Logica Proposizionale

Domenico Cantone

Dipartimento di Matematica e Informatica
Università di Catania

Logica Computazionale, A.A. 2006/07

Outline

Outline

Sintassi della logica proposizionale

Definizione

- Una formula proposizionale atomica è una lettera proposizionale (P_1, P_2, ...), $\top$ o $\bot$.
- L'insieme delle formule proposizionali è il più piccolo insieme $\mathbf{P}$ tale che
  - se A è una formula atomica, allora $A \in \mathbf{P}$;
  - se $X \in \mathbf{P}$, allora $\neg X \in \mathbf{P}$;
  - se $\circ \in \{\wedge, \vee, \rightarrow, \ldots\}$ è un connettivo proposizionale binario e $X, Y \in \mathbf{P}$, allora $(X \circ Y) \in \mathbf{P}$.

Semantica della logica proposizionale

Definizione

Una valutazione booleana è un'applicazione $v : \mathbf{P} \longrightarrow \{\mathbf{t}, \mathbf{f}\}$ tale che

- $v(\top) = \mathbf{t}$ e $v(\bot) = \mathbf{f}$;
- $v(\neg X) = \neg v(X)$;
- $v(X \circ Y) = v(X) \circ v(Y)$, dove $\circ$ sta per un qualsiasi connettivo binario.

Definizione

- Una formula proposizionale X è una tautologia se $v(X) = \mathbf{t}$ per ogni valutazione booleana v.
- Un insieme S di formule proposizionali è soddisfacibile se esiste una valutazione booleana v tale che $v(X) = \mathbf{t}$ per ogni $X \in S$.

Outline

- Per dimostrare un enunciato φ in una teoria $\mathbb{T}$ con il metodo dei tableau, si ricerca sistematicamente una refutazione di $\neg\varphi$ applicando opportune regole di inferenza basate sulla semantica dei connettivi logici, quantificatori, e, se presenti, costrutti extra-logici.
- In generale, se la teoria $\mathbb{T}$ è indecidibile, la ricerca di una refutazione può non terminare.
- Ma in caso di terminazione,
  - se i rami di un tableau per φ sono tutti contraddittori, il tableau stesso costituisce una dimostrazione leggibile del fatto che φ è un teorema della teoria $\mathbb{T}$;
  - altrimenti, se il tableau ha un ramo non contraddittorio ϑ, allora solitamente l'informazione su ϑ può essere utilizzata per costruire un modello per $\mathbb{T} \cup \{\neg\varphi\}$, cioè un controesempio di φ in $\mathbb{T}$.

Ulteriori argomentazioni a favore dei tableaux semantici

In molti sistemi di tableaux semantici, le regole di inferenza sono analitiche, nel senso che consentono soltanto la derivazione di sottoformule della formula iniziale o della sua negazione.
È la proprietà di analiticità che rende i tableaux particolarmente adatti all'automazione.

Ulteriori argomentazioni a favore dei tableaux semantici (cntd)

Ulteriori argomentazioni a favore dei tableaux semantici sono:

- i calcoli a tableaux possono essere facilmente estesi mediante nuove regole, permettendo così, in casi favorevoli, generalizzazioni a teorie più espressive;
- i calcoli a tableaux possono essere facilmente dotati di euristiche e vari tipi di controlli che consentono, ad esempio, di
  - disattivare temporaneamente alcune regole, oppure
  - introdurre nuove regole derivate (cioè, regole che non cambiano il potere inferenziale del calcolo), oppure
  - imporre dei criteri di precedenza sull'ordine con cui le regole debbono essere applicate.

Mediante un calcolo a tableau per la logica proposizionale si intende formalizzare il seguente tipo di ragionamento:

Per dimostrare che un dato enunciato proposizionale ***D*** è una tautologia, cioè è vero per ogni interpretazione delle sue variabili, si prova a derivare una contraddizione dall'assunzione che esista una valutazione booleana ***v*** rispetto alla quale ***D*** risulti falso.

Nel corso di tale processo, possono essere utilizzate le seguenti regole di inferenza, basate sulla semantica dei connettivi:

(a) se la formula C è vera ed ha la forma $\neg\neg A$, allora A è vera;
(b) se C è vera ed ha la forma $A \wedge B$, allora A e B sono entrambe vere;
(c) se C è vera ed ha la forma $\neg(A \wedge B)$, allora almeno una tra $\neg A$ e $\neg B$ è vera;
(d) se C è vera ed ha la forma $A \rightarrow B$, allora almeno una tra $\neg A$ e B è vera;
(e) se C è vera ed ha la forma $\neg(A \rightarrow B)$, allora A e $\neg B$ sono entrambe vere;
(f) ecc.

(la verità è ovviamente intesa rispetto ad una data valutazione booleana v).

Si ha una contraddizione non appena una formula e la sua negazione dovessero risultare entrambe vere.

Esempio (1)

Si consideri l'enunciato $(P_1 \vee P_2) \to \neg(\neg P_1 \wedge \neg P_2)$.
Una dimostrazione intuitiva che esso è una tautologia può essere condotta come segue:
Se $(P_1 \vee P_2) \to \neg(\neg P_1 \wedge \neg P_2)$ *non fosse una tautologia, allora esisterebbe una valutazione booleana* $\mathbf{v}$ *che lo rende falso.*
Quindi:

./.

Esempio (1, cntd)

1: $\neg((P_1 \vee P_2) \rightarrow \neg(\neg P_1 \wedge \neg P_2))$ è *vero*, *per cui*
2: $P_1 \vee P_2$ è *vero e*
3: $\neg\neg(\neg P_1 \wedge \neg P_2)$ è *vero. Da quest'ultimo fatto segue che*
4: $\neg P_1 \wedge \neg P_2$ è *vero, e pertanto*
5: $\neg P_1$ è *vero e*
6: $\neg P_2$ è *vero.*

Dalla verità di $P_1 \vee P_2$ *segue che* P_1 *o* P_2 è *vero.*
Supponiamo, inizialmente, che

7.1: P_1 *sia vero.*
Si ha una contraddizione, poiché abbiamo già che $\neg P_1$ è *vero. Pertanto deve aversi*

7.2: P_2 è *vero.*
Ma anche in questo caso si ha una contraddizione, poiché $\neg P_2$ è *vero.*

Esempio (1, cntd)

Poiché in tutti i casi viene raggiunta una contraddizione, segue che la nostra ipotesi iniziale circa l'esistenza di una valutazione booleana $\mathbf{v}$ *tale che* $\mathbf{v}((P_1 \vee P_2) \rightarrow \neg(\neg P_1 \wedge \neg P_2)) = \mathbf{f}$ *è contraddittoria, e quindi il nostro enunciato è una tautologia.* $\square$

Esempio (2)

Si consideri adesso il tentativo di dimostrare con la stessa tecnica che l'enunciato $(P_1 \vee P_2) \rightarrow \neg(P_1 \wedge \neg P_2)$ *è una tautologia.*

Se $(P_1 \vee P_2) \rightarrow \neg(P_1 \wedge \neg P_2)$ *non fosse una tautologia, allora esisterebbe una valutazione booleana* $\mathbf{v}$ *che lo rende falso. Quindi,*

1′: $\neg((P_1 \vee P_2) \rightarrow \neg(P_1 \wedge \neg P_2))$ è *vero, cosicché*
2′: $P_1 \vee P_2$ è *vero e*
3′: $\neg\neg(P_1 \wedge \neg P_2)$ è *vero. Quest'ultima implica che*
4′: $P_1 \wedge \neg P_2$ è *vero, che a sua volta implica che*
5′: P_1 è *vero e*
6′: $\neg P_2$ è *vero.*

Dalla verità di $P_1 \vee P_2$ *segue che* P_1 *o* P_2 *è vero.*

Esempio (2, cntd)

Per cominciare, supponiamo che

7.1′: P_2 *sia vero.*
Otteniamo una contraddizione poiché sappiamo già che $\neg P_2$ *è vero. Pertanto si ha:*
7.2′: P_1 *è vero.*

Si noti che in questo caso il ramo con l'enunciato 7.2′ *non contiene alcuna contraddizione. I letterali occorrenti in esso sono* P_1 *e* $\neg P_2$*; quindi, se la nostra ipotesi iniziale fosse corretta, si dovrebbe avere* $v(P_1) = \mathbf{t}$ *e* $v(P_2) = \mathbf{f}$*. È facile verificare che ogni valutazione booleana* v^* *tale che* $v^*(P_1) = \mathbf{t}$ *e* $v^*(P_2) = \mathbf{f}$ *rende veri tutti gli enunciati sul ramo in questione e, in particolare, rende falso il nostro enunciato iniziale. Si osservi che non solo abbiamo una dimostrazione che* $(P_1 \vee P_2) \rightarrow \neg(P_1 \wedge \neg P_2)$ *non è una tautologia, ma abbiamo anche ottenuto un controesempio.* $\square$

I due precedenti esempi rappresentano i due tipici risultati della strategia di dimostrazione tratteggiata prima, nel caso di enunciati proposizionali. Infatti, in alcune situazioni favorevoli, ciò è anche vero nel caso di altre teorie decidibili, come si vedrà più tardi.

La definizione di un calcolo nello stile dei tableaux per la logica proposizionale è semplificata dall'osservazione che le regole di inferenza (a)–(e) viste prima sono di due tipi:
le regole (a), (b) e (e) sono di tipo *congiuntivo*, poiché consentono di derivare uno o due enunciati, invece le regole (c) e (d) sono di tipo *disgiuntivo*, poiché causano uno *splitting*.

Pertanto, seguendo la notazione unificante introdotta da Smullyan, è possibile raggruppare le regole in due regole di base: l'α-regola (per gli enunciati di tipo congiuntivo, anche detti α-enunciati) e la β-regola (per gli enunciati di tipo disgiuntivo, anche detti β-enunciati). Tali regole sono solitamente rappresentate come in Figura 2, dove le α- e β-componenti sono definite come in Figura 1.

α	α_1	α_2
$\neg\neg A$	A	$-$
$A \wedge B$	A	B
$\neg(A \vee B)$	$\neg A$	$\neg B$
$\neg(A \rightarrow B)$	A	$\neg B$
$\ldots$	$\ldots$	$\ldots$

β	β_1	β_2
$A \vee B$	A	B
$\neg(A \wedge B)$	$\neg A$	$\neg B$
$A \rightarrow B$	$\neg A$	B
$\ldots$	$\ldots$	$\ldots$

Figura: α- e β-enunciati

$$\frac{\alpha}{\begin{array}{c}\alpha_1 \\ [\alpha_2]\end{array}} \; (\alpha\text{-rule}) \qquad \frac{\beta}{\beta_1 \mid \beta_2} \; (\beta\text{-rule})$$

Figura: Calcolo a tableaux per la logica proposizionale

Definizione (Tableaux Proposizionali)

Dato un enunciato proposizionale C, l'albero di un solo nodo la cui radice è etichettata con C è il *tableau iniziale* per C.
Un *tableau* per C è un albero ottenuto dal tableau iniziale per C mediante un numero finito di applicazioni di una delle seguenti *regole di espansione*:

> *Sia* **T** *un tableau per* C *e sia* ϑ *un ramo di* **T** *contenente almeno un non-letterale tra i suoi enunciati. Sia* A *un non-letterale occorrente in* ϑ*. Allora l'albero ottenuto da* **T** *prolungando* ϑ *con le* α*-componenti di* A*, purché* A *sia un* α*-enunciato, o spezzando* ϑ *con le* β*-componenti di* A*, purché* A *sia un* β*-enunciato, è un tableau per* C*. In tali casi, si dice che all'enunciato* A *sul ramo* ϑ *è stata applicata una regola di espansione per tableau.* □

Osservazione

In modo naturale, la definizione precedente può essere estesa ad ogni collezione finita $\mathbb{F} = \{A_1, \ldots, A_n\}$ *di enunciati, pur di definire come* *tableau iniziale* *per* $\mathbb{F}$ *l'albero costituito dal solo ramo*

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

$\square$

Per definire i teoremi del nostro calcolo a tableau, dobbiamo introdurre la seguente nozione di chiusura.

Definizione (Condizione di chiusura)

Un ramo ϑ di un tableau proposizionale è *chiuso* se esso contiene due enunciati complementari A e $\neg A$, altrimenti è *aperto*. Se ϑ contiene due letterali complementari, allora esso è *chiuso atomicamente*. Un tableau è (atomicamente) chiuso se tutti i suoi rami sono (atomicamente) chiusi. Un tableau o un ramo che non è (atomicamente) chiuso è detto (atomicamente) aperto. □

Definizione (Dimostrazioni e refutazioni nel sistema dei tableaux)

Un enunciato C è un *teorema* del calcolo proposizionale a tableaux se esiste un tableau chiuso per $\neg C$. Ciascun tableau chiuso per $\neg C$ è una *dimostrazione nel sistema dei tableaux* per C.
Una *refutazione nel sistema dei tableaux* per C è un qualunque tableau chiuso per C. □

I tentativi di dimostrazione illustrati nei precedenti esempi possono essere riassunti con i seguenti tableaux, i cui rami chiusi sono contrassegnati con il simbolo $\perp$.

1: $\neg((P_1 \vee P_2) \rightarrow \neg(\neg P_1 \wedge \neg P_2))$

2: $P_1 \vee P_2$

3: $\neg\neg(\neg P_1 \wedge \neg P_2)$

4: $\neg P_1 \wedge \neg P_2$

5: $\neg P_1$

6: $\neg P_2$

7.1: P_1 $\perp$ 7.2: P_2 $\perp$

1′: $\neg((P_1 \vee P_2) \rightarrow \neg(P_1 \wedge \neg P_2))$

2′: $P_1 \vee P_2$

3′: $\neg\neg(P_1 \wedge \neg P_2)$

4′: $P_1 \wedge \neg P_2$

5′: P_1

6′: $\neg P_2$

7.1′: P_1 7.2′: P_2 $\perp$

Correttezza e Completezza

Al fine di stabilire l'adeguatezza del nostro calcolo per caratterizzare tutte e sole le tautologie, occorre dimostrare che esso è sia *corretto* che *completo*, cioè occorre dimostrare che

- ogni teorema del nostro calcolo è una tautologia
- ogni tautologia ha una dimostrazione nel nostro calcolo.

Teorema (Correttezza dei tableaux proposizionali)

Ogni teorema del calcolo dei tableaux proposizionali è una tautologia.

Dimostrazione

Sia C un teorema del calcolo dei tableaux proposizionali e sia $\mathbf{T}$ un tableau chiuso per $\neg C$. Se C non fosse una tautologia, esisterebbe una valutazione booleana v tale che $v(C) = \mathbf{f}$. Per induzione, segue facilmente che $\mathbf{T}$ deve possedere un ramo ϑ i cui enunciati sono veri rispetto alla valutazione v. Ma ciò è impossibile, in quanto per la chiusura di $\mathbf{T}$, il ramo ϑ deve contenere due enunciati complementari. $\square$

Dimostriamo adesso la completezza di una specifica strategia di costruzione di tableaux, più appropriata per l'implementazione. Ovviamente ciò implicherà la completezza nel nostro calcolo.
A tal fine, conviene introdurre la seguente terminologia.

Si dice che un'occorrenza di un enunciato $\boldsymbol{A}$ su un ramo ϑ di un dato tableau $\mathbf{T}$ è *usata* su ϑ, se durante il processo di costruzione di $\mathbf{T}$ è stata applicata una regola di espansione a tale occorrenza di $\boldsymbol{A}$ su un ramo ϑ', che successivamente è stato esteso a ϑ.

Un ramo di tableau proposizionale è *saturo* se tutte le occorrenze dei suoi non-letterali sono stati usati su esso.

Un tableau è (atomicamente) saturo se tutti i suoi rami (atomicamente) aperti sono saturi.

Si consideri adesso la seguente procedura di saturazione :

procedure Saturate(**T**);
  while **T** non è saturo **do**
    - sia ϑ un ramo non saturo e (atomicamente) aperto di **T** e sia ***A*** un'occorrenza di un non-letterale che non è stata usata su ϑ;
    - si applichi l'opportuna regola di espansione ad ***A*** su ϑ;
    - sia **T** il tableau risultante;
  end while;
  return **T**;
end Saturate;

Lemma

La procedura Saturate *termina sempre producendo un tableau saturo.*

La dimostrazione è lasciata per esercizio.

Osservazioni

I seguenti punti sono rilevanti per questioni di ottimizzazione:

- *La procedura* Saturate *può produrre tableaux più ridotti operando in modalità "non atomica". D'altra parte, è più facile stabilire se un ramo sia atomicamente aperto piuttosto che "aperto in generale" ed infatti la modalità "atomica" è quella preferita in pratica.*
- *In genere è sufficiente che la procedura* Saturate *si fermi non appena trova un ramo aperto e saturo.*

./.

Osservazioni, cntd

- *Conviene che le applicazioni della* β*-regola siano ritardate il più possibile, poiché causano delle biforcazioni (non desiderabili dal punto di vista della complessità). A tal proposito, è sufficiente assegnare alla* α*-regola una precedenza più alta della* β*-regola. Questo è un semplice esempio della flessibilità dei calcoli a tableaux, cui si faceva riferimento prima.*
- *Un ben maggior grado di ottimizzazione si raggiunge utilizzando un sistema di regole che limita la ridondanza nei tableaux, come accenneremo più avanti.* $\square$

Per dimostrare che il nostro calcolo è completo, useremo il seguente lemma elementare.

Lemma

Sia ϑ un ramo aperto e saturo in un dato tableau proposizionale e sia $\boldsymbol{v}$ una valutazione booleana rispetto alla quale tutti i letterali su ϑ siano veri. Allora tutti gli enunciati su ϑ sono veri rispetto a $\boldsymbol{v}$.

La dimostrazione è lasciata per esercizio.

Teorema (Completezza dei tableaux proposizionali)

Il calcolo a tableau per la logica proposizionale è completo.

Dimostrazione

Sia D una tautologia nella logica proposizionale e sia $\mathbf{T}$ un tableau saturo per $\neg D$. Per il lemma precedente, tutti i rami di $\mathbf{T}$ debbono essere chiusi, il che implica che $\mathbf{T}$ è una dimostrazione per tableaux per D. $\square$

I risultati precedenti implicano direttamente la correttezza della seguente procedura di decisione per la logica proposizionale.

TableauTautologyTest

procedure TableauTautologyTest(C);
- sia **T** il tableau iniziale per $\neg C$;

T := **Saturate(T)**;
if T è chisuo **then**
return "**T** è una dimostrazione per tableau di C";
else
- sia ϑ un ramo aperto di **T** e sia v una valutazione booleana rispetto alla quale tutti i letterali in ϑ siano veri;

return "v è una valutazione booleana che rende falsa C";
end if;
end TableauTautologyTest;

Esercizio 1

Dimostrare che un ramo saturo in un tableau proposizionale è chiuso se e solo se esso è atomicamente chiuso.

Esercizio 2

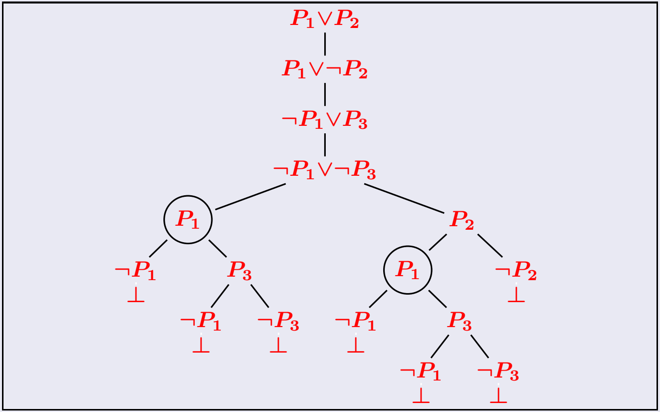
Sia $\mathbf{T}$ *tableau saturo per un enunciato* C. *Sia* $D = \bigvee_{\vartheta \in \mathbf{Open}(\mathbf{T})} A_\vartheta$, *dove* A_ϑ *è la congiunzione di tutti i letterali occorrenti nel ramo* ϑ *e* $\mathbf{Open}(\mathbf{T})$ *è l'insieme di tutti i rami aperti di* $\mathbf{T}$. *Dimostrare che* D *è tautologicamente equivalente a* C, *cioè* D *è una forma normale disgiuntiva di* C, *dove per convenzione la disgiunzione vuota è equivalente a* $\mathbf{f}$.

Outline

- Sebbene il calcolo a tableaux per la logica proposizionale appena illustrato sia molto naturale, in alcuni casi esso produce tableaux con un certo grado di ridondanza.
- Per illustrare questo punto si consideri la seguente collezione di enunciati

$$\mathbb{F} = \{P_1 \vee P_2, P_1 \vee \neg P_2, \neg P_1 \vee P_3, \neg P_1 \vee \neg P_3\}.$$

Un tableau chiuso minimale per $\mathbb{F}$ è il seguente (si noti che i sottoalberi radicati nei nodi cerchiati sono identici).



- Una spiegazione intuitiva di questo fatto è che le regole del calcolo non consentono di effettuare un taglio netto nello spazio di ricerca dei possibili controesempi.
- Ad esempio, quando la prima ramificazione ha luogo, il sottoalbero sinistro si prende carico dei controesempi in cui P_1 è vero. Parimenti, il sottoalbero destro si prende carico dei controesempi in cui P_2 è vero. Pertanto, la dimostrazione che non esiste alcun controesempio che renda veri sia P_1 che P_2 deve essere ripetuta due volte.

Una prima soluzione immediata al precedente problema consiste nel sostituire la β-rule con una delle seguenti varianti (o con entrambe)

$$\frac{\beta}{\beta_1 \;\Big|\; \begin{array}{c}\beta_1^c\\ \beta_2\end{array}} \qquad\qquad \frac{\beta}{\begin{array}{c}\beta_1\\ \beta_2^c\end{array} \;\Big|\; \beta_2}$$

dove β_i^c sta per il complementare di β_i, $i = 1, 2$.

Una soluzione migliore tiene conto anche del fatto che se un ramo contiene un enunciato β e il complementare di una delle sue β-componenti, ad esempio β_i^c, con $i \in \{1, 2\}$, allora dopo un'applicazione della β-regola a β, il ramo contenente β_i si chiude immediatamente.

Il calcolo risultante, che è una variante del calcolo **KE** introdotto da Mondadori, è il seguente:

$$\frac{\beta}{\beta_1 \mid \beta_1^c}(\beta 1) \qquad \frac{\beta}{\beta_2 \mid \beta_2^c}(\beta 2) \qquad \frac{\begin{array}{c}\beta\\ \beta_1^c\end{array}}{\beta_2}(\beta 3) \qquad \frac{\begin{array}{c}\beta\\ \beta_2^c\end{array}}{\beta_1}(\beta 4)$$

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ [\alpha_2]\end{array}}(\alpha)$$

Le sue regole sono di due tipi: *regole lineari* (α, $\beta 3$, $\beta 4$) e *regole di taglio* ($\beta 1$, $\beta 2$).
Si dimostra che il calcolo **KE** è corretto e completo.

Alcune definizioni

- Un ramo ϑ di un **KE**-tableau è *linearmente saturo* se non può essere aggiunto ad esso alcun nuovo enunciato prodotto da un'applicazione su ϑ di una regola lineare.
- Parimenti, un **KE**-tableau è *linearmente saturo* se tutti i suoi rami sono linearmente saturi.
- Un enunciato β è *trattato* su un ramo ϑ se almeno una delle sue componenti β_1 e β_2 occorre in ϑ.
- Un ramo ϑ è *trattato* se tutti i suoi β-enunciati sono trattati su esso.
- un **KE**-tableau è *trattato* se tutti i suoi rami sono trattati.

Si consideri la seguente strategia di saturazione:

Procedura di saturazione

procedure KE-Saturate(**T**);
  repeat
    - si saturi linearmente **T**;
    if T ha un ramo ϑ non trattato **do**
      - si selezioni su ϑ un enunciato non trattato β;
      - si applichi la β**1**- o la β**2**-regola a β su ϑ;
    end if;
  until **T** è chiuso oppure è linearmente saturo e trattato;
  return **T**;
end KE-Saturate;

È facile verificare che

- la procedura KE-Saturate termina sempre;
- la procedura KE-Saturate è completa, nel senso che se **T** è insoddisfacibile, allora il tableau costruito dall'esecuzione di KE-Saturate(T) è chiuso.

Esercizio

Si dimostri quanto affermato sopra.

Come ci si aspetta, il calcolo **KE** produce tableaux più piccoli del calcolo visto precedentemente. Per esempio, la figura seguente mostra un **KE**-tableau chiuso per l'insieme di enunciati

$$\{P_1 \vee P_2,\ P_1 \vee \neg P_2,\ \neg P_1 \vee P_3,\ \neg P_1 \vee \neg P_3\}$$

costruito dalla procedura KE-Saturate.