Altri sistemi deduttivi - risoluzione

Corso di Logica

Brunella Gerla

Università dell'Insubria, Varese
brunella.gerla@uninsubria.it

Presentiamo altri metodi che permettono di stabilire se una formula è una tautologia o se è soddisfacibile senza guardare la tavola di verità . Tali metodi vengono chiamati metodi o sistemi **deduttivi**.

In questo ambito parleremo di **dimostrazioni** (che sono appunto tutti i passi che ci permettono di dire che una certa formula è vera) e di **teoremi**.

Storicamente, il primo metodo per dedurre formule *vere* è quello utilizzato nei trattati di geometria di Euclide (323 a.C - 285 a.C): si parte da un insieme di formule che sono intuitivamente vere (per esempio tra due punti passa una retta sola) e da questi si derivano tutti gli altri.

Un sistema deduttivo di questo tipo è chiamato **sistema alla Hilbert** (1862-1943) o **assiomatico**.

Metodo assiomatico

Definizione

Gli **assiomi** della logica proposizionale:

1. $A \to (B \to A)$
2. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
3. $\neg A \to (A \to B)$
4. $(A \to B) \to ((\neg A \to B) \to B)$.

La **regola di deduzione** è il modus ponens: da A e $A \to B$ ricavo B.

Definizione

Una **dimostrazione** di una formula P è una sequenza di formule

$$P_1, \ldots, P_n$$

tale che $P_n = P$ e per ogni $i = 1, \ldots, n-1$ si ha che o P_i è un assioma (o una istanza di assioma) oppure P_i si ottiene da due formule precedenti utilizzando il modus ponens, cioè , esistono $j, k < i$ tali che $P_k = P_j \to P_i$.

Definizione

Una formula P è un **teorema** se esiste una dimostrazione di P. In questo caso scriviamo $\vdash_H P$.

Esempio

Proviamo che la formula $A \to A$ è un teorema. Dobbiamo quindi mostrare una dimostrazione di $A \to A$, cioè una opportuna sequenza di formule. Consideriamo l'assioma 2, ponendo $A \to A$ al posto di B e A al posto di C. Abbiamo

$$P_1 = (A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A)).$$

Consideriamo quindi l'assioma 1 con $A \to A$ al posto di B:

$$P_2 = A \to ((A \to A) \to A)).$$

La formula

$$P_3 = (A \to (A \to A)) \to (A \to A))$$

si ottiene da P_1 e P_2 tramite modus ponens.

$$P_3 = (A \to (A \to A)) \to (A \to A))$$

Consideriamo ora di nuovo l'assioma 1 ma con A al posto di B:

$$P_4 = A \to (A \to A)\,.$$

Applicando il modus ponens abbiamo

$$P_5 = A \to A$$

che è la formula che volevamo dimostrare.

Teorema (Teorema di completezza e correttezza)

P è una tautologia se e solo se è un teorema:

$$\models P \quad \Leftrightarrow \quad \vdash_H P\,.$$

La dimostrazione della correttezza (se P è un teorema allora è una tautologia) non è difficile (provare a pensarci...).

Invece dimostrare la completezza (tutto quello che è vero è anche dimostrabile) è un po' più complicato.

Metodo di risoluzione - Clausole

Vediamo ora un altro metodo di dimostrazione che in alcuni casi risulta molto efficiente e che è alla base della programmazione logica.

Definizione

Una **clausola** è una disgiunzione di letterali.

Per comodità possiamo vedere una clausola come un insieme di letterali. Infatti:

- La disgiunzione è commutativa e associativa, quindi non importa in che ordine scriviamo i letterali;
- la disgiunzione è idempotente e quindi non importa ripetere due volte lo stesso letterale.

Esempio

La formula $(X \vee Y) \vee \neg Z$ è una clausola che possiamo rappresentare come $\{X, Y, \neg Z\}$.

Dato che ogni formula è equivalente ad una formula in forma normale congiuntiva, possiamo vedere una formula come una congiunzione di clausole.
Anche in questo caso, useremo una notazione insiemistica: una formula può essere rappresentata come un insieme di clausole, cioè un insieme di insiemi di letterali.

Esempio

La formula $(X \vee \neg Y) \wedge (Z \vee \neg X \vee Y)$ si rappresenta come $\{\{X, \neg Y\}, \{Z, \neg X, Y\}\}$.

Definizione

La **clausola vuota** (denotata con $\square$) è l'insieme vuoto di letterali.

Non bisogna confondere la clausola vuota $\square$ con l'insieme vuoto di clausole che rappresenteremo con l'usuale simbolo $\emptyset$.

Semantica delle clausole

Adattando la nozione di valutazione agli insiemi di clausole abbiamo:

Definizione

Sia S un insieme di clausole. Una valutazione è una funzione $v : Var \to \{0, 1\}$. Per definire quando v soddisfa S (in simboli $v \vDash S$) procediamo nel seguente modo:

- Se $X \in Var$ allora $v \vDash X$ se $v(X) = 1$ e $v \vDash \neg X$ se $v(X) = 0$;
- per ogni clausola $C \in S$, con $C = \{L_1, \ldots, L_n\}$ si ha $v \vDash C$ se esiste $i \in \{1, \ldots, n\}$ tale che $v \vDash L_i$;
- $v \vDash S$ se per ogni $C \in S$ si ha $v \vDash C$.

Nei casi particolari della clausola vuota e dell'insieme vuoto di clausole abbiamo che

La clausola vuota $\square$ è sempre insoddisfacibile.
Ogni insieme di clausole che contiene $\square$ è insoddisfacibile.
L'insieme vuoto di clausole $\emptyset$ è soddisfatto da ogni interpretazione.

Definizione

Due insiemi di clausole S e S' sono logicamente equivalenti ($S \equiv S'$) se sono soddisfatti dalle stesse valutazioni.
S' è una conseguenza logica di S se ogni valutazione che soddisfa S soddisfa anche S'.

Proposizione

Una clausola è una tautologia se e solo se contiene un letterale e la sua negazione.
Sia S' l'insieme ottenuto da S cancellando una tautologia. Allora $S \equiv S'$.

Esempio

$S = \{\{X, Y, \neg Z\}, \{X, \neg Y\}, \{X, \neg X, Y\}\}$ è logicamente equivalente a $S' = \{\{X, Y, \neg Z\}, \{X, \neg Y\}\}$. Controllare che le formule $(X \vee Y \vee \neg Z) \wedge (X \vee \neg Y) \wedge (X \vee \neg X \vee Y)$ e $(X \vee Y \vee \neg Z) \wedge (X \vee \neg Y)$ sono logicamente equivalenti.

Vogliamo trovare un metodo veloce per stabilire se una formula in CNF (e quindi se un insieme di clausole) è soddisfacibile.
Questo problema è chiamato CNF-SAT e si dimostra essere NP-completo (come il problema SAT).

Definizione

Siano C_1 e C_2 due clausole tali che esista un letterale $L \in C_1$ e $\neg L \in C_2$. Allora il **risolvente** R di C_1 e C_2 (rispetto al letterale L) è la clausola

$$R = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{\neg L\}).$$

Diciamo anche che R si ottiene per **risoluzione** da C_1 e C_2.

Esempio

Se $C_1 = \{\neg X, \neg Y, Z\}$ e $C_2 = \{Y, H, Z\}$ allora $R = \{\neg X, Z, H\}$ è il risolvente di C_1 e C_2 rispetto a Y.

Proposizione: correttezza della risoluzione

Il risolvente R è conseguenza logica della congiunzione $\{C_1, C_2\}$.

Dimostrazione.

Sia v una valutazione tale che $v \vDash C_1$ e $v \vDash C_2$. Questo vuol dire che esistono $M \in C_1$ e $N \in C_2$ tali che $v(M) = v(N) = 1$.
Se fosse $M = L$ e $N = \overline{L}$ non potrebbe essere $v(M) = v(N) = 1$, quindi almeno uno tra M e N appartiene a R e quindi R è soddisfacibile. $\square$

Si ha quindi che

$$\{C_1, C_2\} \equiv \{C_1, C_2, R\}.$$

Nota che se $R = \square$ allora si ha $\{C_1, C_2\} \equiv \{C_1, C_2, \square\}$ che è insoddisfacibile e quindi:

Se da C_1 e C_2 ottengo $\square$ tramite risoluzione, allora l'insieme $\{C_1, C_2\}$ è insoddisfacibile.

Questo procedimento si può ripetere, applicando la risoluzione più volte.

Esempio

Sia $P = X \wedge (\neg X \vee Y) \wedge \neg Y$. Possiamo rappresentare P come l'insieme

$$P = \{\{X\}, \{\neg X, Y\}, \{\neg Y\}\}.$$

Applicando una prima volta la risoluzione alle prime due clausole otteniamo la clausola $\{Y\}$ e quindi

$$P \equiv P \cup \{Y\}.$$

Possiamo procedere ulteriormente e applicare la risoluzione alla nuova clausola e a $\{\neg Y\}$. Otteniamo la clausola vuota e quindi si ha che $P \equiv P \cup \{Y, \Box\}$. Quest'ultimo insieme è chiaramente insoddisfacibile e quindi anche P è insoddisfacibile.

Quindi la risoluzione ci suggerisce un modo per capire se un insieme di clausole è soddisfacibile.

Definizione

Una clausola C è derivabile **per risoluzione** da un insieme di clausole S se esiste una sequenza $C_1, \ldots, C_n$ di clausole tale che $C_n = C$ e per ogni $i = 1, \ldots, n-1$ si ha che $C_i \in S$ oppure C_i si ottiene per risoluzione da clausole di S e da qualche C_j con $j < i$.
In questo caso scriviamo

$$S \vdash_R C\,.$$

Definizione

Una **refutazione** di S è una derivazione della clausola vuota $\Box$ da S.
S è refutabile se $S \vdash_R \Box$.

Teorema

$S \vdash_R \Box$ se e solo se S è insoddisfacibile.

Esempio

Nella definizione di derivazione per risoluzione, non abbiamo fissato un ordine particolare di applicazione della risoluzione.
Partiamo da $S = \{\{X, Y, \neg Z\}, \{\neg X\}, \{X, Y, Z\}, \{X, \neg Y\}\}$ e proviamo a derivare la clausola vuota.
Applicando la risoluzione alla prima e alla terza clausola ottengo $\{X, Y\}$ e applicando la risoluzione a questa clausola e alla seconda si ottiene $\{Y\}$.
Applicando alla seconda e alla quarta si ottiene $\{\neg Y\}$.
Applicando quindi la risoluzione a $\{Y\}$ e a $\{\neg Y\}$ si ottiene $\Box$.

Cerchiamo un metodo più efficace per applicare la risoluzione.

Definizione

Se C e G sono due clausole e $C \subseteq G$ (ma $C \neq G$) allora diciamo che C **sussume** G (o che G **è sussunta** da C).

Proposizione

Sia S' l'insieme ottenuto cancellando da S tutte le clausole G sussunte da altre clausole $C \in S$. Allora $S' \equiv S$.

Esempio

Sia $S = \{\{X, Y, \neg Z\}, \{X, Y\}, \{Z, Y\}\}$ e si noti che S è soddisfatta da $v(X) = 1$, $v(Y) = 0$ e $v(Z) = 1$.
La clausola $\{X, Y\}$ sussume $\{X, Y, \neg Z\}$ e infatti l'insieme

$$S' = \{\{X, Y\}, \{Z, Y\}\}$$

è ancora soddisfatto dalla valutazione v.

Procedura di Davis-Putnam

E' un algoritmo che semplifica un insieme finito di clausole al fine di determinare se è soddisfacibile oppure no.

Definizione

Se X è una variabile, si dice che una clausola è X-*esonerata* se non contiene né X né $\neg X$.
Dato un insieme di clausole S, gli X-*risolventi* di S sono tutte le clausole che si ottengono da S facendo la risoluzione rispetto a X e $\neg X$.

Sia S l'insieme di clausole considerato.
Iniziamo con il togliere da S tutte le tautologie e le clausole sussunte.
Poi trasformiamo S con una sequenza di passi.

Procedura di Davis-Putnam

Da S otteniamo un insieme S_1 nel seguente modo:

- Si eliminano da S tutte le tautologie e tutte le clausole sussunte.
- Si sceglie una variabile X (detta il **pivot**) che occorre nella clausola più corta. Nel caso di parità di lunghezza si applica l'ordine alfabetico.
- Si aggiungono a S_1 tutte le clausole X-*esonerate* di S.
- Si aggiungono a S_1 tutti gli X-risolventi di S fatti su clausole che non sono X-esonerate.
- Si rimuovono da S_1 tutte le eventuali tautologie e le clausole sussunte.

Dopo questo primo passo la variabile X non sarà presente in S_1.

Nota che se in S ci sono solo clausole che contengono X o solo clausole che contengono $\neg X$, allora in S_1 tali clausole non saranno presenti.

Per quanto detto finora, S_1 è soddisfacibile se e solo se S è soddisfacibile.

Ripetendo questo procedimento su tutte le variabili, si arriva o ad ottenere la clausola vuota o l'insieme vuoto di clausole.

Esempio

Sia $S = \{\{X, Y, \neg Z\}, \{X, \neg Y\}, \{\neg Y, Z\}, \{\neg X, Y\}\}$. Scegliamo X come pivot. Iniziamo a mettere in S_1 le clausole X-esonerate:

$$\{\neg Y, Z\} \in S_1.$$

Gli X-risolventi tra le clausole rimaste sono $\{Y, \neg Z\}$ e $\{\neg Y, Y\}$. Poiché quest'ultima è una tautologia, si ha

$$S_1 = \{\{\neg Y, Z\}, \{Y, \neg Z\}\}.$$

Scegliamo Y come secondo pivot. Non ci sono clausole Y-esonerate, quindi passiamo alle risolventi. L'unica risolvente è $\{\neg Z, Z\}$ che è una tautologia e va quindi eliminata. Quindi

$$S_2 = \emptyset$$

e abbiamo ottenuto l'insieme vuoto di clausole.

Esempio

Sia $S = \{\{A, B, \neg C\}, \{\neg A\}, \{A, B, C\}, \{A, \neg B\}\}$ e scegliamo A come pivot. Non ci sono clausole A-esonerate, quindi calcoliamo gli A-risolventi e aggiungiamoli a S_1:

$$S_1 = \{\{B, \neg C\}, \{B, C\}, \{\neg B\}\}\,.$$

Scegliamo B come pivot, non ci sono clausole B esonerate e calcoliamo i B-risolventi:

$$S_2 = \{\{\neg C\}, \{C\}\}\,.$$

Scegliamo C come pivot, applicando la risoluzione otteniamo la clausola vuota:

$$S_3 = \{\square\}\,.$$

Teorema

Sia S un insieme di clausole nelle variabili $X_1, \ldots, X_n$. Allora dopo t passi (con $t \leq n$) l'insieme S_t è costituito solo dalla clausola vuota, oppure è vuoto. Nel primo caso S è insoddisfacibile, nel secondo caso è soddisfacibile.

Cenni di dimostrazione

Se $S_t = \{\Box\}$ allora S è insoddisfacibile per la correttezza della risoluzione.
Supponiamo invece che $S_t = \emptyset$ e troviamo una valutazione che soddisfi tutte le clausole di S.
Ad ogni passo i supponiamo di eliminare la variabile X_i (quindi in S_i ci sono le variabili $X_{i+1}, \ldots, X_t$).
Al passo t abbiamo l'insieme vuoto di clausole che è soddisfatto da qualsiasi valutazione. Per semplicità supponiamo che $t = n$ numero totale di variabili.
Supponiamo che al passo $i + 1$ l'insieme S_i sia soddisfatto da una valutazione v_{i+1} definita su $\{X_{i+1}, \ldots, X_t\}$ e procediamo a definire v_i sulla variabile X_i che soddisfi S_{i-1}.

Dimostrazione.

Nel passaggio da S_{i-1} a S_i è stata eliminata la variabile X_i e ci sono vari modi per ottenere questo:

- Se S_i è stata ottenuta da S_{i-1} *solo* raccogliendo tutte le clausole X_i-esonerate (e quindi non facendo risoluzione), allora vuol dire che in S_{i-1} ci sono solo clausole che contengono X_i o solo clausole che contengono $\neg X_i$. Nel primo caso poniamo $v_i(X_i) = 1$, nel secondo caso poniamo $v_i(X_i) = 0$;
- se invece X_i è stata eliminata utilizzando la risoluzione, allora sicuramente possiamo definire v_i su X_i in modo da soddisfare S_i (qui mancano i dettagli, vedi esempio);
- negli altri casi (eliminazione di tautologie), non è importante che valore si dà alla variabile X_i, poniamo per esempio $v_i(X_i) = 1$.

Se $t < n$ allora si deve procedere in un passo ad estendere la valutazione a più variabili... $\square$

Esempio

Consideriamo l'insieme di clausole
$S_0 = S = \{\{Y, \neg Z\}, \{X, \neg Y\}, \{\neg X, \neg Z\}, \{X\}\}$. I passi della procedure di Davis Putnam sono i seguenti:

- $S_1 = \{\{Y, \neg Z\}, \{\neg Y, \neg Z\}, \{\neg Z\}\}$;
- $S_2 = \{\{\neg Z\}\}$;
- $S_3 = \emptyset$.

Le valutazioni delle variabili sono:

- $v_3(Z) = 0$ perché S_3 si ottiene da S_2 raccogliendo la clausole Z-esonerate e S_2 contiene solo clausole contenenti $\neg Z$.
- Qui possiamo scegliere $v_2(Y) = 1$ o $v_2(Y) = 0$, entrambe soddisfano S_2;
- deve necessariamente essere $v(X) = 1$ per soddisfare l'ultima clausola di S.

Clausole di Krom

Per alcuni insiemi di clausole, la risoluzione (e quindi il metodo di Davis Putnam) è molto veloce.

Definizione

Una **clausola di Krom** è una clausola in cui compaiono al più 2 letterali.

Nota che nella procedura di Davis-Putnam se si parte da clausole di Krom, si generano sempre clausole di Krom.

Esempio

Sia $S = \{\{A, B\}, \{\neg A, C\}, \{B, C\}, \{A, \neg B\}, \{C, \neg D\}, \{B, D\}\}$ formato da clausole di Krom.
Scegliendo A come primo pivot si ottiene

$$S_1 = \{\{B, C\}, \{C, \neg D\}, \{B, D\}, \{B, C\}, \{C, \neg B\}\}.$$

Esempio

$$S_1 = \{\{B, C\}, \{C, \neg D\}, \{B, D\}, \{B, C\}, \{C, \neg B\}\}.$$

Scegliendo B come pivot si ottiene

$$S_2 = \{\{C, \neg D\}, \{C\}, \{C, D\}\}$$

e quindi scegliendo C come pivot si ottiene

$$S_3 = \emptyset\,.$$

Proposizione

Un insieme di clausole di Krom in n variabili è processato da DPP in al più n passi in ognuno dei quali si generano al più $2n^2 + n + 1$ clausole.

Dimostrazione.

Basta notare che con n variabili si possono scrivere $2n$ letterali, quindi $2n$ clausole che contengono un solo letterale, e $2n(2n-1)/2$ clausole che contengono due letterali diversi. Aggiungendo la clausola vuota si ottengono le $2n^2 + n + 1$ clausole richieste. $\square$

Il problema di stabilire se un insieme di clausole di Krom è soddisfacibile è chiamato 2-SAT ed è un esempio di problema risolvibile in tempo polinomiale (a differenza di SAT).

Esempio

Consideriamo il problema di capire se in un poligono con n lati possiamo colorare i vertici con m colori diversi in modo che due vertici adiacenti non abbiano lo stesso colore.
Per semplicità consideriamo il caso $n = 3$ e $m = 2$, cioè il problema di colorare i vertici di un triangolo con due colori diversi. (Chiaramente il problema non sarà risolvibile, lo usiamo come esempio da formalizzare nella logica proposizionale):
Introduciamo delle variabili proposizionali

$$X_{ir} \text{ sta per } \textit{il vertice i è rosso} \qquad X_{ib} \text{ sta per } \textit{il vertice i è blu}\,.$$

Il problema che vogliamo risolvere si può esprimere con le seguenti formule:

Ogni vertice o è rosso o è blu	$X_{1r} \vee X_{1b}$ $X_{2r} \vee X_{2b}$ $X_{3r} \vee X_{3b}$
Ogni vertice non può essere sia rosso che blu	$\neg X_{1r} \vee \neg X_{1b}$ $\neg X_{2r} \vee \neg X_{2b}$ $\neg X_{3r} \vee \neg X_{3b}$
Se un vertice è rosso quello adiacente è blu	$X_{1r} \to X_{2b}$ $X_{2r} \to X_{3b}$ $X_{3r} \to X_{1b}$
Se un vertice è blu quello adiacente è rosso	$X_{1b} \to X_{2r}$ $X_{2b} \to X_{3r}$ $X_{3b} \to X_{1r}$

Tali formule possono essere scritte come clausole, consideriamo l'insieme di tutte queste clausole e controlliamo se è soddisfacibile o meno.

Sia

$$
\begin{aligned}
S \;=\; & \{\{X_{1r}, X_{1b}\}, \{X_{2r}, X_{2b}\}, \{X_{3r}, X_{3b}\}, \{\neg X_{1r}, \neg X_{1b}\}, \{\neg X_{2r}, \neg X_{2b}\}, \\
& \{\neg X_{3r}, \neg X_{3b}\}, \{\neg X_{1r}, X_{2b}\}, \{\neg X_{2r}, X_{3b}\}, \{\neg X_{3r}, X_{1b}\}, \\
& \{\neg X_{1b}, X_{2r}\}, \{\neg X_{2b}, X_{3r}\}, \{\neg X_{3b}, X_{1r}\}\}
\end{aligned}
$$

e applichiamo la procedura di Davis-Putnam partendo dal pivot X_{1r}:

$$
\begin{aligned}
S_1 \;=\; & \{\{X_{2r}, X_{2b}\}, \{X_{3r}, X_{3b}\}, \{\neg X_{2r}, \neg X_{2b}\}, \{\neg X_{3r}, \neg X_{3b}\}, \\
& \{\neg X_{2r}, X_{3b}\}, \{\neg X_{3r}, X_{1b}\}, \{\neg X_{1b}, X_{2r}\}, \{\neg X_{2b}, X_{3r}\}, \\
& \cancel{\{X_{1b}, \neg X_{1b}\}}, \{X_{1b}, X_{2b}\}, \{\neg X_{1b}, \neg X_{3b}\}, \{X_{2b}, \neg X_{3b}\}\}
\end{aligned}
$$

Quindi scegliamo come pivot X_{1b}:

$$
\begin{aligned}
S_2 \;=\; & \{\{X_{2r}, X_{2b}\}, \{X_{3r}, X_{3b}\}, \{\neg X_{2r}, \neg X_{2b}\}, \{\neg X_{3r}, \neg X_{3b}\}, \\
& \{\neg X_{2r}, X_{3b}\}, \{\neg X_{2b}, X_{3r}\}, \{X_{2b}, \neg X_{3b}\}, \\
& \{\neg X_{3r}, X_{2r}\}, \cancel{\{\neg X_{3r}, \neg X_{3b}\}}, \cancel{\{X_{2b}, X_{2r}\}}, \cancel{\{X_{2b}, \neg X_{3b}\}}\}
\end{aligned}
$$

Scegliamo come pivot X_{2r}:

$$\begin{aligned} S_3 \quad = \quad & \{\{X_{3r}, X_{3b}\}, \{\neg X_{3r}, \neg X_{3b}\}, \\ & \{\neg X_{2b}, X_{3r}\}, \{X_{2b}, \neg X_{3b}\}, \\ & \cancel{\{X_{2b}, \neg X_{2b}\}}, \{X_{2b}, X_{3b}\}, \{\neg X_{2b}, \neg X_{3r}\}, \{X_{3b}, \neg X_{3r}\}\} \end{aligned}$$

Scegliamo come pivot X_{2b}:

$$\begin{aligned} S_4 \quad = \quad & \{\{X_{3r}, X_{3b}\}, \{\neg X_{3r}, \neg X_{3b}\}, \{X_{3b}, \neg X_{3r}\}, \\ & \{X_{3r}, \neg X_{3b}\}, \cancel{\{X_{3r}, X_{3b}\}}, \cancel{\{\neg X_{3b}, \neg X_{3r}\}}, \cancel{\{X_{3b}, \neg X_{3r}\}}\} \end{aligned}$$

Scegliamo come pivot X_{3r}:

$$S_5 = \{\cancel{\{X_{3b}, \neg X_{3b}\}}, \{X_{3b}\}, \{\neg X_{3b}\}\}$$

Scegliamo infine come pivot X_{3b}:

$$S_6 = \{\square\}.$$

L'insieme S di partenza quindi è insoddisfacibile

Clausole di Horn

Definizione

Una **clausola di Horn** è una clausola nella quale compare al più un letterale non negato.

Esempio

La clausola vuota $\square$ è un esempio di clausola di Horn.

Anche le clausole unitarie, che contengono cioè un solo letterale, sono clausole di Horn.

$\{A, \neg B, \neg C\}$ è una clausola di Horn, mentre $\{A, B, \neg C\}$ non lo è .

Nota che applicando la risoluzione a clausole di Horn si ottiene una clausola di Horn.

Consideriamo la clausola di Horn $C = \{A, \neg B_1, \neg B_2, \ldots, \neg B_n\}$.
Ricordando il significato delle clausola, abbiamo che C rappresenta la formula

$$A \vee \neg B_1 \vee \ldots \vee \neg B_n$$

che è equivalente a

$$A \vee \neg (B_1 \wedge \cdots \wedge B_n)$$

che è equivalente a

$$(B_1 \wedge \cdots \wedge B_n) \rightarrow A\,.$$

Vedremo meglio il ruolo delle clausole di Horn nella programmazione logica quando si studierà la logica dei predicati.

Esempio

Consideriamo le clausole $\{A, \neg B, \neg C\}$ e $\{C, \neg D\}$. applicando la risoluzione alla variabile C otteniamo la clausola $\{A, \neg B, \neg D\}$ che è ancora una clausola di Horn.

Proviamo a scrivere le formule corrispondenti:

$$B \wedge C \to A \qquad D \to C\,.$$

Applicando la risoluzione si ottiene

$$B \wedge D \to A\,.$$

Proposizione

Se S è un insieme di m clausole di Horn con n variabili, allora la procedura DPP termina dopo n passi in ognuno dei quali non vengono generate più di m clausole.

Si può dimostrare che il problema della soddisfacibilità di insiemi di clausole di Horn è un problema polinomiale (a differenza del problema generale che è NP-completo).

Definizione

Una prova per **risoluzione lineare** di una clausola C a partire da un insieme di clausole S è una sequenza di clausole $C_1, \dots, C_n$ tali che $C_n = C$, e ogni C_i si ottiene per risoluzione da C_{i-1} e da una clausola B che o appartiene a S oppure è stata ottenuta precedentemente per risoluzione.

Nella risoluzione lineare abbiamo quindi un vincolo sull'ordine da utilizzare per applicare la risoluzione.

Esempio

Sia $S = \{\{A, B\}, \{A, \neg B\}, \{\neg A, B\}, \{\neg A, \neg B\}\}$. Una derivazione di $\Box$ da S tramite risoluzione lineare è la sequenza:

$$\{A, B\}, \{A, \neg B\}, \{A\}, \{\neg A, B\}, \{B\}, \{\neg A, \neg B\}, \{\neg A\}, \{A\}, \Box, .$$

Proposizione

La risoluzione lineare è completa (per refutazione), cioè se un insieme di clausole S è insoddisfacibile allora esiste una derivazione tramite risoluzione lineare della clausola vuota da S.

Provare a trasformare in risoluzione lineare gli esempi precedenti.

Definizione

Una prova per **risoluzione da input** di una clausola C da un insieme di clausole S è una sequenza $C_1, \ldots, C_n$ tale che $C_n = C$ e ad ogni passo una delle clausole risolventi è un elemento di S.

La risoluzione da input non è completa per refutazione. Ad esempio dall'insieme insoddisfacibile

$$S = \{\{A, B\}, \{A, \neg B\}, \{\neg A, B\}, \{\neg A, \neg B\}\}$$

non è possibile derivare la clausola vuota utilizzando la risoluzione da input.

Proposizione

La risoluzione da input è completa rispetto ad insiemi di clausole di Horn.

Esempio

Sia $S = \{\{A, \neg B, \neg C\}, \{B, \neg C\}, \{C, \neg D\}, \{D\}, \{\neg A\}\}$. L'insieme S è insoddisfacibile ed esiste una derivazione di $\Box$ da S usando la risoluzione da input.