Premessa

Ovviamente ci sono tante soluzioni ammissibili, quindi questa è solo una di tante. Nell'esame non e' necessario scrivere in un modo molto verboso come qui, visto che lo scopo di questa soluzione è di far capire e non di dimostrare la conoscenza delle materie.

1 Modellazione

La soluzione più semplice è avere sei variabili A, B, C, D, E, F, la verità di ognuno rappresenta che il valore nel circuito sia 1, e la falsità che il valore nel circuito sia 0.

Per costruire la formula, osserviamo la semantica dei componenti nel circuito. Possiamo scrivere tabelle in cui i_1 e i_2 sono l'input (a sinistra):

i_1	i_2	> 1
0	0	0
0	1	0
1	0	0
1	1	1

Quindi la componente > 1 funziona come una congiunzione.

i_1	i_2	> 0
0	0	0
0	1	1
1	0	1
1	1	1

La componente > 1 funziona come una congiunzione.

i_1	i_2	$= 1$
0	0	0
0	1	1
1	0	1
1	1	0

La componente $= 1$ funziona come una disgiunzione esclusiva, ovvero la negazione di equivalenza.

Poi possiamo scrivere che il valore di ogni output (D, E, F) deve essere esattamente uguale (cioè ci vuole l'equivalenza) all'output dell'ultimo componente, che al suo posto può essere espresso tramite una formula con solo variabili dell'input.

$$(D \leftrightarrow ((A \wedge B) \vee E)) \wedge$$
$$(E \leftrightarrow (\neg(A \leftrightarrow B) \vee C)) \wedge$$
$$(F \leftrightarrow (\neg(\neg(A \leftrightarrow B) \leftrightarrow C))$$

2 Formule

Formula (1) è una formula della logica del primo ordine, formula (2) è una QBF, formula (3) è una formula della logica proposizionale. Quindi per stabilire la soddisfacibilità possiamo utilizzare: per (1) la risoluzione del primo ordine o il calcolo dei sequenti, per (2) DLL-QSAT, per (3) la risoluzione proposizionale, DLL o una tabella di verità. Ci sono anche altri metodi, ma non li abbiamo discussi in profondità nel corso.

2.1 Risoluzione del Primo Ordine

La soddisfacibilità della formula (1) può essere determinata tramite la risoluzione del primo ordine.

Prima trasformiamo la formula in Skolem prenex CNF:

$$\begin{aligned}
&\exists A \forall B((\neg p(A, f(B)) \rightarrow \neg q(A, B)) \wedge (q(A, B) \rightarrow \neg p(A, f(B)))) \equiv \\
&\exists A \forall B((\neg\neg p(A, f(B)) \vee \neg q(A, B)) \wedge (\neg q(A, B) \vee \neg p(A, f(B)))) \equiv \\
&\exists A \forall B((p(A, f(B)) \vee \neg q(A, B)) \wedge (\neg q(A, B) \vee \neg p(A, f(B))))=| \\
&\forall B((p(s, f(B)) \vee \neg q(s, B)) \wedge (\neg q(s, B) \vee \neg p(s, f(B))))
\end{aligned}$$

Poi tentiamo di trovare una rifiutazione:

$C_1 = \{p(s, f(B)), \neg q(s, B)\}$
$C_2 = \{\neg p(s, f(B)), \neg q(s, B)\}$
$C_1\sigma_1 = \{p(s, f(C)), \neg q(s, C)\} \qquad \sigma_1 = \{C/B\}$
$C_2\sigma_2 = \{\neg p(s, f(D)), \neg q(s, D)\} \qquad \sigma_2 = \{D/B\}$
$C_3 = \{\neg q(s, C)\}$ risoluzione di $C_1\sigma_1$ e $C_2\sigma_2$ su $p(s, f(C))$ e $\neg p(s, f(D))$ con unificatore $\{C/D\}$

Visto che non è possibile nessun altra risoluzione, possiamo concludere che la formule è soddisfacibile.

2.2 Risoluzione Proposizionale

La soddisfacibilità della formula (3) può essere determinata tramite la risoluzione proposizionale.

Trasformiamo la formula in CNF:

$$\begin{aligned}
&((\neg A \vee \neg B) \wedge (B \vee \neg A) \wedge (\neg A \rightarrow A)) \equiv \\
&((\neg A \vee \neg B) \wedge (B \vee \neg A) \wedge (\neg\neg A \vee A)) \equiv \\
&((\neg A \vee \neg B) \wedge (B \vee \neg A) \wedge (A \vee A)) \equiv \\
&((\neg A \vee \neg B) \wedge (B \vee \neg A) \wedge A)
\end{aligned}$$

Poi tentiamo di trovare una rifiutazione:

$C_1 = \{\neg A, \neg B\}$
$C_2 = \{\neg A, B\}$
$C_3 = \{A\}$
$C_4 = \{\neg A\}$ risoluzione di C_1 e C_2 su B e $\neg B$
$C_5 = \square$ risoluzione di C_3 e C_4 su A e $\neg A$

Quindi la formula è insoddisfacibile.

3 Unificazione

Si inizia con $k = 0$ e $\delta_0 = \emptyset$ ed $E\delta_0 = \{f(X, c(X, X)), f(b(Y), c(b(Y), Y))\}$. $D(E\delta_0) = \{X, b(Y)\}$. C'è la variabile X, e quindi $k = 1$, $\delta_1 = \{b(Y)/X\}$. $E\delta_1 = \{f(bar(Y), c(bar(Y), bar(Y))), foo(bar(Y), c(bar(Y), Y))\}$. $D(E\delta_1) = \{bar(Y), Y\}$. La variabile Y occorre in $bar(Y)$ e quindi $f(X, c(X, X))$ e $f(b(Y), Y)$ non sono unificabili.

4 Logica e Basi di Dati

Modellazione: Una relazione *stazione* con un attributo (un nome "standardizzato" per gestire stazioni omonimi) che serve anche come chiave; una relazione *collegamento* con due attributi che sono nomi di stazioni, questi due sono la chiave della relazione; una relazione *treno* con tre attributi, uno è l'identificatore del treno (chiave), e gli altri due sono la chiave del collegamento su cui viaggia. Quindi:

$$stazione(nome)$$
$$collegamento(staz1, staz2)$$
$$treno(id, staz1, staz2)$$

1. **Algebra relazionale:**

   $$stazione - (\pi_1 collegamento \cup \pi_2 collegamento)$$

   Calcolo relazionale:

   $$\{S \mid stazione(S) \wedge \neg\exists T(collegamento(S, T) \vee collegamento(T, S))\}$$

   Datalog:

   $$\begin{array}{lcl} stazione\_collegata(S) & \leftarrow & collegamento(S, T). \\ stazione\_collegata(S) & \leftarrow & collegamento(T, S). \\ stazione\_noncollegata(S) & \leftarrow & stazione(S), \texttt{not } stazione\_collegata(S). \end{array}$$

2. **Algebra relazionale:**

   $$stazione - (\pi_2 treno \cup \pi_3 treno)$$

   Calcolo relazionale:

   $$\{S \mid stazione(S) \wedge \neg\exists T \exists R(treno(T, S, R) \vee treno(T, R, S))\}$$

   Datalog:

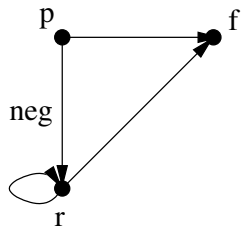
   $$\begin{array}{lcl} stazione\_servita(S) & \leftarrow & treno(T, S, R). \\ stazione\_servita(S) & \leftarrow & treno(T, R, S). \\ stazione\_nonservita(S) & \leftarrow & stazione(S), \texttt{not } stazione\_servita(S). \end{array}$$

3. **Algebra relazionale:** Non possibile.
   Calcolo relazionale: Non possibile.
   Datalog:

$$
\begin{array}{lcl}
\mathit{stazione\_servita}(S) & \leftarrow & \mathit{treno}(T,S,R). \\
\mathit{stazione\_servita}(S) & \leftarrow & \mathit{treno}(T,R,S). \\
\mathit{connessione}(S,R) & \leftarrow & \mathit{treno}(T,S,R). \\
\mathit{connessione}(S,R) & \leftarrow & \mathit{connessione}(S,T), \mathit{connessione}(T,R). \\
\mathit{non\_connesso\_a\_tutte}(S) & \leftarrow & \mathit{stazione}(S), \mathit{stazione\_servita}(T), \\
 & & S \neq T, \texttt{not } \mathit{connessione}(S,T). \\
\mathit{connesso\_a\_tutte}(S) & \leftarrow & \mathit{stazione}(S), \texttt{not } \mathit{non\_connesso\_a\_tutte}(S).
\end{array}
$$

5 Programmi Logici

Segue il grafo delle dipendenze del programma P:



Il programma è stratificabile perché non ci sono cicli che coinvolgono archi negativi.

Qualsiasi stratificazione λ deve soddisfare i sequenti vincoli:

$$\lambda(p) > \lambda(r)$$
$$\lambda(p) \geq \lambda(f)$$
$$\lambda(r) \geq \lambda(f)$$

Quindi una scelta ammissibile sarebbe $\lambda(p) = 1$, $\lambda(r) = \lambda(f) = 0$. Quindi la partizione del programma al rispetto di λ è:

$$P_0 = \{f(a).\ f(b).\ r(X) \leftarrow r(X), f(X).\}$$
$$P_1 = \{p(X) \leftarrow f(X), \texttt{not } r(X).\}$$

Le versioni ground al rispetto della base di Herbrand di P sono:

$$GP_0 = \{f(a).\ f(b).\ r(a) \leftarrow r(a), f(a).\ r(b) \leftarrow r(b), f(b).\}$$
$$GP_1 = \{p(a) \leftarrow f(a), \texttt{not } r(a).\ p(b) \leftarrow f(b), \texttt{not } r(b).\}$$

$T_{GP_0}(\emptyset) = \{f(a), f(b)\}$
$T_{GP_0}(\{f(a), f(b)\}) = \{f(a), f(b)\} = T^{\infty}_{GP_0} =: I_1$
$T_{GP_1}(I_1) = \{f(a), f(b), p(a), p(b)\}$
$T_{GP_1}(\{f(a), f(b), p(a), p(b)\}) = \{f(a), f(b), p(a), p(b)\} = T^{\infty}_{GP_1} =: I_2$

I_2 è il modello perfetto e quindi anche l'unico modello stabile.