$\boxed{AR = CR}$

$AR \subseteq CR$

- QUEST'IMPLICAZIONE È PIÙ SEMPLICE DA DIMOSTRARE RISPETTO A $CR \subseteq AR$. LA PROVA È COSTRUTTIVA E PROCEDE PER INDUZIONE SULLA STRUTTURA DELL'ESPRESSIONE ALGEBRICA A CHE VOGLIAMO TRADURRE IN CALCOLO RELAZIONALE

A QUERY REAZIONALE DI ARIETÀ m E FACENTE RIFERIMENTO A $R_1, \dots, R_k$ $\Rightarrow$ F_A SICURA FACENTE RIFERIMENTO $R_1, R_2, \dots, R_k$ E CON m VARIABILI LIBERE $f_1, \dots, f_m$.

- OCCORRE, A PRIORI, DIRE QUALCOSA SULLE QUERY DI SELEZIONI: ALTRIMENTI NON RIUSCIREMMO A FAR FUNZIONARE L'INDUZIONE
- DEFINIAMO SEMPLICE UNA QUERY RELAZIONALE Q TALE CHE TUTTI GLI OPERATORI DI SELEZIONE $\sigma_c(R)$ IN Q SONO TALI CHE C È UN OPERATORE ARITMETICO ($i=j$ O $i<j$) O LA SUA NEGAZIONE.
- LEMMA

  PER OGNI Q ESISTE P SEMPLICE TALE CHE $[\![Q]\!] = [\![P]\!]$.

  DIMOSTRIAMOLO.

  - LE UGUAGLIANZE DI DE-MORGAN

  $$\neg(A \vee B) \sim \neg A \wedge \neg B \qquad \neg(A \wedge B) \sim \neg A \vee \neg B$$

  CI PERMETTONO DI ASSUMERE CHE TUTTE LE OCCORRENZE DI $\neg$ IN C PER OGNI OCCORRENZA DI $\sigma_c(R)$ IN Q SIANO IMMEDIATAMENTE VICINE AD UN OPERATORE

ARITMETICO.

- PROCEDIAMO, PRIMA DI PASSARE ALLA PROVA VERA E PROPRIA, A DIMOSTRARE UN ULTERIORE RISULTATO AUSILIARIO, OSSIA CHE SE Q È SEMPLICE, ALLORA ESISTE UNA QUERY SEMPLICE R EQUIVALENTE A $\sigma_c(Q)$. PROCEDIAMO PER INDUZIONE SU C:

PROVA DEL SOTTO-LEMMA AUSILIARIO.

- SE C È ARITMETICA OPPURE LA SUA NEGAZIONE, ALLORA $R = \sigma_c(Q)$ E INFATTI $[\![\sigma_c(Q)]\!] = [\![R]\!]$.
- SE C È NELLA FORMA $d \wedge e$ ALLORA POSSIAMO APPLICARE L'IPOTESI INDUTTIVA A d, OTTENENDO T SEMPLICE TALE CHE $[\![T]\!] = [\![\sigma_d(Q)]\!]$. POSSIAMO A QUESTO PUNTO APPLICARE NUOVAMENTE L'IPOTESI INDUTTIVA, QUESTA VOLTA A $\sigma_e(T)$, OTTENENDO R SEMPLICE CON $[\![R]\!] = [\![\sigma_e(T)]\!]$. ORA

$$[\![R]\!] = [\![\sigma_e(T)]\!] = [\![\sigma_e(\sigma_d(Q))]\!] = [\![\sigma_{e \wedge d}(Q)]\!]$$

CHE È LA TESI
- SE C È NELLA FORMA $d \vee e$, APPLICHIAMO L'IPOTESI INDUTTIVA A $\sigma_d(Q)$ E $\sigma_e(Q)$ OTTENENDO S E T TALI PER CUI $[\![S]\!] = [\![\sigma_d(Q)]\!]$, $[\![T]\!] = [\![\sigma_e(Q)]\!]$ E S, T SONO SEMPLICI. A QUESTO PUNTO

$$[\![S \cup T]\!] = [\![\sigma_d(Q)]\!] \cup [\![\sigma_e(Q)]\!] = [\![\sigma_{d \vee e}(Q)]\!]$$

CHE È LA TESI. $\boxtimes$

- A QUESTO PUNTO SIAMO IN GRADO DI DIMOSTRARE IL LEMMA VERO E PROPRIO, PROCEDENDO PER INDUZIONE SU Q

- TUTTI I CASI DIVERSI DALLA SELEZIONE SONO TRIVIALI, AD ESEMPIO, SE $Q = S \times T$ ALLORA OTTERREMMO PER IPOTESI INDUTTIVA DA S E T DELLE QUERY SEMPLICI W E Z, MENTRE LA QUERY R CHE CERCHIAMO SAREBBE NIENT'ALTRO CHE $W \times Z$, PERCHE

$$[\![R]\!] = [\![W \times Z]\!] = [\![W]\!] \times [\![Z]\!] = [\![S]\!] \times [\![T]\!] = [\![Q]\!]$$

E R SAREBBE OVVIAMENTE SEMPLICE.

- CONSIDERIAMO INVECE IL CASO $Q = \sigma_c(S)$. APPLICHIAMO L'IPOTESI INDUTTIVA AD S, OTTENENDO T SEMPLICE EQUIVALENTE AD S. A QUESTO PUNTO POSSIAMO APPLICARE IL SOTTO-LEMMA A $\sigma_c(T)$ OTTENDO PROPRIO R SEMPLICE EQUIVALENTE AD ESSA. INFATTI

$$[\![R]\!] = [\![\sigma_c(T)]\!] = [\![\sigma_c(S)]\!] = [\![Q]\!].$$

$\boxtimes$

· POSSIAMO FINALMENTE TORNARE ALLA DIMOSTRAZIONE DI $AR \subseteq \mathcal{R}$ E PARTIRE CON L'INDUZIONE, CHE IN QUESTO CASO SARÀ SULLA STRUTTURA DELLA QUERY Q CHE VOGLIAMO TRADURRE IN CALCOLO RELAZIONALE SICURO:

FAREMO IN MODO CHE $FV(F_Q) = \{x_1, \ldots, x_m\}$ DOVE $m = ar(Q)$

- SE Q È UNA RELAZIONE R_i IN $\{R_1, \ldots, R_k\}$ ALLORA

$$F_Q = R_i(x_1, \ldots, x_m)$$

SI VEDE FACILMENTE CHE $[\![F_Q]\!] = [\![Q]\!]$. INOLTRE, F_Q È SICURA, PERCHÉ TUTTE LE VARIABILI SONO LIMITATE, GRAZIE ALLA PRIMA DELLE TRE CLAUSOLE.

- SE $Q = P \cup R$, ALLORA

$$F_Q = F_P \vee F_R$$

INFATTI, $[\![F_Q]\!] = [\![F_P]\!] \cup [\![F_R]\!] = [\![P]\!] \cup [\![R]\!] = [\![Q]\!]$. DOBBIAMO VERIFICARE, PERÒ, CHE F_Q SIA SICURA. E QUESTO RICHIEDE CHE $FV(F_P) = FV(F_R)$. INFATTI $FV(F_P) = \{x_1, \ldots, x_{ar(P)}\}$ E PARIMENTI PER R. MA SICCOME VALGONO I VINCOLI DI INTEGRITÀ, $ar(P) = ar(R)$.

- SE $Q = P - R$, ALLORA

$$F_Q = F_P \wedge \neg F_R$$

OVVIAMENTE $[\![F_Q]\!] = [\![Q]\!]$. F_Q È SICURA PERCHÉ:

- TUTTE LE VARIABILI IN $FV(F_R)$ SONO LIMITATE GRAZIE A F_P
- F_P NON È A SUA VOLTA LA NEGAZIONE DI UNA FORMULA.

- SE $Q = P \times R$, ALLORA

$$F_Q = F_P \wedge F'_R$$

DOVE F'_R È OTTENUTA DALLA FORMULA F_R CHE COSTRUIAMO GRAZIE ALL'IPOTESI INDUTTIVA DOVE PERÒ LE VARIABILI LIBERE $f_1, \dots, f_{ar(R)}$ VENGONO RINOMINATE IN $f_{ar(P)+1}, \dots, f_{ar(P)+ar(R)}$

- SE $Q = \pi_{i_1, \dots, i_n}(P)$, APPLICHIAMO L'IPOTESI INDUTTIVA A P, OTTENENDO UNA FORMULA F_P. SIANO $j_1, \dots, j_m$ GLI INDICI IN $\{1, \dots, ar(P)\}$ CHE NON OCCORRONO NELLA LISTA $i_1, \dots, i_n$. LA FORMULA F_Q SARÀ

$$F_Q = \exists f_{j_1} \dots \exists f_{j_m}. F_P.$$

R CHE CONTIENE DUE CAMPI, IL PRIMO COGNOME E IL SECONDO NOME

$\pi_1(R)$

$\exists f_2 . R(f_1, f_2)$

- SE $Q = \sigma_C(R)$, POSSIAMO GRAZIE AL LEMMA DIMOSTRATO PRECEDENTEMENTE, SUPPORRE CHE Q SIA

SEMPLICE, OSSIA CHE C SIA UN VINCOLO ARITMETICO $i \$ j$ O LA SUA NEGAZIONE $\neg(i \$ j)$.

- NEL PRIMO CASO

$$F_Q = F_R \wedge x_i \$ x_j$$

- NEL SECONDO CASO

$$F_Q = F_R \wedge \neg(x_i \$ x_j)$$

IN ENTRAMBI I CASI $[\![F_Q]\!] = [\![Q]\!]$ PERCHÉ (AD ESEMPIO NEL PRIMO CASO)

$$\left[[\![F_Q]\!](R_1, \ldots, R_k)\right] \ni (v_1, \ldots, v_m) \Leftrightarrow$$

$$\left[[\![F_R]\!](R_1, \ldots, R_k)\right] \ni (v_1, \ldots, v_m) \wedge v_i \$ v_j$$

LE VARIABILI LIBERE IL F_Q SONO LIBERE ANCHE IN F_R E IN QUEST'ULTIMA LIMITATE PER IPOTESI INDUTTIVA.

$\boxtimes$

$\boxed{\mathcal{CR} \subseteq \mathcal{AR}}$

- LA DIMOSTRAZIONE DELL'INCLUSIONE $\mathcal{CR} \subseteq \mathcal{AR}$ PASSA ATTRAVERSO UN LINGUAGGIO INTERMEDIO, CHE OCCORRE DESCRIVERE CON UN MINIMO DI DETTAGLIO, E CHE RISULTA INTERESSANTE INDIPENDENTEMENTE DA QUESTA PROVA.

DATALOG

- È UN FRAMMENTO DEL LINGUAGGIO DI PROGRAMMAZIONE PROLOG, OVVERO IL PIÙ DIFFUSO LINGUAGGIO PER LA PROGRAMMAZIONE LOGICA.
- UN PROGRAMMA DATALOG CONSISTE IN UN INSIEME FINITO DI REGOLE $M_1, \dots, M_n$, CIASCUNA DELLE QUALI ABBIA LA FORMA

$$H :- B_1 \,\&\, B_2 \,\&\, \dots \,\&\, B_q$$

DOVE H È LA <u>TESTA</u> DELLA REGOLA MENTRE $B_1 \& \dots \& B_q$ È IL <u>CORPO</u> DELLA REGOLA.

- ORA:
  - LA TESTA DI OGNI REGOLA HA UNA FORMA PARTICOLARE, OSSIA

$$P(A_1, \dots, A_n)$$

DOVE P È SIMBOLO RELAZIONALE E $A_1, \dots, A_n$ SONO VARIABILI OPPURE COSTANTI; AD ESEMPIO

$$P_1(x_1, x_2, 20)$$
$$P_2(\text{"Rossi"}, x_2, x_3)$$

  - CIASCUNO DEI B_i NEL CORPO PUÒ INVECE ESSERE:
    - UNA FORMULA ATOMICA OPPURE LA SUA NEGAZIONE, OPPURE
    - UN PREDICATO $A = B$ O $A \leq B$ OPPURE LA SUA NEGAZIONE,

- UNA RELAZIONE R_i OPPURE LA SUA NEGAZIONE; AD ESEMPIO

AD ESEMPIO

$$R_1(x_2, x_2)$$
$$\neg R_3(x_4, x_5)$$
$$x_1 \leq 20$$
$$x_2 = \text{"Rossi"}$$
$$P_1(20, x_5)$$
$$\neg P_7(x_5, x_7, x_8)$$

- ESEMPI DI REGOLE POSSONO QUINDI ESSERE

$$P(x,y) :- R_1(x,z) \,\&\, R_2(z,y)$$
$$S(x,20) :- R_4(x,10) \,\&\, \neg(x \leq 7)$$

$\boxed{CR \leq AR}$

$$\begin{tikzcd}
\text{CALCOLO RELAZIONE} \arrow[rr, Rightarrow] \arrow[dr, Rightarrow] & & \text{ALGEBRA RELAZIONALE} \\
& \text{DATALOG} \arrow[ur, Rightarrow] &
\end{tikzcd}$$

ESEMPIO

ANNI(A) :- SOCI(IDV, C, N, A, S) &
PARTITE(IDV, IDP, PV, PP) & ←
¬(PV ≤ PP)

ANNI(A) :- SOCI(IDV, C, N, A, S) &
PARTIRE(IDP, IDV, PP, PV) &
¬(PV ≤ PP)

SEMANTICA

- AD OGNI PROGRAMMA DATALOG È POSSIBILE DARE UNA SEMANTICA PASSANDO ATTRAVERSO LA LOGICA PREDICA
- PIÙ IN DETTAGLIO, DATO UN PROGRAMMA DATALOG CHE CONSISTE IN n REGOLE $\mu_1, \dots, \mu_n$, LA RELATIVA FORMULA SARÀ

$$FORM(\mu_1) \wedge FORM(\mu_2) \wedge \dots \wedge FORM(\mu_n)$$

DOVE OGNI $FORM(\mu_i)$ È DEFINITA NEL MODO SEGUENTE.

- SUPPONIAMO CHE μ ABBIA LA FORMA

$$H :- B_1 \& B_2 \& \dots \& B_q$$

DOVE $FV(H) \cup \bigcup_{i=1}^{q} FV(B_i) = \{X_1, \dots, X_\ell\}$. ALLORA

$$FORM(\mu) \overset{DEF}{=} \forall X_1 \dots \forall X_\ell \left[(B_1 \wedge B_2 \wedge \dots \wedge B_q) \rightarrow H \right]$$

- VOGLIAMO CAPIRE SE UN TALE PROGRAMMA DATALOG CALCOLI IN QUALCHE MODO UNA QUERY.
- SIANO $R_1, \dots, R_k$ I SIMBOLI PREDICATIVI CORRISPONDENTI ALLE RELAZIONI DELLA NOSTRA BASE DI DATI E SIANO $P_1 \dots P_s$ I SIMBOLI RELAZIONALI AUSILIARI (E.G. "ANNI" NELL'ESEMPIO). HA SENSO CHIEDERCI SE DATE $\mathcal{R}_1 \dots \mathcal{R}_k$ ESISTANO <u>UNICHE</u> $\mathcal{P}_1, \dots \mathcal{P}_s$ TALI CHE (*)

$$(D, \{\mathcal{R}_1, \dots, \mathcal{R}_k, \mathcal{P}_1, \dots, \mathcal{P}_s\}) \models \mathrm{FORM}(\mathcal{M}_1) \wedge \dots \wedge \mathrm{FORM}(\mathcal{M}_n)$$

- LA RISPOSTA È NEGATIVA! ANCHE QUANDO CI INTERESSASSIMO ALLA PIÙ PICCOLA TUPLA DI RELAZIONI $\mathcal{P}_1, \dots, \mathcal{P}_s$ CHE SODDISFI (*).
- PER ESEMPIO, CONSIDERIAMO UNA BASE DI DATI CHE CONSISTE IN UN'UNICA RELAZIONE UNARIA $\mathcal{R} = \{(1)\}$, E CONSIDERIAMO IL SEGUENTE PROGRAMMA DATALOG

$$P_1(X) :- R(X)\ \&\ \neg P_2(X)$$
$$P_2(X) :- R(X)\ \&\ \neg P_1(X)$$

ESISTONO DUE "SOLUZIONI" ENTRAMBE MINIMALI OVVERO

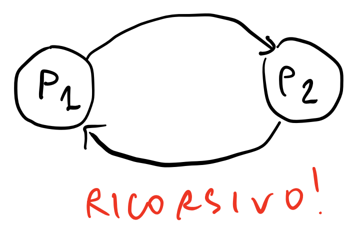
$$\mathcal{P}_1 = \{(1)\} \qquad \mathcal{P}_1 = \emptyset$$
$$\mathcal{P}_2 = \emptyset \qquad \mathcal{P}_2 = \{(1)\}$$

- OCCORRE, QUINDI, TOGLIERE DI MEZZO LA POSSIBILITÀ DI DEFINIRE QUERY RICORSIVE.
- ANCHE IN CASO DI PROGRAMMI DATALOG NON RICORSIVI, DOBBIAMO EVITARE FENOMENI COME QUELLI CHE SI VERIFICANO IN PRESENZA DI REGOLE QUALI

ANNI(A) :- A ≥ 0

ABBIAMO IN ALTRE PAROLE BISOGNO DI UNA NOZIONE DI SICUREZZA DEL TUTTO SIMILE A QUELLA CHE ABBIAMO VISTO PER IL CALCOLO RELAZIONALE.

- RIASSUMENDO E CERCANDO DI ESSERE PIÙ FORMALI, CI INTERESSANO PROGRAMMI DATALOG CHE SIANO
  - ☒ NON RICORSIVI
    - DEFINIAMO GRAFO DELLE DIPENDENZE DI UN PROGRAMMA DATALOG IL GRAFO I CUI NODI SONO I SIMBOLI RELAZIONALI AUSILIARI E ESISTE UN ARCO TRA P_i E P_j OGNIQUALVOLTA UNA REGOLA M_h HA IN TESTA P_i E NEL CORPO P_j
    - CI INTERESSANO I PROGRAMMI DATALOG IL CUI GRAFO DELLE DIPENDENZE SIA ACICLICO, CHE CHIAMIAMO QUINDI NON RICORSIVI
    - ESEMPI



  - ☒ SICURI
    - OCCORRE GARANTIRE CHE TUTTE LE REGOLE M_i DEL PROGRAMMA $\{M_1, \dots, M_n\}$ SIANO SICURE IN UN SENSO DEL TUTTO SIMILE A QUELLO CHE ABBIAMO GIÀ VISTO NEL CALCOLO RELAZIONALE
    - IN PARTICOLARE LA REGOLA

$$M_i \equiv H :- B_1 \& B_2 \& \dots \& B_q$$

È SICURA SE OGNI VARIABILE X IN $FV(H) \cup \bigcup_{i=1}^{q} FV(B_i)$ È DIMOSTRABIL-

MENTE LIMITATA, OSSIA

- ESISTE UN B_i NELLA FORMA $B_i = P(A_1 \dots A_t)$ OPPURE $B_i = R(A_1 \dots A_t)$ E $A_i = X$
- ESISTE UN B_i CON $B_i = (X = a)$ OPPURE $B_i = (a = X)$
- ESISTE UN B_i CON $B_i = (X = Y)$ OPPURE $B_i = (Y = X)$ DOVE Y È LIMITATA ESSA STESSA.

- AD ESEMPIO, LE REGOLE

$$\left.\begin{array}{l} P_1(X,Y) :- X = Y \\ P_2(X) :- X = X \end{array}\right\} \text{NON LIMITATE}$$

$$\left.\begin{array}{l} P_3(X,Y) :- X = Y \ \& \ Y = c \\ P_4(X,Y) :- Y = X \ \& \ R(X) \end{array}\right\} \text{LIMITATE}$$

- DICIAMO CHE UN PROGRAMMA DATALOG HA SEMANTICA BEN DEFINITA SSE PER OGNI $R_1, \dots R_k$ ESISTONO $\mathcal{P}_1, \dots \mathcal{P}_s$ FINITE TALI CHE

$$(D, \{R_1, \dots R_k, \mathcal{P}_1, \dots, \mathcal{P}_s\}) \models \bigwedge_{i=1}^{n} \text{FORM}(M_i) \qquad (**)$$

E OGNI ALTRA TUPLA $Q_1, \dots Q_s$ CHE SODDISFI $(**)$ È TALE PER CUI $Q_1 \supseteq \mathcal{P}_1, \dots Q_s \supseteq \mathcal{P}_s$

- LEMMA

OGNI PROGRAMMA DATALOG NON RICORSIVO E SICURO $\mathcal{D}$ HA SEMANTICA BEN DEFINITA, CHE INTERPRETEREMO COME UNA FUNZIONE

$$[\![\mathcal{D}]\!] : \mathcal{P}_{FIN}(D^{n_1}) \times \dots \times \mathcal{P}_{FIN}(D^{n_k}) \to \mathcal{P}_{FIN}(D^{p})$$

NEL FAR COSÌ, OVVIAMENTE, STIAMO SUPPONENDO CHE TRA I SIMBOLI AUSILIARI DI $\mathcal{D}$ CE NE SIA UNO "PRINCIPALE", CHIAMIAMOLO MAIN.

- A CHE PUNTO SIAMO? ABBIAMO INDIVIDUATO L'IDENTITÀ DEL FORMALISMO INTERMEDIO TRA CALCOLO RELAZIONALE SICURO E ALGEBRA RELAZIONALE;

$$\text{CALCOLO RELAZIONALE SICURO} \xRightarrow{C2D} \text{DATALOG NON RICORSIVO E SICURO} \xRightarrow{D2A} \text{ALGEBRA RELAZIONALE}$$

$$[\![C2D(F)]\!] = [\![F]\!] \qquad [\![D2A(Q)]\!] = [\![Q]\!]$$

SE RIUSCIAMO A COSTRUIRE LE DUE TRASFORMAZIONI DI CUI SOPRA, ABBIAMO IMPLICITAMENTE DIMOSTRATO CHE $CRS \subseteq AR$, PERCHÉ ABBIAMO UN MODO PER MAPPARE OGNI FORMULA DEL CALCOLO RELAZIONALE SICURO IN UNA QUERY EQUIVALENTE DELL'ALGEBRA RELAZIONALE

$$F \longmapsto D2A(C2D(F))$$

- <u>C2D, OVVERO DAL CALCOLO RELAZIONALE A DATALOG.</u>
  - LA TRADUZIONE IN DATALOG C2D(F) SARÀ DEFINITA PER RICORSIONE SULLA STRUTTURA DI F.
  - OCCORRERÀ IN ALTRE PAROLE DISTINGUERE ALLA STRUTTURA DI F.
  - L'UNICA DIFFICOLTÀ CHE INCONTREREMO RIGUARDA IL CASO IN CUI $F = G_1 \wedge G_2 \wedge \ldots \wedge G_m$ OCCORRERÀ IN TAL CASO DISTINGUERE DUE POSSIBILI EVENTUALITÀ:
    1. TUTTI I G_i NON SONO ULTERIORMENTE DECOMPONIBILI (OSSIA SONO SIMBOLI RELAZIONALI, ARITMETICI O LORO NEGA=

ZIONI).

2. ESISTE UN G_i CHE SIA ULTERIORMENTE DECOMPONIBILE.

$\boxed{C2D}$

- VOGLIAMO COSTRUIRE, DATA UNA QUALUNQUE FORMULA F DEL CALCOLO RELAZIONALE SICURO, UN PROGRAMMA DATALOG $\mathcal{P}_F$ CORRISPONDENTE AD F (I.E. $[\![F]\!] = [\![\mathcal{P}_F]\!]$). PER FAR CIÒ FAREMO IN MODO CHE $\mathcal{P}_F$ ABBIA UN SIMBOLO RELAZIONALE "PRINCIPALE", CHE CHIAMEREMO P_F
- PROCEDIAMO DEFINENDO $C2D(\cdot)$ PER RICORSIONE:
  - SE L'ARGOMENTO F È NELLA FORMA $G_1 \wedge G_2 \wedge \ldots \wedge G_m$ (DOVE TUTTE LE G_i NON SONO DECOMPONIBILI) ALLORA POSSIAMO FACILMENTE COSTRUIRE UN SEMPLICE PROGRAMMA DATALOG COME SEGUE

$$P_F(X_1, \ldots, X_n) :- G_1 \,\&\, G_2 \,\&\, \ldots \,\&\, G_m$$

DOVE $\{X_1, \ldots, X_n\} = FV(F)$. NOTIAMO COME $\mathcal{P}_F$ SIA NON-RICORSIVO E SICURO.

  - SE L'ARGOMENTO F, INVECE, È NELLA FORMA $F = \exists X_i . G$, DOVE $FV(G) = \{X_1, \ldots, X_n\}$, ALLORA IL PROGRAMMA $\mathcal{P}_F$ SARÀ OTTENUTO DA $\mathcal{P}_G$ AGGIUNGENDO A QUEST'ULTIMO UN'UNICA REGOLA OSSIA

$$P_F(X_1, \ldots, X_{i-1}, X_{i+1} \ldots X_n) :- P_G(X_1, \ldots, X_n)$$

OSSERVIAMO CHE CONCLUDERE CHE P_F VALE SU DEI VALORI $(A_1 \ldots A_{i-1}, A_{i+1} \ldots A_n)$ RICHIEDE DI DIMOSTRARE CHE ESISTE UN VALORE A_i PER CUI P_G VALE IN $(A_1, \ldots, A_n)$.

  - SE $F = G \vee H$, ALLORA SAPPIAMO CHE $FV(G) = FV(H)$, PERCHÉ F È SICURA. POSSIAMO CALCOLARE C2D A G E AD H, OTTENENDO DUE PROGRAMMI DATALOG $\mathcal{P}_G$ E $\mathcal{P}_H$. A QUESTO PUNTO $\mathcal{P}_F$ CONTERRÀ TUTTE LE REGOLE DI $\mathcal{P}_G$ E $\mathcal{P}_H$ OLTRE ALLE SEGUENTI:

$$P_F(x_1,\dots,x_n) :- P_G(x_1,\dots,x_n)$$
$$P_F(x_1,\dots,x_n) :- P_H(x_1,\dots,x_n)$$

DOVE $\{x_1,\dots,x_n\} = FV(G) = FV(H)$

- C'È UN ULTIMO CASO DA CONSIDERARE, OSSIA QUELLO IN CUI F È UNA CONGIUNZIONE DI FORMULE $G_1 \wedge \dots \wedge G_m$ DI CUI ALMENO UNA SIA ULTERIORMENTE DECOMPONIBILE. DEFINIREMO QUINDI $\mathcal{D}_F$ COME IL PROGRAMMA DATALOG CHE INCLUDE LA REGOLA

$$P_F(x_1,\dots x_n) :- S_1 \,\&\, S_2 \,\&\, \dots \,\&\, S_m$$

E TUTTE LE REGOLE DI $\mathcal{D}_{G_1} \dots \mathcal{D}_{G_m}$, DOVE:

- $\{x_1,\dots,x_n\} = FV(F)$
- SE G_i NON È ULTERIORMENTE DECOMPONIBILE ALLORA $S_i = G_i$ E $\mathcal{D}_{G_i} = \emptyset$
- SE G_i È ULTERIORMENTE DECOMPONIBILE, POSSIAMO C2D AD ESSA, OTTENENDO UN PROGRAMMA $\mathcal{D}_{G_i}$ E $S_i = P_{G_i}(x_1,\dots,x_\ell)$ DOVE $\{x_1,\dots,x_\ell\} = FV(G_i)$.

ANCHE QUI, È EVIDENTE CHE IL PROGRAMMA DATALOG OTTENUTO NON È RICORSIVO ED È SICURO.

- PRIMA DI PASSARE A D2A, VEDIAMO C2D ALL'OPERA IN UN ESEMPIO, OVVERO LA FORMULA F RELATIVA AL CLUB DI TENNIS:

$$F \equiv \Big[\exists p.\exists s.\exists c.\exists n.\exists o.\exists pp.\exists ps.\, R_1(p,c,n,f,o) \wedge R_2(p,s,pp,ps) \wedge (pp > ps)\Big]$$
$$\vee$$
$$\Big[\exists p.\exists s.\exists c.\exists n.\exists o.\exists pp.\exists ps.\, R_1(s,c,n,f,o) \wedge R_2(p,s,pp,ps) \wedge (ps > pp)\Big]$$

$P_F(f) :- P_{FIRST}(f)$ $\quad$ $P_F(f) :- P_{SECOND}(f)$

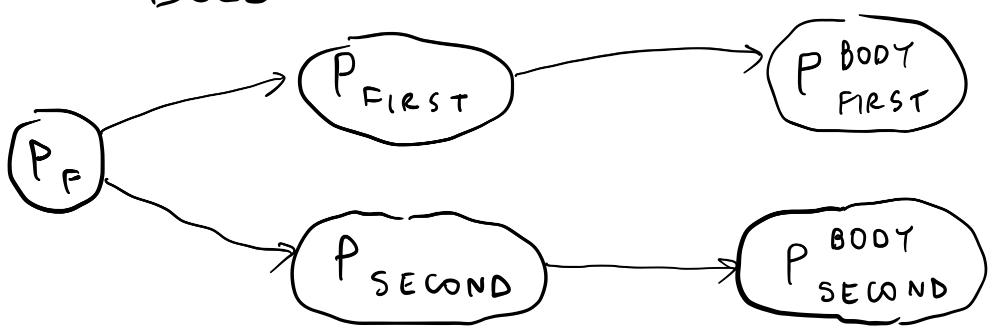
$P_{FIRST}(f) :- P^{BODY}_{FIRST}(p, s, c, n, o, pp, ps, f)$

$P_{SECOND}(f) :- P^{BODY}_{SECOND}(p, s, c, n, o, pp, ps, f)$

$$P^{BODY}_{FIRST}(p, s, c, n, o, pp, ps, f) :- R_1(p, c, n, f, o)\ \&\ R_2(p, s, pp, ps)\ \&\ (pp > ps)$$

$$P^{BODY}_{SECOND}(p, s, c, n, o, pp, ps, f) :- R_1(s, c, n, f, o)\ \&\ R_2(p, s, pp, ps)\ \&\ (ps > pp)$$

GRAFO DELLE DIPENDENZE



D2A

- TRADURRE UN PROGRAMMA DATALOG (NON RICORSIVO E SICURO) IN UNA QUERY DELL'ALGEBRA RELAZIONALE <u>NON È BANALE</u>, VISTA LA DISTANZA CONCETTUALE TRA I DUE FORMALISMI
- SI PROCEDERÀ ATTRAVERSO CINQUE FASI SUCCESSIVE

  1. RETTIFICAZIONE DELLE REGOLE
  2. CALCOLO DELL'ESPRESSIONE DOM
  3. CALCOLO DELL'ORDINE TOPOLOGICO DEL GRAFO DELLE DIPENDENZE
  4. CALCOLO DI UN'ESPRESSIONE DELL'AL-

GEBRA RELAZIONALE PER CIASCUNA REGOLA DEL PROGRAMMA DATALOG.

$\boxed{5}$ CALCOLO DI UN'ESPRESSIONE DELL'ALGEBRA RELAZIONALE PER CIASCUNA RELAZIONE AUSILIARIA DEL PROGRAMMA DATALOG.

$\boxed{1}$ RETTIFICAZIONE DELLE REGOLE

- DA QUI IN POI, SUPPONIAMO CHE IL PROGRAMMA DATALOG CON CUI LAVORIAMO CONSTI DI n REGOLE $\mu_1, \mu_2, \dots, \mu_n$.
- VOGLIAMO TUTTE LE REGOLE TRA $\mu_1, \dots, \mu_n$ CHE IN TESTA HANNO LO STESSO SIMBOLO RELAZIONALE AUSILIARIO ABBIANO <u>LA STESSA TESTA</u>.
- A CIASCUN SIMBOLO R, ATTRIBUIREMO UNA LISTA DI VARIABILI "CANONICHE", CHIAMIAMOLA $X_1^R, \dots X_m^R$ (DOVE m È L'ARIETÀ DI R) CHE SIANO "FRESCHE".
- OGNI REGOLA

$$\mu_i \equiv R(A_1, \dots, A_m) :- B_1 \,\&\, \dots \,\&\, B_q$$

DIVENTERÀ QUINDI LA REGOLA

$$\mu_i' \equiv R(X_1^R, \dots, X_m^R) :- B_1 \,\&\, \dots \,\&\, B_q \,\&\, X_1^R = A_1 \,\&\, \dots \,\&\, X_m^R = A_m.$$

. AD ESEMPIO

$$R_1(X, Y, c, X) :- R_2(X, Y)$$
$$R_1(X, X, Z, d) :- X = d \,\&\, Z = e$$
$$R_2(X, X) :- X = f$$

DIVENTA

$$R_1(X_1^{R_1}, X_2^{R_1}, X_3^{R_1}, X_4^{R_1}) :- R_2(X, Y) \,\&$$

$X_1^{R_2} = X$ &

$X_2^{R_1} = Y$ &

$X_3^{R_1} = c$ &

$X_4^{R_2} = X$

⋮

$\boxed{2}$ CALCOLO DELL'ESPRESSIONE DOM

- AVREMO BISOGNO NEL SEGUITO, DI UN'ESPRESSIONE DELL'ALGEBRA RELAZIONALE CHE VALUTI AD UNA RELAZIONE, DI ARIETÀ 1, CONTENENENTE TUTTI E SOLI I VALORI DEL DOMINIO D CHE OCCORRONO:
  - NELLA BASE DI DATI
  - NEL PROGRAMMA DATALOG $\{\mu_1, \dots, \mu_n\}$
- SUPPONIAMO DI VOLER RACCOGLIERE IN UNA RELAZIONE TUTTI I VALORI DEL DOMINIO CHE OCCORRONO IN R_i (DI ARIETÀ m. LO POSSIAMO FARE AGEVOLMENTE TRAMITE

$$Q_{R_i} \equiv \Pi_1(R_i) \cup \Pi_2(R_i) \cup \dots \cup \Pi_m(R_i)$$

- PER QUEL CHE RIGUARDA I VALORI CHE OCCORRONO IN $\{\mu_1, \dots, \mu_n\}$ OCCORRERÀ COSTRUIRE UNA RELAZIONE "COSTANTE" OSSIA UN'ESPRESSIONE NELLA FORMA

E' UNA QUERY DELL'ALGEBRA RELAZIONALE ESTESA.

$$Q_{\{\mu_1, \dots \mu_n\}} \equiv \{(d_1)\} \cup \{(d_2)\} \cup \dots \cup \{(d_k)\}$$

DOVE $d_1, \dots, d_k$ SONO LE COSTANTI CHE OCCORRONO IN $\{\mu_1, \dots \mu_n\}$

- L'ESPRESSIONE CHE STIAMO CERCANDO SARÀ

$$DOM = Q_{R_1} \cup Q_{R_2} \cup \dots \cup Q_{R_m} \cup Q_{\{M_1, \dots, M_n\}}$$

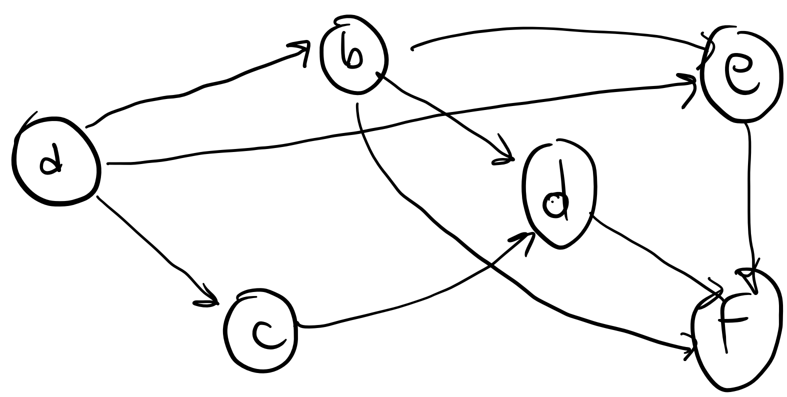
$\boxed{3}$ CALCOLO DELL'ORDINE TOPOLOGICO DEL GRAFO DELLE DIPENDENZE

- I NODI DI OGNI GRAFO ACICLICO POSSONO ESSERE ORDINATI LINEARMENTE IN MODO TALE CHE SE

$$n_i < n_j$$

ALLORA NON È POSSIBILE COSTRUIRE UN CAMMINO DA n_j A n_i

- PER ESEMPIO UN GRAFO COME IL SEGUENTE



AMMETTE, TRA GLI ALTRI, I SEGUENTI ORDINAMENTI TOPOLOGICI

$$a > c > b > d > e > f$$

a > b, c > d > e > f

- SI TRATTA SOLTANTO DI OTTENERE UNO TRA I MOLTI ORDINAMENTI TOPOLOGICI, LA QUAL COSA PUÒ ESSERE FATTA TRAMITE UNA SEMPLICE VISITA DEL GRAFO.

$\boxed{4}$ CALCOLO DI UN'ESPRESSIONE DELL'ALGEBRA RELAZIONALE EQUIVALENTE AL CORPO DI OGNUNA DELLE REGOLE

- COSTRUIAMO UNA TALE ESPRESSIONE A PARTIRE DA UNA REGOLA DI ESEMPIO, OSSIA

$$\text{ANNI}(A) :- \text{SOCI}(IDV, C, N, A, S) \,\&\, \text{PARTITE}(IDV, IDP, PV, PP) \,\&\, \neg(PV \leq PP)$$

L'ESPRESSIONE CHE CI SERVE SARÀ

$$\sigma_{(1=6) \wedge \neg(8 \leq 9)}(\overset{\downarrow}{\text{SOCI}} \times \overset{\downarrow}{\text{PARTITE}})$$

- C'È UN MODO PER GENERALIZZARE TUTTO QUESTO, MA LASCIAMO LA QUESTIONE IN SOSPESO.

$\boxed{5}$ CALCOLO DI UN'ESPRESSIONE DELL'ALGEBRA RELAZIONALE PER

CIASCUNA RELAZIONE AUSILIARIA DEL PROGRAMMA DATALOG

- L'ESPRESSIONE DI CUI ABBIAMO BISOGNO SI PUÒ COSTRUIRE PER INDUZIONE SULLA POSIZIONE DEL SIMBOLO RELAZIONALE IN OGGETTO NELL'ORDINAMENTO TOPOLOGICO DEL GRAFO DELLE DIPENDENZE
  → TRATTEREMO I SIMBOLI "PIÙ GRANDI" PRIMA DEI SIMBOLI "PIÙ PICCOLI".

- CASO BASE:
  - IL CASO BASE RIGUARDA I SIMBOLI RELAZIONALI CHE NON FANNO RIFERIMENTO AD ALTRI SIMBOLI RELAZIONALI.
  - GRAZIE AI PUNTI $\boxed{1}$ E $\boxed{4}$ SAPPIAMO CHE LE RELATIVE REGOLE SARANNO NELLA FORMA

$$R(x_1, \dots, x_n) :- \mathcal{B}_1$$
$$\vdots$$
$$R(x_1, \dots, x_n) :- \mathcal{B}_p$$

  E SAPPIAMO ANCHE CHE A $\mathcal{B}_1, \dots, \mathcal{B}_p$ CORRISPONDONO DELLE QUERY $Q_1, \dots, Q_p$ DELL'ALGEBRA RELAZIONALE.
  - LA QUERY CHE CI SERVE SARÀ

$$\Pi_{i_1^1,\dots,i_n^1}(Q_1) \cup$$
$$\Pi_{i_1^2,\dots,i_n^2}(Q_2) \cup$$
$$\vdots$$
$$\Pi_{i_2^p,\dots,i_n^p}(Q_p)$$

DOVE L'INDICE i_j^s È QUELLO DELLA VARIABILE j NELLA QUERY s (IL QUALE ESISTE SEMPRE)

- CASO INDUTTIVO:

  SI PROCEDE ESATTAMENTE COME NEL CASO PRECEDENTE, STANDO ATTENTI A GESTIRE LA TRADUZIONE DI B_i IN Q_i NEL MODO OPPORTUNO, OSSIA FACENDO USO DELL'IPOTESI INDUTTIVA.

D2A

- CI MANCA L'ULTIMO "PEZZO" DELLA TRADUZIONE, OSSIA QUELLO CHE PERMETTE DI RISCRIVERE IL CORPO DI OGNI REGOLA DI DATALOG SICURO IN UNA QUERY DELL'ALGEBRA RELAZIONALE.
- CONSIDERIAMO UN ESEMPIO:

$$P(X_1, X_2) :- \neg R(X_1, X_1, X_4) \;\&\; Q(X_3) \;\&\; X_2 = X_3 \;\&\; \ldots X_4 = a \;\&\; X_1 = b$$

ENTRAMBE QUESTE VARIABILI A DESTRA DEVONO OCCORRERE

PARTE RELAZIONALE

PARTE ARITMETICA

LA PARTE RELAZIONALE CONTRIBUIRÀ ALLA FORMAZIONE DELLA QUERY TRAMITE UN PRODOTTO CARTESIANO, MENTRE LA PARTE ARITMETICA SARÀ RIFLESSA NELLA QUERY ATTRAVERSO UN'OPERATORE DI SELEZIONE

$$\Pi_{1,5}\left\{\sigma_C\left[\underbrace{\left(DOM^3 - R\right)}_{3 \text{ POSIZIONI}} \times \underbrace{Q}_{1 \text{ POSIZIONE}} \times \underbrace{DOM}_{1 \text{ POSIZIONE}} \times \{(a)\} \times \{(b)\}\right]\right\}$$

X_1	X_1	X_4	X_3	X_2		
1	2	3	4	5	6	7

QUESTA CONDIZIONE PERMETTE DI SELEZIONARE LE TUPLE "INTERESSANTI"

$$C = (1=2) \wedge \underbrace{(4=5)}_{X_2 = X_3} \wedge (3=6) \wedge (1=7)$$

CERCHIAMO DI SISTEMATIZZARE IL TUTTO, DANDO UNO SCHEMA DI TRADUZIONE IL PIÙ POSSIBILE GENERALE:

$$P(X_1,\dots,X_p) :- B_1 \,\&\, B_2 \,\dots\, \&\, B_n$$

PARTE RELAZIONALE

$B_{i_1},\dots,B_{i_l}$

(COSSIA TUTTI I FATTI NELLA FORMA $R(A_1,\dots A_m)$ O $\neg R(A_1,\dots,A_m)$)

- LE VARIABILI IN GIOCO SONO GLI ELEMENTI DELL'INSIEME

$$\{\underbrace{X_1,\dots,X_p}_{\text{VARIABILI IN TESTA}},\ \underbrace{X_{p+1},\dots X_{p+c}}_{\text{VARIABILI CHE STANNO SOLO NEL CORPO}}\}$$

- UN SOTTOINSIEME VARPRED DI $\{X_1,\dots,X_{p+c}\}$ CONTIENE TUTTE E SOLE LE VARIABILI CHE OCCORRONO NELLA PARTE PREDICATIVA.
- LA PARTE PREDICATIVA CORRISPONDE AD UNA QUERY DELL'ALGEBRA RELAZIONALE NELLA FORMA

PARTE ARITMETICA

$B_{j_1},\dots,B_{j_k}$

- È POSSIBILE ORA, SULLA BASE DEI POS(i) E POS(a), DEFINIRE UNA CONGIUNZIONE LOGICA

$$C_1 \wedge C_2 \wedge \dots \wedge C_k$$

DOVE C_i È OTTENUTO DA B_{j_i} SOSTITUENDO X_i CON POS(i) E a CON POS(a)

$$Q = Q_1 \times \dots \times Q_l \times DOM^{c+p-|VARPRED|} \times \{(a_1)\} \times \dots \times \{(a_s)\}$$

DOVE
- $a_1,\dots,a_s$ SONO LE COSTANTI CHE OCCORRONO NELLA REGOLA.
- Q_h VIENE COSTRUITO NEL MODO OVVIO A PARTIRE DA B_{i_h}, OSSIA
  - → SE B_{i_h} È $R(A_1,\dots A_r)$ ALLORA $Q_h = R$
  - → SE B_{i_h} È $\neg R(A_1 \dots A_r)$ ALLORA $Q_h = DOM^r - R$

- C'È BISOGNO DI METTERE IN RELAZIONE VARIABILI E INDICI DI COLONNA. IN QUESTO SENSO DEFINIAMO:

- → INDEX(i) È L'INSIEME DEGLI INDICI DELLE COLONNE DI Q CORRISPONDENTI A X_i
- → POS(i) È UNO DEGLI ELEMENTI DI INDEX(i)
- → AND(i) LA SEGUENTE CONGIUNZIONE LOGICA
  $$(POS(i) = q_1) \wedge \ldots \wedge (POS(i) = q_r) \quad \text{DOVE} \quad INDEX(i) = \{POS(i), q_2, \ldots, q_r\}$$
- → DATA UNA QUALUNQUE COSTANTE a CHE OCCORRA NELLA REGOLA, POS(a) È L'INDICE IN Q DELLA RELAZIONE COSTANTE $\{(a)\}$ CORRISPONDENTE.

LA QUERY CHE CERCHIAMO NON SARÀ ALTRO CHE:

$$\pi_{POS(1),\ldots,POS(p)}\left(\sigma_{C_1 \wedge \ldots \wedge C_k \wedge \bigwedge_i AND(i)}(Q)\right)$$

(OSSERVIAMO COME SIA STATA FATTA L'ASSUNZIONE CHE LA PARTE RELAZIONALE NON CONTENGA COSTANTI; TALE ASSUNZIONE NON FA PERDERE GENERALITÀ)