exists o. \exists pp. \exists ps. R_1(s, c, n, f, o) \wedge R_2(p, s, pp, ps) \wedge (ps > pp) \end{aligned}$$

Calcolo Relazionale — Semantica

- ▶ Osserviamo che:
 - ▶ L'universo in cui *interpretare* una formula F del calcolo relazionale è D .

Calcolo Relazionale — Semantica

- ▶ Osserviamo che:
 - ▶ L'universo in cui *interpretare* una formula F del calcolo relazionale è D .
 - ▶ Gli unici simboli che occorre *interpretare* per dare un semantica a F sono R_1, \dots, R_k . È quindi naturale vedere la base di dati $\{\mathcal{R}_1, \dots, \mathcal{R}_k\}$ come una tale interpretazione.

- ▶ Osserviamo che:
 - ▶ L'universo in cui *interpretare* una formula F del calcolo relazionale è D .
 - ▶ Gli unici simboli che occorre *interpretare* per dare un semantica a F sono R_1, \dots, R_k . È quindi naturale vedere la base di dati $\{\mathcal{R}_1, \dots, \mathcal{R}_k\}$ come una tale interpretazione.
 - ▶ Se vale che

$$(D, \{\mathcal{R}_1, \dots, \mathcal{R}_k\}), \xi \models F,$$

ciò significa che $(\xi(f_1), \dots, \xi(f_m))$ deve stare nella relazione Q .

► Osserviamo che:

- L'universo in cui *interpretare* una formula F del calcolo relazionale è D .
- Gli unici simboli che occorre *interpretare* per dare un semantica a F sono R_1, \dots, R_k . È quindi naturale vedere la base di dati $\{\mathcal{R}_1, \dots, \mathcal{R}_k\}$ come una tale interpretazione.
- Se vale che

$$(D, \{\mathcal{R}_1, \dots, \mathcal{R}_k\}), \xi \models F,$$

ciò significa che $(\xi(f_1), \dots, \xi(f_m))$ deve stare nella relazione Q .

- Di conseguenza, possiamo porre $\llbracket F \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k)$ pari a

$$\{(\xi(f_1), \dots, \xi(f_m)) \mid (D, \{\mathcal{R}_1, \dots, \mathcal{R}_k\}), \xi \models F\}.$$

Calcolo Relazionale — Semantica

► Osserviamo che:

- L'universo in cui *interpretare* una formula F del calcolo relazionale è D .
- Gli unici simboli che occorre *interpretare* per dare un semantica a F sono R_1, \dots, R_k . È quindi naturale vedere la base di dati $\{\mathcal{R}_1, \dots, \mathcal{R}_k\}$ come una tale interpretazione.
- Se vale che

$$(D, \{\mathcal{R}_1, \dots, \mathcal{R}_k\}), \xi \models F,$$

ciò significa che $(\xi(f_1), \dots, \xi(f_m))$ deve stare nella relazione Q .

► Di conseguenza, possiamo porre $\llbracket F \rrbracket(\mathcal{R}_1, \dots, \mathcal{R}_k)$ pari a

$$\{(\xi(f_1), \dots, \xi(f_m)) \mid (D, \{\mathcal{R}_1, \dots, \mathcal{R}_k\}), \xi \models F\}.$$

► In questo modo, $\llbracket F \rrbracket \subseteq D^m$, ma *non è detto che $\llbracket F \rrbracket$ sia finita!*

- Basti considerare la formula $F = (f_1 = f_1)$.

Il Calcolo Relazionale Sicuro

- ▶ Occorre quindi isolare un sottoinsieme delle formule del calcolo relazionale, che chiameremo **formule sicure**.

Il Calcolo Relazionale Sicuro

- ▶ Occorre quindi isolare un sottoinsieme delle formule del calcolo relazionale, che chiameremo **formule sicure**.
- ▶ Le formule sicure, descritte nella prossima trasparenza, hanno la proprietà che se F è sicura, allora ogni tupla in $\llbracket F \rrbracket$ contiene valori tra quelli che occorrono in F e quelli che troviamo nelle tuple in $\mathcal{R}_1, \dots, \mathcal{R}_k$.

Il Calcolo Relazionale Sicuro

- ▶ Occorre quindi isolare un sottoinsieme delle formule del calcolo relazionale, che chiameremo **formule sicure**.
- ▶ Le formule sicure, descritte nella prossima trasparenza, hanno la proprietà che se F è sicura, allora ogni tupla in $\llbracket F \rrbracket$ contiene valori tra quelli che occorrono in F e quelli che troviamo nelle tuple in $\mathcal{R}_1, \dots, \mathcal{R}_k$.
- ▶ Di conseguenza, $\llbracket F \rrbracket$ è sempre finita.
- ▶ Nel definire il calcolo relazionale sicuro, si utilizza spesso l'insieme delle variabili che *occorrono* libere in una formula F , che scriveremo $FV(F)$.

Il Calcolo Relazionale Sicuro

1. L'uso del quantificatore *universale* non è permesso.
2. Ogniqualvolta si utilizza l'operatore \vee per formare $F \vee G$, deve valere che $FV(F) = FV(G)$.
3. Se una sottoformula della formula data si può scrivere come $F_1 \wedge \dots \wedge F_m$, (dove $m \geq 1$ è massimale), allora ogni $x \in \cup_{1 \leq i \leq m} FV(F_i)$ devono essere *limitate*, ossia deve esistere *almeno una* formula F_j tale che:
 - 3.1 $x \in FV(F_j)$ e F_j non è un predicato aritmetico e non è nella forma $\neg G$.
 - 3.2 F_j è nella forma $x = c$ oppure $c = x$, dove c è una costante.
 - 3.3 F_j è nella forma $x = y$ dove y è anch'essa limitata.
4. L'unico uso permesso dell'operatore di negazione è in un *uno* delle formule F_j di una congiunzione $F_1 \wedge \dots \wedge F_m$ in cui vi sia almeno una delle F_p che *non* sia essa stessa negata.

Calcolo Relazionale Sicuro — Potere Espressivo

- ▶ Siamo finalmente in grado di definire l'insieme delle funzioni che il calcolo relazionale sicuro cattura:

$$\mathcal{CR} = \{\llbracket F \rrbracket \mid F \text{ è una formula sicura del calcolo relazionale}\}.$$

Calcolo Relazionale Sicuro — Potere Espressivo

- ▶ Siamo finalmente in grado di definire l'insieme delle funzioni che il calcolo relazionale sicuro cattura:

$$\mathcal{CR} = \{ \llbracket F \rrbracket \mid F \text{ è una formula sicura del calcolo relazionale} \}.$$

Teorema

$$\mathcal{AR} = \mathcal{CR}.$$