LEZIONI DI LOGICA MATEMATICA DEL 6 E 9 OTTOBRE 2008:LOGICA PROPOSIZIONALE

1. Semantica

Definitione 1.1. *Sia v una* valutazione *proposizionale, cioè una funzione dall'insieme delle variabili proposizionali $P_1, \ldots, P_n, \ldots$ all 'insieme dei valori di verità $\{V, F\}$. La funzione v si estende alle formule proposizionali utilizzando le definizioni viste nel corso di Elementi di Logica.*

1. *Una valutazione v è* modello *di una formula α se $v(\alpha) = V$;*
2. *una formula si dice* soddisfacibile *se ha un modello,* insoddisfacibile *se non ne ha;*
3. *α è* valida *se, per ogni valutazione v vale $v(\alpha) = V$;*
4. *un insieme di formule Σ si dice* soddisfacibile *se esiste un'interpretazione v tale che $v(\alpha) = V$, per ogni $\alpha \in \Sigma$;*
5. *un insieme di formule Σ si dice* insoddisfacibile *se non è soddisfacibile.*

Esempio
Sia v una valutazione tale che $v(A) = V, v(B) = v(C) = v(D) = V$. Se α è la formula $\alpha = (\neg A \vee B) \to C \wedge D$ allora $v(\alpha) = V$ e v è modello di α.
La formula α è quindi soddisfacibile, mentre la formula $A \wedge \neg A$ non lo è, visto che tutte le possibili valutazioni la rendono falsa.
L'insieme $\Sigma = \{P, \neg P \to P, Q \vee \neg P\}$ è soddisfacibile, ed un suo modello è dato da ogni valutazione che rende vere sia P che Q.
La formula $\alpha = ((A \to B) \to A) \to A$ è valida: data una qualsiasi valutazione v, se $v(A) = V$ allora v rende vero il conseguente dell'implicazione α e quindi rende vera α; se invece $v(A) = F$, allora $(A \to B)$ è vero e quindi è falso l'antecedente $(A \to B) \to A)$ di α (perché è vero $A \to B$, ma falso A). Ne segue che $v(\alpha) = V$.
L'insieme

$$\Sigma = \{P, P \to Q, \neg Q\}$$

è insoddisfacibile.

Dato un insieme insoddisfacibile Σ, ogni insieme Σ' che lo contiene è ancora insoddisfacibile.

Per costruire un insieme infinito insoddisfacibile è allora sufficiente partire da un insieme finito insoddisfacibile e completarlo con un arbitrario insieme infinito di formule. Il Teorema di compattezza, di cui vedremo la dimostrazione nelle prossime lezioni, ci assicura che questa condizione è anche necessaria: ogni insieme infinito di formule, se è insoddisfacibile, deve contenere un sottoinsieme finito di formule che è a sua volta insoddisfacibile.

Definitione 1.2. Equivalenza Logica

1. *Due proposizioni α, β sono* logicamente equivalenti *(notazione: $\alpha \equiv \beta$) se ogni modello di α è modello di β e* <u>viceversa</u>*, ogni modello di β è modello di α;*
2. *una formula α ha come* conseguenza logica *una formula β (notazione $\alpha \models \beta$) se ogni modello di α è anche modello di β;*

(3) *un insieme di formule Σ ha come conseguenza logica una formula β (notazione $\Sigma \models \beta$) se ogni modello di Σ è anche modello di β, cioè, se per ogni valutazione v con $v(\alpha) = V$ per ogni $\alpha \in \Sigma$ si ha $v(\beta) = V$.*

Se $\Sigma = \{\alpha_1, \ldots, \alpha_n\}$ è un insieme finito di formule, allora $\alpha_1, \ldots, \alpha_n \models \beta$ sta per $\Sigma \models \beta$.

NOTA BENE Per ogni coppia di proposizioni α, β vale:

$$\alpha \equiv \beta \qquad \Leftrightarrow \qquad \alpha \leftrightarrow \beta \text{ è una tautologia;}$$

$$\alpha \models \beta \qquad \Leftrightarrow \qquad \alpha \to \beta \text{ è una tautologia;}$$

Tutte le valutazioni soddisfano (sono modelli) dell'insieme vuoto di formule. Ne segue che

$$\emptyset \models \alpha \qquad \Leftrightarrow \qquad \alpha \text{ è una tautologia.}$$

Scriveremo anche $\models \alpha$ al posto di $\emptyset \models \alpha$.

Esempio

$$\begin{array}{ccc} (A \vee B) \wedge C & \equiv & (A \wedge C) \vee (B \wedge C) \\ A \to B & \equiv & \neg A \vee B \\ \neg(A \to B) & \equiv & A \wedge \neg B \end{array}$$

1.1. **Esercizi.** Dimostrare che:

(1) $\Sigma \models \alpha \quad \Leftrightarrow \quad \Sigma \cup \{\neg\alpha\}$ è insoddisfacibile;
(2) se $\Sigma = \{\alpha_1, \ldots, \alpha_n\}$ è un insieme finito di formule, allora

$$\Sigma \models \alpha \quad \Leftrightarrow \quad \models \alpha_1 \wedge \ldots \alpha_n \to \alpha.$$

(3) Verificare la validità o meno delle formule

$$A \to (B \to A \wedge B), \quad (A \to B) \to (\neg A \to \neg B).$$

Il problema $SODD$ della soddisfacibilità delle formule proposizionali è in NP (basta risolvere il problema usando le tavole di verità), ma non si sa se lo stesso vale per il problema della insoddisfacibilità che è in co-NP. Per il Teorema di Cook si ha:

$$SODD \in P \Leftrightarrow P = NP.$$

2. Forma Normale Congiuntiva

Definitione 2.1. *Un* letterale *è una variabile proposizionale oppure la negazione di una variabile proposiuzionale. Una formula α è in* forma normale congiuntiva *(abbreviato con FNC) se*

$$\alpha = C_1 \wedge \ldots \wedge C_n,$$

dove ogni C_i è una disgiunzione di letterali. Una formula β è una forma normale congiuntiva *di α se β è in FNC ed è equivalente ad α.*

Ad esempio, le formule $\neg A \wedge (A \vee B) \wedge (\neg B \vee C)$, $A \wedge B$, $\neg A \vee B$ sono tutte in forma normale congiuntiva, mentre $P \wedge (\neg P \vee Q)$ è una FNC di $P \wedge (P \to Q)$.

Lemma 2.1. *Per ogni formula α esiste almeno una forma normale congiuntiva β di α.*

Dimostrazione. Con l'algoritmo di Fitting, oppure eliminando $\to, \leftrightarrow$ dalla formula utilizando le equivalenze

$$\alpha \to \beta \equiv \neg\alpha \lor \beta, \quad \alpha \leftrightarrow \beta \equiv (\neg\alpha \lor \beta) \land (\neg\beta \lor \alpha);$$

poi si portano le negazioni di fronte alle variabili proposizionali utilizzando le equivalenze

$$\neg(\alpha \land \beta) \equiv \neg\alpha \lor \neg\beta, \quad \neg(\alpha \lor \beta) \equiv \neg\alpha \land \neg\beta, \quad \neg\neg\alpha \equiv \alpha;$$

si ottiene così una formula con i soli connettivi $\lor, \land$ applicati a letterali. Per ottenere una forma normale congiuntiva è allora sufficiente estrarre le congiunzioni più interne utilizzando ripetutamente le equivalenze

$$\alpha \lor (\beta \land \gamma) \equiv (\alpha \lor \beta) \land (\alpha \lor \gamma), \qquad (\beta \land \gamma) \lor \alpha \equiv (\beta \lor \alpha) \land (\gamma \lor \alpha)$$

□

Esempio

$$\neg(A \to B) \lor \neg C \equiv \neg(\neg A \lor B) \lor \neg C \equiv (A \land \neg B) \lor \neg C \equiv (A \lor \neg C) \land (\neg B \lor \neg C).$$

Quindi la formula $(A \lor \neg C) \land (\neg B \lor \neg C)$ è una FNC di $\neg(A \to B) \lor \neg C$.

Definitione 2.2. *Un* letterale ℓ *è una variabile proposizionale (letterale* positivo*) oppure la negazione di una variabile proposizionale (letterale* negativo*).*

Se ℓ è un letterale, allora

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

Definitione 2.3. *Una* clausola *è una disgiunzione di un numero finito di letterali:* $C = \ell_1 \lor \dots \ell_n$.

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

Da un punto di vista sintattico la clausola $P \lor Q$ è distinta dalla clausola $Q \lor P$, 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 clausola $P \lor \neg Q$, $\neg Q \lor P$ con l'insieme $\{P, \neg Q\}$. Abbiamo quindi identificato le clausole con insiemi finiti non vuoti di letterali. Generalizzando questa corrispondenza, chiameremo *clausola vuota* l'insieme vuoto di letterali, e la indicheremo con il simbolo $\Box$.

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 S sia insoddisfacibile: ad esempio, se S contiene la clausola vuota oppure le due clausole $\{\ell\}, \{\bar{\ell}\}$, per un dato letterale ℓ. Un esempio un poco più complicato si ottiene considerando due letterali ℓ_1, ℓ_2, ed un insieme S che contiene le clausole:

$$\{\ell_1, \ell_2\}, \quad \{\bar{\ell}_1, \ell_2\}, \quad \{\ell_1, \bar{\ell}_2\}, \quad \{\bar{\ell}_1, \bar{\ell}_2\}.$$

Possiamo costruire un insieme un poco più complesso, ma ancora soddifacibile, considerando un terzo letterale ℓ_3 e aggiungendolo, una volta semplice e una volta negato, a tutti gli insiemi precedenti. L'insieme ottenuto sarà ancora insoddisfacibile (vedi esercizio 5). L'insieme vuoto $S = \emptyset$ di clausole è invece sempre soddisfacibile (ogni valutazione lo rende vero).

Proviamo ora, invece di aggiungere letterali, a toglierli.

Sia S un insieme di clausole ed ℓ un letterale che compare in S. Dividiamo S in tre sottoinsiemi,

$$S = S_1 \cup S_2 \cup S_3 \cup S_4$$

dove S_1 consiste nelle clausole di S che contengono ℓ ed $\overline{\ell}$, S_2 consiste nelle clausole di S che contengono ℓ ma non $\overline{\ell}$, S_3 consiste nelle clausole di S che contengono $\overline{\ell}$ ma non ℓ, S_4 consiste nelle clausole di S che non contengono nè ℓ nè $\overline{\ell}$.

Sia

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

Si ha

$$S^\ell = \{C \setminus \{\overline{\ell}\} : C \in S_3\} \cup S_4,$$
$$S^{\overline{\ell}} = \{C \setminus \{\ell\} : C \in S_2\} \cup S_4.$$

Si ha

Lemma 2.2. *Sia S un insieme di clausole ed ℓ un letterale. Si ha:*

$$S \text{ è soddisfacibile } \Leftrightarrow S^\ell \text{ è soddisfacibile, oppure } S^{\overline{\ell}} \text{ è soddisfacibili.}$$

Dimostrazione. ($\Rightarrow$) Se v è una valutazione che rende vera S e $v(\ell) = V$ allora $v(S^\ell) = V$, mentre se $v(\ell) = F$ allora $v(S^{\overline{\ell}}) = V$.

($\Leftarrow$) Supponiamo che esista una valutazione v con $v(S^\ell) = V$. Poiché nessuna clausola di S^ℓ contiene ℓ o $\overline{\ell}$, possiamo variare il valore di verità di v su ℓ, senza che questo influisca sulla verità delle formule di S^ℓ. Possiamo allora supporre che esista una valutazione v con $v(\ell) = V$ e $v(S^\ell) = V$. Mostriamo che $v(S) = V$. Questo è ovvio per le clausole in $S_1 \cup S_3$ poichè tali clausole sono in S^ℓ oppure contengono il letterale ℓ. Se invece $C \in S_2$, allora $C' = C \setminus \{\overline{\ell}\} \in S^\ell$ e quindi $v(C') = V$. A maggior ragione si avrà $v(C) = V$. $\square$

Corollario 2.3. *Un insieme di clausole S è insoddisfacibile se e solo se S^ℓ e $S^{\bar{l}}$ sono entrambi insoddisfacibili.*

Esempi

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

dove

$$\begin{array}{lcl} C_1 & = & \{P, \neg Q\} \\ C_2 & = & \{P, R\} \\ C_3 & = & \{\neg P, \neg Q\} \\ C_4 & = & \{\neg P, S\} \\ C_5 & = & \{Q, \neg R\} \\ C_6 & = & \{Q, \neg U\} \end{array}$$

Si ha:

$$S^P = \{\{\neg Q\}, \{S\}, C_5, C_6\}, \qquad S^{\overline{P}} = \{\{\neg Q\}, \{R\}, C_5, C_6\}.$$

Inoltre, $\square \in (S^P)^Q$, quindi $(S^P)^Q$ è insoddisfacibile, mentre

$$(S^P)^{\neg Q} = \{\{S\}, \{\neg R\}, \{\neg U\}\};$$

siccome l'insieme $(S^P)^{\neg Q}$ è soddisfatta dalla valutazione v tale che $v(S) = v(\neg R) = v(\neg U) = V$, se poniamo $v(\neg Q) = V$ riusciamo a soddisfare anche S^P e se poniamo $v(P) = V$ la valutazione v soddisferà anche S.

Se invece consideriamo l'insieme

$$S = \{C_1, C_2, C_3, C_4\},$$

dove

$$\begin{aligned} C_1 &= \{\neg Q, \neg R\} \\ C_2 &= \{R\} \\ C_3 &= \{P, Q\} \\ C_4 &= \{\neg P, \neg Q\} \end{aligned}$$

abbiamo:

$$S^P = \{\{\neg Q, \neg R\}, \{R\}, \{\neg Q\}\}, \qquad (S^P)^Q = \{\{\neg R\}, \{R\}, \square\}, \qquad ((S^P)^Q)^R = \{\square\}.$$

In modo analogo, si dimostra che la clausola vuota appartiene ad ogni insieme

$$((S^{\ell_1})^{\ell_2})^{\ell_3},$$

per letterali relativi a variabili distinte.

Le costruzioni $S^\ell, S^{\bar\ell}$ ci permettono di passare da un insieme di clausole S con $L(S) = n$ a insiemi che hanno un letterale in meno. Sfrutteremo questa proprietà per dimostrare la Completezza del metodo di Risoluzione.

3. Risoluzione

Definitione 3.1. *Siano $C_1.C_2$ due causole 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)*

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

Definitione 3.2. *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 risoluazione: 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 $\square \in Res^n(S)$ se e solo se esiste un albero di refutazione di S di altezza al più n (vedi gli esercizi alla fine).

Esempio 3.1. *Se S è l'insieme di clausole*

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

allora $Res_{\neg P}(\{\neg P, U\}, \{P, R\}) = \{U, R\} \in Res^1(S)$;
la clausola $\{Q, U\} = Res_{\neg R}(\{Q, \neg R, \}, \{U, R\})$ appartiene invece a $Res^2(S)$.

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

Lemma 3.1. *Se esiste $n \in N$ con $C \in Res^n(S)$, allora $S \models C$.*

Dimostrazione. La dimostrazione procede per induzione su n. Il caso base è banale perché se $C \in Res^0(S) = S$ allora $S \models C$.
Nel passo induttivo, supponiamo che $C \in Res^{n+1}(S)$: se $C \in Res^n(S)$ otteniamo $S \models C$ per induzione, altrimenti esistono un letterale ℓ e due clausole C_1, C_2 in $Res^n(S)$ tali che $\ell \in C_1$, $\overline{\ell} \in C_2$ e $C = Res_\ell(C_1, C_2)$. 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 dimostare questa conseguenza logica, sia v una valutazione tale che $v(C_1) = v(C_2) = V$. Vogliamo dimostrare che $v(C) = V$, dove $C = Res_\ell(C_1, C_2)$. La dimostrazione è lasciata per esercizio. □

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 $Res^1(S) = S$, quindi $Res^n(S) = S$ per ogni n e nessuno di questi insiemi contiene C.

Possiamo però dimostrare:

Teorema 3.2. *Se S è insoddisfacibile (cioè se $S \models \square$), allora esiste numero naturale n tale che $\square \in Res^n(S)$.*

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 Terorema di Compattezza.

Nel caso in cui l'insieme S è finito, dimostreremo il Teorema 3.2 per induzione, ma il giusto parametro su cui basare l'induzione non è il numero delle clausole di S bensí il numero delle variabili proposizionali P per cui P o $\neg P$ appare in S (in altre parole: la cardinalità del linguaggio $L(S)$). Indicheremo tale numero con $n(S)$. Il risultato che permette di far funzionare l'induzione si basa sulla costruzione dell'insieme S^ℓ:

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

Ricordiamo che vale:

Lemma 3.3. *Un insieme di clausole S è soddisfacibile se e solo se S^ℓ è soddisfacibile oppure $S^{\overline{\ell}}$ è soddisfacibile.*

Corollario 3.4. *Un insieme di clausole S è insoddisfacibile se e solo se S^ℓ e $S^{\overline{l}}$ sono entrambi insoddisfacibili.*

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

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

Insieme al Lemma 3.3, questo risultato, suggerisce di utilizzare una dimostrazione induttiva su $n(S)$ per provare:

Teorema 3.5. *(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 dimsotrazione procede per induzione sul numero $n(S)$ di variabili proposizionali P tali che P o $\overline{P}$ 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 $\overline{P}$ appartiene ad una delle clausole di S. Consideriamo i due insiemi di clausole

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

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

Poiché S è insoddisfacibile, dal Lemma 3.3 segue che S^P e $S^{\overline{P}}$ sono insoddisfacibili. Ma né P né $\overline{P}$ appartengono a S^P o a $S^{\overline{P}}$, quindi

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

Per ipotesi induttiva, quindi, possiamo trovare un numero n_1 tale che $\Box \in Res^{n_1}(S^P)$ e possiamo trovare un numero n_2 tale che $\Box \in Res^{n_2}(S^{\overline{P}})$. Questo significa che esistono due alberi T_1, di altezza n_1 e T_2, di altezza n_2, che sono di refutazione per $S^P, S^{\overline{P}}$, rispettivamante. 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 $l \notin C$, $\bar{l} \notin C$), 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 $S^{\overline{P}} \setminus S$. Rietichettiamo allora gli alberi T_1, T_2 come segue.

Per quanto riguarda T_1, aggiungiamo $\overline{P}$ 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 $\overline{P}$ anche alle etichette dei nodi interni, se $\overline{P}$ compare nelle etichette di almeno uno dei due figli. In questo modo si ottiene un albero di risoluzione T_1' di $\overline{P}$ 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 $\Box$ da S. Per quanto detto, questo albero avrà altezza pari a $n = max\{n_1, n_2\} + 1$ e quindi $\Box \in Res^n(S)$, come volevasi dimostrare. $\Box$

Esempio Sia

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

Abbiamo:

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

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

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

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

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 risoluazione 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 $S^{\overline{Q}}$, che, insieme a quella di S^Q ci permette di ottenere la refutazione di S.

Il Teorema 3.2 vale anche nel caso S non sia un insieme finito. La dimostrazione però usa il Teorema di Compattezza, che dimostreremo nella prossima sezione.

4. Il Teorema di Compattezza per la Logica Proposizionale

Sia $UNSAT$ l'insieme degli insiemi di clausole insoddisfacibili:

$$UNSAT = \{S : S \text{ è un insieme di clausole insoddisfacibile}\}.$$

Cerchiamo di capire meglio come sono fatti gli insiemi di clausole in $UNSAT$. Se, ad esempio, S contiene un insieme finito insoddisfacibile, allora $S \in UNSAT$. Vedremo che vale anche il viceversa: se un insieme di clausole è insoddisfacibile, allora appartiene ad $UNSAT$.

Per dimostrare questa proprietà, utilizzeremo la successione d'insiemi $(U_n)_{n\in N}$ definita induttivamente da:

$$U_0 = \{S : \square \in S\},$$
$$U_{n+1} = U_n \cup \{S : \text{ esiste un letterale } \ell \text{ che compare in } S \text{ tale che } S^\ell \in U_n, S^{\overline{\ell}} \in U_n\}.$$

Notiamo che:

$$S \in U_0 \Leftrightarrow \square \in S,$$
$$S \in U_1 \setminus U_0 \Leftrightarrow \square \notin S \text{ e } \exists \ell \quad \{\ell\} \in S \text{ e } \{\overline{\ell}\} \in S.$$

Lemma 4.1. *Ogni insieme di clausole $S \in U_n$ possiede un sottoinsieme finito insoddisfacibile,*

Dimostrazione. Per induzione su n. Ovvio per $n = 0$, perché se $S \in U_0$ allora $\square \in S$. Per il passo induttivo, sappiamo che

$$U_{n+1} = \{S : \exists \ell, S^\ell \in U_n, S^{\overline{\ell}} \in U_n\} \cup U_n.$$

Se $S \in U_{n+1}$ ci sono quindi due possibilità: la prima, in cui $S \in U_n$, ci assicura per induzione che S ha un sottoinsieme finito insoddisfacibile, la seconda, in cui esiste un letterale ℓ tale che $S^\ell \in U_n, S^{\overline{\ell}} \in U_n$. In quest'ultimo caso, sappiamo per induzione che esistono due sottoinsiemi finiti $F \subseteq S^\ell, G \subseteq S^{\overline{\ell}}$. Esercizio: terminare la dimostrazione (la dimostrazione completa si troverà nella prossima dispensa).

$\square$

4.1. Esercizi.

(1) Trasformare ognuna delle formule seguenti in un insieme di clausole logicamente equivalente e stabilire se tale insieme è insoddisfacibile utilizzando il metodo di risoluzione.

$$((A \to C) \land (B \to C) \to (A \lor B \to C);$$

$$(P \to Q) \land (P \to R) \land P \land (\neg Q \lor \neg R);$$

$$[(P \to Q) \lor (P \to R)] \land P \land (\neg Q \lor \neg R);$$

$$(P \to R) \land (Q \to R) \land (P \lor Q) \land \neg R;$$

(2) Descrivere un semplice algoritmo per stabilire se una formula in FNC è una tautologia.

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

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

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

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

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

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

(5) 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

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

Dimostrare con il metodo di risoluzione che S è insoddisfacibile.

(6) Dimostrare la *validità* delle seguenti formule, dimostrando che la negazione della formula è insoddisfacibile 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).

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

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

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

(7) Siano

$$C_1 = \{l, l'\} \cup C_1', \qquad C_2 = \{\bar{l}, \bar{l}'\} \cup C_2'.$$

Considera la "regola" che permette di passare dalle clausole C_1, C_2 alla clausola $C_1' \cup C_2'$. Tale regole è corretta? (cioè , è vero che

$$C_1, C_2 \models C_1' \cup C_2',$$

come avviene con risoluzione?)

(8) 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).

(9) Sia S un insieme di clausole (anche infinito), ℓ un letterale e C una clausola che non contiene ℓ; dimostrare che vale:

$$S \models C \Rightarrow S^{\ell} \models C \setminus \{\bar{\ell}\}.$$

(10) Dato un insieme di clausole e un ℓ un letterale tale che né ℓ, né $\bar{\ell}$ compaiono in clausole di S, definiamo

$$S_\ell = \{C \cup \{\bar{\ell}\} : C \in S\}.$$

Dimostrare che:

$$S \text{ è insoddisfacibile } \Leftrightarrow S_\ell \cup S_{\bar{\ell}} \text{ è insoddisfacibile .}$$

A cosa è uguale l'insieme $(S_\ell \cup S_{\bar{\ell}})^{\ell}$?

(11) Dimostrare che una formula proposizionale è sempre equisoddisfacibile ad un insieme di clausole in cui ogni clausola contiene al più 3 letterali. (Suggerimento: dimostrare per prima cosa che se A è una variabile che non appartiene ad una clausola $C = \{A_1, \ldots, A_n\}$ e S è un insieme che contiene C, allora S è soddisfacibile se e solo se l'insieme di clausole $(S \setminus \{C\}) \cup \{\{A_1, A_2, A\}, \{A_3, \ldots, A_k, \neg A\}\}$ lo è)

(12) Dimostrare che esiste una formula della logica proposizionale che non è logicamente equivalente ad alcun insieme di clausole che contengono cal più 3 letterali (anche aggiungendo nuove variabili).

(13) Considera la frase A seguente:

"Se il Congresso rifiuta di promulgare nuove leggi, allora lo sciopero non finirà, a meno che duri più di un anno e il Presidente della fabbrica si dimetta".

Rappresentare questa frase nei seguenti modi:

i con una formula della logica proposizionale;

ii con un insieme di clausole;

Se aggiungiamo l'informazione B seguente,

B="il congresso rifiuterà di promulgare nuove leggi" e "lo sciopero è terminato"

possiamo dimostrare che

C= lo sciopero è durato più di un anno e il Presidente si è dimesso? In altre parole, si ha $A, B \models C$?

(Suggerimento: tradurre il connettivo P "a meno che" Q con $\neg Q \to P$).

(14) Dato un insieme di clausole S, sia $L(S)$ l'insieme delle variabili proposizionali che appaiono in clausole di S. Se $(U_n)_{n\in\mathbb{N}}$ è la successione di insiemi definita nel paragrafo 3, dimostrare che, per ogni n,

$$U_n = \{S : \exists S' \subseteq S, S' \text{ finito e insoddisfacibile e } |L(S')| \leq n\}.$$