

Soluzione Esercitazione 3

Esercizio 1

Trovare per ogni film i suoi seguiti diretti o indiretti.

$$SEQUEL(X, Y) : -SEQUEL_DIRETTO(X, Y)$$

$$SEQUEL(X, Y) : -SEQUEL_DIRETTO(X, Z), SEQUEL(Z, Y)$$

$$? - SEQUEL(X, Y)$$

La prima regola è la regola base di inizializzazione, contiene solo estensionali nel corpo. La seconda regola è ricorsiva: ha lo stesso predicato nel corpo e in testa!

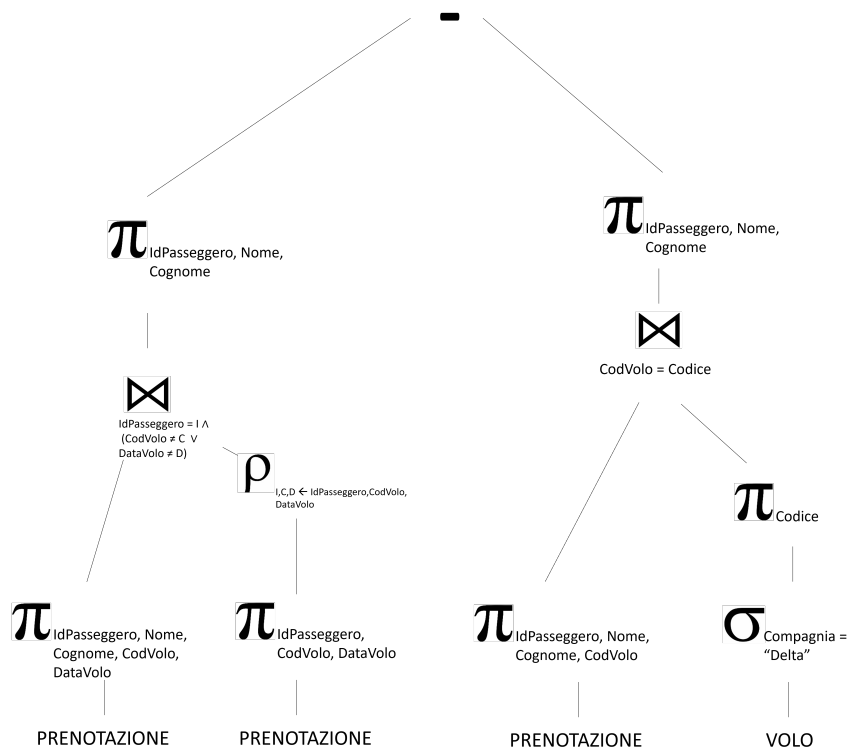
La terminazione si verifica quando non si può derivare altra conoscenza: la regola 2 continua ad attivarsi e ad aggiornare il database finché non vi sono più attivazioni possibili.

Attenzione: non c'è un "ordine" delle regole Datalog, se ne occupa la macchina di selezionare la regola da attivare ad ogni step (con tecniche avanzate che non vediamo).

Esercizio 2

Scrivere la seguente query in AR, CR e Datalog: trovare id, nome e cognome dei passeggeri che hanno prenotato almeno due voli ma nessuno della compagnia "Delta".

Algebra Relazionale:



In Calcolo Relazionale:

$$\{t|\exists t_1 \in PRENOTAZIONE, \exists t_2 \in PRENOTAZIONE, \\ (t[id, nome, cognome] = t_1[IdPasseggero, Nome, Cognome] \wedge \\ t_1[IdPasseggero] = t_2[IdPasseggero] \wedge (t_1[CodVolo] <> t_2[CodVolo] \\ \vee t_1[DataVolo] <> t_2[DataVolo]) \wedge \neg(\exists t_3 \in \\ PRENOTAZIONE, \exists t_4 \in VOLO, (t_4[Compagnia] = "Delta" \wedge t_3[CodVolo] = \\ t_4[CodVolo] \wedge t_3[IdPasseggero] = t_1[IdPasseggero])))\}$$

In Datalog:

$$\begin{aligned} PasseggeroDelta(p) : & \neg Prenotazione(p, v, -, -, -, -), Volo(v, "Delta", -, -, -, -) \\ PasseggeroScelto(p, n, c) : & \neg Prenotazione(p, v1, -, n, c, -, -), Prenotazione(p, v2, -, -, -, -), \\ & \neg PasseggeroDelta(p), v1 <> v2 \\ PasseggeroScelto(p, n, c) : & \neg Prenotazione(p, -, d1, n, c, -, -), Prenotazione(p, -, d2, -, -, -, -), \\ & \neg PasseggeroDelta(p), d1 <> d2 \\ ? - & PasseggeroScelto(x, y, z) \end{aligned}$$

Distributori Automatici

Esercizio 3

Trovare i distributori che non hanno fatto nessuna vendita successiva al loro ultimo rifornimento.

$$\{t|\exists t_1 \in DISTRIBUTORE(t[Codice] = t_1[Codice] \wedge \neg(\exists t_2 \in VENDITA \\ (t_2[CodiceDistributore] = t_1[Codice] \wedge (t_2[Data] > t_1[DataUltimoRif] \vee \\ (t_2[Data] = t_1[DataUltimoRif] \wedge t_2[Ora] > t_1[OraUltimoRif])))))\}$$

In Datalog:

$$\begin{aligned} DISTR_CON_VENDITA(C) : & \neg VENDITA(C, D1, O1, -, -, -), \\ & DISTRIBUTORE(C, -, D2, O2), D1 > D2 \\ DISTR_CON_VENDITA(C) : & \neg VENDITA(C, D1, O1, -, -, -), \\ & DISTRIBUTORE(C, -, D2, O2), D1 = D2, O1 > O2 \\ NO_SUCCESSIVA(C) : & \neg DISTRIBUTORE(C, -, -, -), \neg DISTR_CON_VENDITA(C) \\ ? - & NO_SUCCESSIVA(X) \end{aligned}$$

Esercizio 4

Trovare l'ultima vendita effettuata di ciascun distributore.

$$\{t|\exists t_1 \in VENDITA(t[CD, D, O] = t_1[CodDistributore, Data, Ora] \wedge \neg(\exists t_2 \in VENDITA \\ (t_1[CodDistributore] = t_2[CodDistributore] \wedge (t_2[Data] > t_1[Data] \vee \\ (t_2[Data] = t_1[Data] \wedge t_2[Ora] > t_1[Ora])))))\}$$

In Datalog:

$$\begin{aligned}
&NON_ULTIME(C1, D1, O1) : -VENDITA(C1, D1, O1, -, -, -), \\
&\quad VENDITA(C1, D2, O2, -, -, -), D2 > D1 \\
&NON_ULTIME(C1, D1, O1) : -VENDITA(C1, D1, O1, -, -, -), \\
&\quad VENDITA(C1, D2, O2, -, -, -), D1 = D2, O2 > O1 \\
&ULTIMA(C, D, O) : -VENDITA(C, D, O, -, -, -), \neg NON_ULTIME(C, D, O) \\
&? - ULTIMA(X_1, X_2, X_3)
\end{aligned}$$

Esercizio 5

Trovare le bevande che sono state vendute almeno due volte oppure che hanno un prezzo maggiore di 2 euro.

$$\begin{aligned}
&\{t | (\exists t_1 \in VENDITA, \exists t_2 \in VENDITA, \\
&\quad (t[B] = t_1[CodBevanda] \wedge t_1[CodBevanda] = t_2[CodBevanda] \wedge \\
&\quad (t_1[CodDistributore] <> t_2[CodDistributore] \vee t_1[Data] <> t_2[Data] \vee \\
&\quad t_1[Ora] <> t_2[Ora]))) \vee (\exists t_3 \in BEVANDA, (t[B] = t_3[Codice], t_3[Prezzo] > 2))\}
\end{aligned}$$

In Datalog:

$$\begin{aligned}
&DUEVENDITE_O_COSTOSA(B) : -VENDITA(C1, -, -, B, -, -), \\
&\quad VENDITA(C2, -, -, B, -, -), C1 \neq C2 \\
&DUEVENDITE_O_COSTOSA(B) : -VENDITA(-, D1, -, B, -, -), \\
&\quad VENDITA(-, D2, -, B, -, -), D1 \neq D2 \\
&DUEVENDITE_O_COSTOSA(B) : -VENDITA(-, -, O1, B, -, -), \\
&\quad VENDITA(-, -, O2, B, -, -), O1 \neq O2 \\
&DUEVENDITE_O_COSTOSA(B) : -BEVANDA(C., P, -), P > 2 \\
&? - DUEVENDITE_O_COSTOSA(X)
\end{aligned}$$

Esercizio 6

Trovare la bevanda più costosa.

$$\begin{aligned}
&\{t | \exists t_1 \in BEVANDA, (t[B] = t_1[Codice] \wedge \\
&\quad \neg(\exists t_2 \in BEVANDA, (t_2[Codice] \neq t_1[Codice], t_2[Prezzo] > t_1[Prezzo])))\}
\end{aligned}$$

In Datalog:

$$\begin{aligned}
&BEVANDA_ECONOMICA(B1) : -BEVANDA(B1, -, P1, -), \\
&\quad BEVANDA(B2, -, P2, -), B1 \neq B2, P2 > P1 \\
&BEVANDA_CARA(B) : -BEVANDA(B, -, -, -), \\
&\quad \neg BEVANDA_ECONOMICA(B) \\
&? - BEVANDA_CARA(X)
\end{aligned}$$

Esercizio 7

Trovare i distributori che hanno venduto tutte le bevande.

$$\{t | \exists t_1 \in VENDITA, (t[D] = t_1[CodDistributore] \wedge \neg(\exists t_2 \in BEVANDA, \neg(\exists t_3 \in VENDITA(t_3[CodDistributore] = t_1[CodDistributore], t_3[CodBevanda] = t_2[Bevanda]))))\}$$

In Datalog:

$$\begin{aligned} VENDITE_FATTE(CD, CB) &: \neg VENDITA(CD, -, -, CB, -, -). \\ SENZA_VENDITA(CD) &: \neg VENDITA(CD, -, -, -, -, -), \\ &\quad BEVANDA(B, -, -, -), \neg VENDITE_FATTE(CD, B). \\ TUTTE_VENDUTE(CD) &: \neg VENDITA(CD, -, -, -, -, -), \neg SENZA_VENDITA(CD). \\ ? - TUTTE_VENDUTE(CD) \end{aligned}$$