

RISOLUZIONE IN LOGICA PROPOSIZIONALE

Giovanna D'Agostino
Dipartimento di Matematica e Informatica, Università di Udine
Versione del 30/11/08.)

1. RISOLUZIONE

Definizione 1.1. Un letterale ℓ è una variabile proposizionale (letterale positivo) oppure la negazione di una variabile proposizionale (letterale negativo).

Se ℓ è un letterale, allora

$$\bar{\ell} = \begin{cases} \neg P, & \text{se } \ell = P \in \mathcal{P} \\ P, & \text{se } \ell = \neg P \text{ con } P \in \mathcal{P} \end{cases}$$

Definizione 1.2. Una clausola è F oppure una disgiunzione di un numero finito di letterali: $C = \ell_1 \vee \dots \vee \ell_n$.

Ad esempio sono clausole le formule seguenti: $P \vee \neg Q$; $\neg P$; $\neg P \vee \neg Q$.

Da un punto di vista sintattico la clausola $P \vee Q$ è distinta dalla clausola $Q \vee P$, ma l'ordine non conta se consideriamo le clausole a meno di equivalenza semantica. Per questa ragione adotteremo la rappresentazione insiemistica di una clausola, indicandola semplicemente con l'insieme di letterali di cui è disgiunzione: ad esempio, rappresenteremo entrambe le clausole $P \vee \neg Q$, $\neg Q \vee P$ con l'insieme $\{P, \neg Q\}$. L'insieme vuoto di letterali corrisponde alla clausola $\perp$ e si indica con $\square$.

Ad esclusione della clausola vuota, che è insoddisfacibile, tutte le altre clausole sono soddisfacibili (basta rendere vero un letterale contenuto nella clausola). É invece possibile che un insieme di clausole sia insoddisfacibile: ad esempio, se S contiene la clausola vuota oppure le due clausole $\{\ell\}$, $\{\bar{\ell}\}$, per un dato letterale ℓ . Come vedremo, questi due esempi rappresentano dei casi *semplici* di insoddisfacibilità. L'insieme vuoto $S = \emptyset$ di clausole è invece sempre soddisfacibile (ogni valutazione lo rende vero).

Definizione 1.3. Siano C_1, C_2 due clausole ed ℓ un letterale tale che $\ell \in C_1$, $\bar{\ell} \in C_2$. Il risolvente di C_1, C_2 rispetto al letterale ℓ è la clausola (in notazione insiemistica)

$$Res_\ell(C_1, C_2) = (C_1 \setminus \{\ell\}) \cup (C_2 \setminus \{\bar{\ell}\})$$

Definizione 1.4. Dato un insieme di clausole S definiamo induttivamente l'insieme $Res^n(S)$, per ogni $n \in \mathbb{N}$, in modo che $Res^0(S) = S$ e:

$$Res^{n+1}(S) = Res^n(S) \cup \{C : \exists \ell, \exists C_1 \in Res^n(S), \exists C_2 \in Res^n(S), C = Res_\ell(C_1, C_2)\}.$$

Notiamo che vale $Res^n(S) \subseteq Res^{n+1}(S)$, per ogni $n \in \mathbb{N}$. Un'altra caratterizzazione degli insiemi $Res^n(S)$ ci viene dalla definizione di *albero di risoluzione* di una clausola C da S : un tale albero T è un albero binario finito in cui le foglie sono etichettate da clausole in S , la radice è etichettata da C e ogni nodo interno è etichettato seguendo la regola di risoluzione: se i discendenti diretti del nodo sono etichettati con C_1, C_2 allora il nodo è etichettato con $Res_\ell(C_1, C_2)$, per un opportuno letterale ℓ . Se $C = \square$, l'albero T si dice anche un *albero di refutazione per risoluzione* di S . Non

è allora difficile convincersi per induzione su n che $\square \in \text{Res}^n(S)$ se e solo se esiste un albero di refutazione di S di altezza al più n (vedi gli esercizi alla fine).

Esempio 1.1. Se S è l'insieme di clausole

$$S = \{\{\neg P, U\}, \{\neg Q, U\}, \{\neg U\}, \{P, R\}, \{Q, \neg R\},\}$$

allora $\text{Res}_{-P}(\{\neg P, U\}, \{P, R\}) = \{U, R\} \in \text{Res}^1(S)$;

la clausola $\{Q, U\} = \text{Res}_{-R}(\{Q, \neg R\}, \{U, R\})$ appartiene invece a $\text{Res}^2(S)$.

Gli insiemi $\text{Res}^n(S)$ possono essere utilizzati per costruire un algoritmo, basato sulla regola di risoluzione, che controlli la soddisfacibilità o meno di un insieme di clausole. Per prima cosa si nota che:

Lemma 1.1. Se esiste $n \in \mathbb{N}$ con $C \in \text{Res}^n(S)$, allora $S \models C$.

Dimostrazione. La dimostrazione procede per induzione su n . Il caso base è banale perché se $C \in \text{Res}^0(S) = S$ allora $S \models C$.

Nel passo induttivo, supponiamo che $C \in \text{Res}^{n+1}(S)$: se $C \in \text{Res}^n(S)$ otteniamo $S \models C$ per induzione, altrimenti esistono un letterale ℓ e due clausole C_1, C_2 in $\text{Res}^n(S)$ tali che $\ell \in C_1$, $\bar{\ell} \in C_2$ e $C = \text{Res}_\ell(C_1, C_2)$. Sappiamo inoltre per induzione che le clausole C_1, C_2 sono conseguenze logiche di S . Per dimostrare che C è conseguenza logica di S ci basta allora notare che vale:

$$C_1, C_2 \models C.$$

Per dimostrare questa conseguenza logica, sia v una valutazione tale che $v(C_1) = v(C_2) = V$. Vogliamo dimostrare che $v(C) = V$, dove $C = \text{Res}_\ell(C_1, C_2)$. Dividiamo la dimostrazione in due casi, a seconda che valga $v(\ell) = V$ oppure $v(\ell) = F$. Nel primo caso, poiché $v(C_2) = V$ e $v(\bar{\ell}) = F$, deve esistere un letterale $\ell' \in C_2 \setminus \{\bar{\ell}\}$ con $V(\ell') = V$; tale letterale apparterrà anche a $C = (C_1 \setminus \{\ell\}) \cup (C_2 \setminus \{\bar{\ell}\})$ e quindi $v(C) = V$. Nel secondo caso il ragionamento è analogo al precedente e si ottiene scambiando C_2 con C_1 e ℓ con $\bar{\ell}$. $\square$

L'implicazione opposta del Lemma precedente non è valida: basta considerare l'insieme di clausole (contenente un'unica clausola) $S = \{\{P\}\}$ e $C = \{P, Q\}$: si ha $S \models C$ ma $\text{Res}^1(S) = S$, quindi $\text{Res}^n(S) = S$ per ogni n e nessuno di questi insiemi contiene C .

Possiamo però dimostrare:

Teorema 1.2. Se S è insoddisfacibile (cioè se $S \models \square$), allora esiste numero naturale n tale che $\square \in \text{Res}^n(S)$.

Dimostreremo questo Teorema per prima cosa nel caso in cui S è un insieme finito di clausole. Lo generalizzeremo poi a tutti gli insiemi di clausole utilizzando il Teorema di Compattatezza.

Nel caso in cui l'insieme S è finito, dimostreremo il Teorema 1.2 per induzione, ma il giusto parametro su cui basare l'induzione non è il numero delle clausole di S bensì il numero delle variabili proposizionali P per cui P o $\neg P$ appare in S (in altre parole: la cardinalità del linguaggio di S). Indicheremo tale numero con $n(S)$. Il risultato che permette di far funzionare l'induzione si basa sulla costruzione dell'insieme S^ℓ :

$$S^\ell := \{C \setminus \{\bar{\ell}\} : C \in S, \ell \notin C\}.$$

Ricordiamo che vale:

Lemma 1.3. Un insieme di clausole S è soddisfacibile se e solo se S^ℓ è soddisfacibile oppure $S^{\bar{\ell}}$ è soddisfacibile.

Corollario 1.4. *Un insieme di clausole S è insoddisfacibile se e solo se S^ℓ e $S^{\bar{\ell}}$ sono entrambi insoddisfacibili.*

Notiamo che ℓ non compare più nei letterali presenti nelle clausole di S^ℓ e $S^{\bar{\ell}}$. Ne segue che, se ℓ appartiene al linguaggio di S allora

$$n(S) > n(S^\ell), \quad n(S) > n(S^{\bar{\ell}}).$$

Insieme al Lemma 1.3, questo risultato, suggerisce di utilizzare una dimostrazione induttiva su $n(S)$ per provare:

Teorema 1.5. *(Completezza del metodo di risoluzione, caso finito)*

Se S è un insieme finito di clausole insoddisfacibile, allora esiste un $n \in \mathbb{N}$ tale che $\square \in Res^n(S)$.

Dimostrazione. Sia S un insieme insoddisfacibile di clausole. La dimostrazione procede per induzione sul numero $n(S)$ di variabili proposizionali P tali che P o $\bar{P}$ appartengono a clausole in S . Se $n(S) = 0$, allora $S = \{\square\}$ (S non può essere l'insieme vuoto perché tale insieme è soddisfacibile) e $\square \in S = Res^0(S)$.

Nel passo induttivo $n(S) > 0$ ed esiste almeno una variabile proposizionale P tale che P o $\bar{P}$ appartiene ad una delle clausole di S . Consideriamo i due insiemi di clausole

$$S^P = \{C \setminus \{\bar{P}\} : C \in S, P \notin C\},$$

$$S^{\bar{P}} = \{C \setminus \{P\} : C \in S, \bar{P} \notin C\}.$$

Poiché S è insoddisfacibile, dal Lemma 1.3 segue che S^P e $S^{\bar{P}}$ sono insoddisfacibili. Ma né P né $\bar{P}$ appartengono a S^P o a $S^{\bar{P}}$, quindi

$$n(S) > n(S^P), \quad n(S) > n(S^{\bar{P}}).$$

Per ipotesi induttiva, quindi, possiamo trovare un numero n_1 tale che $\square \in Res^{n_1}(S^P)$ e possiamo trovare un numero n_2 tale che $\square \in Res^{n_2}(S^{\bar{P}})$. Questo significa che esistono due alberi T_1 , di altezza n_1 e T_2 , di altezza n_2 , che sono di refutazione per $S^P, S^{\bar{P}}$, rispettivamente. Se in T_1 le etichette delle foglie sono clausole in S , (questo avviene se le etichette delle foglie in T_1 sono clausole C in S con $\ell \notin C, \bar{\ell} \notin C$), allora abbiamo trovato una derivazione della clausola vuota da S , e lo stesso vale per T_2 . Altrimenti in T_1 c'è almeno una foglia etichettata da una clausola in $S^P \setminus S$ e in T_2 c'è almeno una foglia etichettata da una clausola in $S^{\bar{P}} \setminus S$. Rietichettiamo allora gli alberi T_1, T_2 come segue.

Per quanto riguarda T_1 , aggiungiamo $\bar{P}$ alle foglie etichettate da clausole in $S^P \setminus S$. Affinché l'albero ottenuto sia un albero di risoluzione, però, dovremo cambiare le etichette anche dei nodi interni a partire dai predecessori delle foglie, per poi arrivare fino alla radice: aggiungeremo $\bar{P}$ anche alle etichette dei nodi interni, se $\bar{P}$ compare nelle etichette di almeno uno dei due figli. In questo modo si ottiene un albero di risoluzione T'_1 di $\bar{P}$ da S .

L'albero T_2 viene rietichettato in modo simile, reintroducendo opportunamente la variabile P . In questo modo si ottiene un albero di risoluzione T'_2 di P da S .

Un ultimo passo di risoluzione consente allora di costruire un albero di risoluzione di $\square$ da S . Per quanto detto, questo albero avrà altezza pari a $n = \max\{n_1, n_2\} + 1$ e quindi $\square \in Res^n(S)$, come volevasi dimostrare. $\square$

Esempio Sia

$$S = \{\{Q, R\}, \{Q, \neg R\}, \{\neg Q, S\}, \{\neg S, R\}, \{\neg Q, \neg R\}\}.$$

Abbiamo:

$$S^Q = \{\{S\}, \{\neg S, R\}, \{\neg R\}\} \quad S^{-Q} = \{\{R\}, \{\neg R\}, \{\neg S, R\}\}$$

$$(S^Q)^S = \{\{R\}, \{\neg R\}\} \quad (S^Q)^{\neg S} = \{\square, \{\neg R\}\}$$

$$(S^{-Q})^S = \{\{R\}, \{\neg R\}\} \quad (S^{-Q})^{\neg S} = \{\{R\}, \{\neg R\}\}$$

$$((S^Q)^S)^R = \{\square\} \quad ((S^Q)^S)^{\neg R} = \{\square\}$$

Abbiamo quindi refutazioni di altezza zero degli insiemi $(S^Q)^S{}^R, (S^Q)^S{}^{\neg R}$. Dalla refutazione di $(S^Q)^S{}^R$ otteniamo una risoluzione di altezza zero della clausola $\{\neg R\}$ da $(S^Q)^S$; dalla refutazione di $((S^Q)^S)^{\neg R}$ otteniamo una risoluzione di altezza zero della clausola $\{R\}$ da $(S^Q)^S$. Un ultimo passo di risoluzione ci permette di ottenere una refutazione di $(S^Q)^S$.

Similmente otteniamo una refutazione di $(S^Q)^{\neg S}$.

A partire dalla refutazione di $(S^Q)^S$ possiamo, reinserendo $\neg S$, arrivare ad una risoluzione della clausola $\{\neg S\}$ da S^Q ; a partire dalla refutazione di $(S^Q)^{\neg S}$ possiamo invece, reinserendo S , arrivare ad una risoluzione della clausola $\{S\}$ da S^Q . Mettendo insieme questi alberi, con un'ultima risoluzione sul letterale S arriviamo ad una refutazione di S^Q . Si prosegue poi trovando una refutazione di S^Q , che, insieme a quella di S^Q ci permette di ottenere la refutazione di S .

Il Teorema 1.2 vale anche nel caso S non sia un insieme finito. La dimostrazione usa il Teorema di Compattatezza:

Teorema 1.6. (*Teorema di Completezza, caso generale*). *Se S è un insieme insoddisfacibile di clausole, allora esiste una refutazione per risoluzione di S .*

Dimostrazione. Abbiamo già dimostrato il caso finito nel Teorema 1.2. Se S è infinito e insoddisfacibile, dal Teorema di compattatezza segue l'esistenza di un sottoinsieme S_1 finito di S a sua volta insoddisfacibile. Per la completezza nel caso finito, sappiamo allora che esiste una refutazione per risoluzione di S_1 , e l'albero così trovato è banalmente anche una refutazione per risoluzione di S . $\square$

Notiamo infine che vale:

Lemma 1.7. *Per ogni S finito esiste un n calcolabile a partire da S tale che, per ogni $m \geq n$, vale: $Res^n(S) = Res^m(S)$.*

Dimostrazione. Sia $S = \{C_1, \dots, C_h\}$. È sufficiente allora notare che, se $C_1 \cup \dots \cup C_n = \{\ell_1, \dots, \ell_k\}$, allora per ogni n e $C \in Res^n(S)$ vale $C \subseteq \{\ell_1, \dots, \ell_k\}$; siccome ci possono essere al più 2^k clausole di questo tipo, ne segue che ogni $Res^n(S)$ ha cardinalità al più pari a 2^k ed ogni catena di inclusioni strette fra insiemi del tipo

$$S \subset Res(S) \subset Res^2(S) \subset \dots$$

avrà, al più, lunghezza 2^k . Ne segue che se $m \geq n = 2^k$ si ha $Res^n(S) = Res^m(S)$. $\square$

I teoremi 1.2, 1.1 ed il Lemma 1.7 ci danno un algoritmo per testare l'insoddisfacibilità di un insieme finito di clausole. Generiamo gli insiemi $Res^n(S)$ fino a che non troviamo in essi la clausola vuota oppure ci troviamo un n per cui $Res^n(S) = Res^{n+1}(S)$. Nel primo caso l'insieme è insoddisfacibile, nel secondo caso invece è soddisfacibile. Tale metodo può essere applicato con successo anche al problema di stabilire se una formula F è valida: basta infatti considerare un insieme di clausole che rappresenti una forma normale congiuntiva di $\neg F$ e testare questo insieme per insoddisfacibilità. In pratica dato un insieme S non si genera tutto l'insieme $Res^1(S)$, poi tutto

$Res^2(S)$, etc, nella speranza di trovare la clausola vuota, ma si cerca direttamente un albero di refutazione per S .

1.1. Esercizi.

- (1) Dimostrazione per risoluzione la validità della legge di derivazione per casi

$$((A \rightarrow C) \wedge (B \rightarrow C) \rightarrow (A \vee B \rightarrow C) :$$

- (2) Trasformare ognuna delle formule seguenti in un insieme di clausole logicamente equivalente. Utilizzare poi il metodo di risoluzione per stabilire se la formula di partenza è insoddisfacibile.

$$(P \rightarrow Q) \wedge (P \rightarrow R) \wedge P \wedge (\neg Q \vee \neg R);$$

$$[(P \rightarrow Q) \vee (P \rightarrow R)] \wedge P \wedge (\neg Q \vee \neg R);$$

$$(P \rightarrow R) \wedge (Q \rightarrow R) \wedge (P \vee Q) \wedge \neg R;$$

- (3) Sia S l'insieme di clausole

$$S = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9\},$$

dove

$$\begin{aligned} C_1 &= \{\neg P_0, \neg P_1, P_2\} \\ C_2 &= \{\neg P_0, \neg P_1, P_3\} \\ C_3 &= \{\neg P_4, \neg P_5, P_6\} \\ C_4 &= \{\neg P_6, \neg P_3, P_7\} \\ C_5 &= \{P_0\} \\ C_6 &= \{P_1\} \\ C_7 &= \{P_4\} \\ C_8 &= \{P_5\} \\ C_9 &= \{\neg P_7\} \end{aligned}$$

Dimostrare con il metodo di risoluzione che S è insoddisfacibile.

- (4) Dimostrare per induzione su n che $C \in Res^n(S)$ se e solo se esiste un albero di risoluzione di C da S di altezza minore o uguale ad n .
- (5) Dimostrare la validità delle seguenti formule con il metodo di risoluzione (considerare cioè la *negazione* della formula data, trovare un insieme di clausole logicamente equivalente a tale negazione e applicare risoluzione a questo insieme).

$$(P \vee Q) \wedge \neg P \wedge (Q \rightarrow R \vee S) \wedge (R \rightarrow S) \rightarrow S$$

$$(P \rightarrow Q) \vee (Q \rightarrow P);$$

$$(P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P);$$

(6) Siano

$$C_1 = \{l, l'\} \cup C'_1, \quad C_2 = \{\bar{l}, \bar{l}'\} \cup C'_2.$$

Considera la “regola” che permette di passare dalle clausole C_1, C_2 alla clausola $C'_1 \cup C'_2$. Tale regola è corretta? (cioè, è vero che

$$C_1, C_2 \models C'_1 \cup C'_2,$$

come avviene con risoluzione?)

(7) Considerare il seguente insieme di clausole S :

$$S = \{\{P, Q\}, \{\neg P, \neg R\}, \{\neg P, \neg S\}, \{R, S\}, \{\neg Q, \neg R\}, \{\neg Q, \neg S\}\}.$$

- i Dimostrare che S è insoddisfacibile con il metodo di risoluzione;
- ii dimostrare che non esiste alcun albero di refutazione per risoluzione di S in cui tutte le foglie siano etichettate da clausole distinte (cioè non esiste una refutazione per risoluzione che utilizza una sola volta le clausole di S).