7 Alberi di refutazione

7.1 Il metodo

Un altro metodo per rispondere alle domande sulla verità logica o sulla insoddisfacibilità delle proposizioni, e che risulta più efficiente (almeno sui casi di dimensione ridotta) della ricerca esaustiva offerta dalla costruzione delle tavole di verità, è il metodo degli *alberi di refutazione*[1].

Il nome deriva dal fatto che il procedimento è impostato, per rispondere alla domanda sulla verità logica, secondo la *ricerca del controesempio*: si cerca di scoprire se esiste un'interpretazione che falsifichi la proposizione. Il metodo ha la proprietà che o la trova, se esiste, e quindi fornisce un'interpretazione in cui la negazione della proposizione è vera (controesempio: la proposizione è falsa) oppure mostra che non è possibile che esista, e quindi la proposizione è una tautologia.

Più in generale, il metodo serve a stabilire se esista o no un'interpretazione che soddisfa una proposizione composta, non partendo dal basso dalle possibili interpretazioni delle lettere (*bottom up*) ma dall'alto, dalla proposizione data, scendendo verso le sottoproposizioni componenti (*top down*); nel processo, si accumulano condizioni necessarie che l'ipotetica interpretazione, se esiste e soddisfa la radice, dovrebbe pure soddisfare – nel senso di quali altre proposizioni essa dovrebbe soddisfare o no – fino alle condizioni necessarie riguardanti le proposizioni atomiche; queste, se non sono incompatibili tra di loro, si traducono in condizioni sufficienti per la definizione dell'interpretazione.

Gli alberi di refutazione possono dunque essere usati anche per rispondere alle altre domande semantiche, ad esempio quella sulla soddisfacibilità.

Gli alberi di refutazione sono alberi etichettati con proposizioni. Identifichiamo per comodità di scrittura i nodi con le loro etichette. Nella radice è una proposizione, di cui si vuole sapere se esiste un modello. L'albero è sviluppato secondo la seguente procedura.

Ad ogni stadio, si saranno già prese in considerazione alcune proposizioni, messe tra parentesi quadre o segnate con un asterisco, e ne resteranno da considerare altre. Se sono già state considerate tutte, l'albero è *terminato*; se no, si prende in esame una proposizione A non ancora considerata, e a seconda della sua forma si prolunga l'albero nel modo seguente, dopo aver

[1]Altri nomi usati, insieme a qualche variante di presentazione, sono quelli di *alberi semantici*, oppure di *tableaux* semantici.

segnato A e aver notato quali sono i rami non chiusi che passano per A, dove un ramo si dice *chiuso* se su di esso occorre sia una proposizione sia la sua negazione:

- Se A è una proposizione senza connettivi, non si fa nulla (si va al passo successivo).

- Se A è $B \wedge C$, alla fine di ogni ramo non chiuso passante per A si appendono alla foglia due nodi in serie etichettati con B e C, come nello schema:

$$\begin{array}{c} [B \wedge C] \\ \vdots \\ \downarrow \\ F \\ \downarrow \\ B \\ \downarrow \\ C \end{array}$$

- Se A è $B \vee C$, alla fine di ogni ramo non chiuso passante per A si aggiunge alla foglia una diramazione con due nodi B e C, come nello schema:

$$\begin{array}{ccc} & [B \vee C] & \\ & \vdots & \\ & \downarrow & \\ & F & \\ \swarrow & & \searrow \\ B & & C \end{array}$$

  con l'ovvia generalizzazione (qui e nella precedente regola) che se si tratta di congiunzioni o disgiunzioni generalizzate si appendono in serie o rispettivamente si fanno diramazioni con tanti nodi quante sono le sottoproposizioni immediate.

- Se A è $B \to C$, alla fine di ogni ramo non chiuso passante per A si aggiunge alla foglia una diramazione con due nodi $\neg B$ e C, come nello schema:

$$\begin{array}{ccc} & [B \to C] & \\ & \vdots & \\ & \downarrow & \\ & F & \\ \swarrow & & \searrow \\ \neg B & & C \end{array}$$

- Se A è $\neg B$ e B non ha connettivi, non si fa nulla.

- Se A è della forma $\neg B$ e B è $\neg C$, al fondo di ogni ramo non chiuso passante per A si appende alla foglia il successore C, come nello schema:

$$\begin{array}{c} [\neg\neg C] \\ \vdots \\ \downarrow \\ F \\ \downarrow \\ C \end{array}$$

- Se A è della forma $\neg B$ e B è $B_1 \vee B_2$, alla fine di ogni ramo non chiuso passante per A si aggiungono alla foglia due nodi in serie $\neg B_1$ e $\neg B_2$, come nello schema:

$$\begin{array}{c} [\neg(B_1 \vee B_2)] \\ \vdots \\ \downarrow \\ F \\ \downarrow \\ \neg B_1 \\ \downarrow \\ \neg B_2 \end{array}$$

  con l'ovvia generalizzazione se B è una disgiunzione generalizzata.

- Se A è della forma $\neg B$ e B è $B_1 \to B_2$, alla fine di ogni ramo non chiuso passante per A si appendono alla foglia due successori in serie

B_1 e $\neg B_2$, come nello schema:

$$
\begin{array}{c}
[\neg(B_1 \to B_2)] \\
\vdots \\
\downarrow \\
F \\
\downarrow \\
B_1 \\
\downarrow \\
\neg B_2
\end{array}
$$

- Se A è della forma $\neg B$ e B è $B_1 \wedge B_2$, alla fine di ogni ramo non chiuso passante per A si aggiunge alla foglia una diramazione con due nodi $\neg B_1$ e $\neg B_2$, come nello schema:

$$
\begin{array}{ccc}
 & [\neg(B_1 \wedge B_2)] & \\
 & \vdots & \\
 & \downarrow & \\
 & F & \\
\swarrow & & \searrow \\
\neg B_1 & & \neg B_2
\end{array}
$$

Ovviamente se per il nodo in considerazione non passa alcun ramo non chiuso, non si fa nulla. Dalla formulazione è chiaro che quando tutti i rami sono chiusi il procedimento termina, anche se non tutte le proposizioni sono state considerate, e in tal caso l'albero si considera terminato e si dice *chiuso*.

Non diamo le regole per il bicondizionale (esercizio) perché non sarebbero altro che l'adattamento di quelle che derivano dal fatto che $p \leftrightarrow q$ è equivalente a $(p \to q) \wedge (q \to p)$. Lo stesso per $\oplus$, ma si preferisce eliminare prima questi connettivi (comunque, si diano le regole per $\oplus$ per esercizio), e questa è l'unica preparazione o trasformazione che si fa sulle proposizioni; altrimenti si prendono così come sono, e questo è un vantaggio del metodo, **!!!** nessun *pre-processing*.

Si leggano con attenzione le regole, cogliendone tutte le informazioni e i vincoli: ad esempio, quando si lavora su di un nodo, si aggiungono proposizioni su *tutti* i rami *passanti* per quel nodo, ma *non* sugli altri. **!!!**

Esempio

1. Consideriamo la proposizione $\neg((\neg p \vee q) \wedge p \to q)$ che mettiamo nella radice dell'albero

$$\neg((\neg p \vee q) \wedge p \to q)$$

2. Lavorando su di esso, che è la negazione di un condizionale, otteniamo

$$\begin{array}{c} [\neg((\neg p \vee q) \wedge p \to q)] \\ \downarrow \\ (\neg p \vee q) \wedge p \\ \downarrow \\ \neg q \end{array}$$

3. Lavorando su $(\neg p \vee q) \wedge p$ otteniamo

$$\begin{array}{c} [\neg((\neg p \vee q) \wedge p \to q)] \\ \downarrow \\ [(\neg p \vee q) \wedge p] \\ \downarrow \\ \neg q \\ \downarrow \\ \neg p \vee q \\ \downarrow \\ p \end{array}$$

4. Lavorando prima su $\neg q$, senza alcun effetto, e poi su $\neg p \vee q$

$$\begin{array}{c} [\neg((\neg p \vee q) \wedge p \to q)] \\ \downarrow \\ [(\neg p \vee q) \wedge p] \\ \downarrow \\ [\neg q] \\ \downarrow \\ [\neg p \vee q] \\ \downarrow \\ p \\ \swarrow \quad \searrow \\ \neg p \qquad q\,. \end{array}$$

Non è neanche necessario indicare che si sono presi in considerazione le restanti proposizioni, perché il loro effetto è nullo. L'albero è chiuso, perché su uno dei sue due rami occorrono p e $\neg p$, e sull'altro occorrono q e $\neg q$.

Se si deve interpretare come è stato ottenuto un albero sviluppato, è di aiuto che sia segnato a fianco di ogni proposizione l'ordine in cui è stata presa in considerazione, come in

$$
\begin{array}{c}
[\neg((\neg p \lor q) \land p \to q)]_1 \\
\downarrow \\
[(\neg p \lor q) \land p]_2 \\
\downarrow \\
[\neg q]_3 \\
\downarrow \\
[\neg p \lor q]_4 \\
\downarrow \\
p \\
\swarrow \quad \searrow \\
\neg p \qquad q\,.
\end{array}
$$

Esempio

$$
\begin{array}{c}
\neg((p \land q) \lor (\neg p \land q) \lor \neg q)_1 \\
\downarrow \\
\neg(p \land q)_3 \\
\downarrow \\
\neg(\neg p \land q)_4 \\
\downarrow \\
\neg\neg q_2 \\
\downarrow \\
q \\
\swarrow \quad \searrow \\
\begin{array}{cc}
\begin{array}{c}
\neg p \\
\swarrow \quad \searrow \\
\begin{array}{cc} \begin{array}{c}\neg\neg p \\ \text{chiuso}\end{array} & \neg q \end{array}
\end{array}
&
\begin{array}{c}
\neg q \\
\text{chiuso}
\end{array}
\end{array}
\end{array}
$$

dove il ramo di destra con foglia $\neg q$ non è sviluppato con

$$
\begin{array}{ccc}
 & \neg q & \\
\swarrow & & \searrow \\
\neg\neg p & & \neg q
\end{array}
$$

come dovrebbe essere per il lavoro su $\neg(\neg p \wedge q)$, perché il ramo è già chiuso; il ramo di sinistra non è prolungato con

$$
\begin{array}{c}
\neg\neg p \\
\downarrow \\
p
\end{array}
$$

perché anch'esso chiuso.

Avvertenza Non si confondano gli alberi di refutazione con gli alberi sintattici. L'albero sintattico di una proposizione contiene solo le sottoproposizioni di quella data, l'albero di refutazione anche altre. Ad esempio

$$
\begin{array}{ccc}
 & p \vee q \to p \wedge q & \\
 & \swarrow \quad \searrow & \\
p \vee q & & p \wedge q \\
\swarrow \quad \searrow & & \swarrow \quad \searrow \\
p \qquad q & & p \qquad q
\end{array}
$$

è l'albero sintattico di $p \vee q \to p \wedge q$. Il suo albero di refutazione è invece

$$
\begin{array}{ccc}
 & p \vee q \to p \wedge q & \\
 & \swarrow \quad \searrow & \\
\neg(p \vee q) & & p \wedge q \\
\downarrow & & \downarrow \\
\neg p & & p \\
\downarrow & & \downarrow \\
\neg q & & q
\end{array}
$$

dove, oltre a una struttura diversa, compaiono proposizioni come $\neg p$, $\neg q$, $\neg(p \vee q)$.

In alcune presentazioni gli alberi di refutazione sono alberi i cui nodi sono proposizioni etichettate con i simboli V ed F, per "vero" e "falso". La corrispondenza con la presente versione si ottiene sostituendo $V\,A$ con A e $F\,A$ con $\neg A$, e viceversa.

Ad esempio le due regole per la congiunzione diventano:

$V\wedge$: Quando si lavora su $V(A \wedge B)$ (al fondo di ogni ramo …) si appendono in serie $V\,A$ e $V\,B$.

$F\wedge$: Quando si lavora su A $F(A \wedge B)$ (al fondo di ogni ramo …) si appende una diramazione con con $F\,A$ e $F\,B$.

7.2 Correttezza e completezza

Il primo problema con ogni algoritmo è quello della terminazione, in particolare per gli algoritmi di decisione; se l'algoritmo non si ferma sempre, con una risposta, dopo un numero finito di passi, non ci si può affidare ad esso per decidere le questioni che interessano (nel senso di lanciarlo e stare ad aspettare).

Lemma 7.2.1 (Terminazione)**.** *La costruzione dell'albero di refutazione inizializzato con una proposizione termina sempre in un numero finito di passi.*

Dimostrazione. Se ad ogni stadio si lavora su una proposizione di quelle che hanno altezza massima n tra quelle non ancora considerate, l'applicazione delle regole fa sì che dopo un numero finito di passi tutte quelle di altezza n siano state considerate, e l'altezza massima delle proposizioni non ancora considerate sia quindi $< n$. Infatti le proposizioni introdotte nell'albero con le regole hanno tutte altezza minore della proposizione che governa la regola, salvo il caso di $B \to C$, per cui si introducono $\neg B$ e C, e $\neg B$ può avere la stessa altezza di $B \to C$ (quando? esercizio); ma la successiva applicazione di una delle regole per proposizioni negate a $\neg B$, che si può eseguire subito, la sostituisce con proposizioni di altezza minore.

Anche se dunque nel corso del procedimento il numero di proposizioni nei nodi dell'albero cresce con il crescere dell'albero, diminuisce quello delle proposizioni di altezza massima, e dopo un numero finito di passi ci saranno solo proposizioni di altezza minima, senza connettivi, non ancora considerate, e a quel punto il processo termina, se non è terminato prima per la chiusura dell'albero. □

Per quel che riguarda la correttezza e la completezza del metodo, qualche perplessità potrebbe sorgere dal fatto che non abbiamo introdotto un calcolo.

Abbiamo descritto tuttavia un algoritmo, che è basato solo sulla struttura sintattica, anche se infiorettato di terminologia semantica. Le risposte si evincono dalla struttura dell'albero (chiuso, non chiuso).

Piuttosto qualche ambiguità potrebbe sussistere in quanto le domande possibili sono diverse, ancorché collegate. Per il fatto che

Osservazione Per ogni A,

$$A \text{ è una tautologia se e solo se } \neg A \text{ è insoddisfacibile,}$$

ci si può porre come problema semantico sia il problema della verità logica sia il problema dell'insoddisfacibilità. Un calcolo si può pensare sia come calcolo per stabilire la verità logica sia come un calcolo per stabilire l'insoddisfacibilità. Scegliamo il metodo degli alberi di refutazione per il problema dell'insoddisfacibilità, e come risposta preferenziale affermativa la chiusura dell'albero (un esito in generale più rapido e che non richiede ulteriori elaborazioni); abbiamo allora

Teorema 7.2.2 (Correttezza). *Se l'albero di refutazione con radice A si chiude, allora A è insoddisfacibile.*

Dimostrazione[2]. Procediamo per contrapposizione dimostrando che se esiste un'interpretazione i che soddisfa A, allora a ogni stadio di sviluppo dell'albero esiste almeno un ramo tale che i soddisfa tutti le proposizioni del ramo. Allora l'albero non è mai chiuso, perché se un ramo è chiuso non tutte le sue proposizioni possono essere vere in una stessa interpretazione.

Allo stadio n, consideriamo un ramo σ le cui proposizioni siano tutte soddisfatte da i, e una proposizione B su di esso, quindi vera in i, e non ancora considerata (se non ce ne sono, il lavoro su quel ramo è terminato senza che esso sia chiuso, e tale rimane alla fine, e l'albero finale non è chiuso). Se B è una congiunzione, al ramo sono aggiunti due nodi che sono anch'essi etichettati con proposizioni vere in i, e il ramo prolungato soddisfa, allo stadio successivo, la proprietà richiesta. Se B è una disgiunzione $B_1 \vee B_2$, o il ramo[3] $\sigma^\frown B_1$ o il ramo $\sigma^\frown B_2$ soddisfano la proprietà richiesta, a seconda che B_1 o B_2 siano vere in i. Lo stesso vale per gli altri casi (esercizio). □

Viceversa

[2]Per questo e per il successivo teorema diamo dimostrazioni complete, impostate per induzione.

[3]$\sigma^\frown B_1$ è il ramo prolungato con B_1; la notazione è quella della concatenazione di liste.

Teorema 7.2.3 (Completezza)**.** *Se A è insoddisfacibile, l'albero di refutazione con radice p si chiude.*

Dimostrazione. Dimostriamo che

Lemma 7.2.4. *Se l'albero non si chiude, allora per ogni ramo non chiuso e terminato esiste un'interpretazione i che soddisfa tutti le proposizioni del ramo, inclusa la radice.*

Dimostrazione del lemma. Sia σ un ramo non chiuso dell'albero terminato. Si definisca un'interpretazione i ponendo $i(p) = 1$ per ogni proposizione atomica p che occorre come nodo nel ramo σ, e $i(p) = 0$ per ogni proposizione atomica tale che $\neg p$ occorre come nodo nel ramo σ. Si dimostra ora che ogni proposizione di σ è vera in i. Supponiamo questo verificato per tutte le proposizioni sul ramo che hanno un'altezza minore di un numero fissato n, e facciamo vedere che lo stesso vale per quelle di altezza n. Se B è una congiunzione $B_1 \wedge B_2$, quando è stata presa in considerazione B si sono aggiunti come nodi del ramo sia B_1 che B_2, che sono quindi in σ e hanno altezza minore di n e quindi si suppongono vere in i; dunque anche B è vera in i. Se B è una disgiunzione $B_1 \vee B_2$, quando è stata presa in considerazione B si sono aggiunti a tutti i rami passanti per B, incluso (quello che sarebbe diventato) σ, o B_1 o B_2; quindi una delle due è su σ, e vera in i, quindi anche B è vera. Gli altri casi si trattano nello stesso modo. □ □

Se in un ramo terminato non chiuso manca una lettera che occorre nella radice, nel definire l'interpretazione si può dare ad essa il valore che si vuole; ciò significa che al ramo è associata più di una interpretazione.

L'esito complessivo dei teoremi di correttezza e completezza è che il metodo degli alberi prende in esame *tutte* le possibili strade per provare a definire interpretazioni, e se ce ne sono le fornisce tutte, e se non ce ne sono lo rivela.

La dimostrazione delle proprietà di correttezza e completezza non prende in considerazione l'ordine in cui si sviluppa l'albero. Il procedimento degli alberi di refutazione si può rendere deterministico fissando un ordine progressivo per le proposizioni introdotte e quelle da prendere in considerazione ma proprio il fatto che la dimostrazione è indipendente dall'ordine permette di vedere che la risposta dell'albero e le sue proprietà non dipendono dall'ordine eventualmente fissato; lavorare su una proposizione prima che su di un'altra può modificare l'albero ma non la risposta finale; ogni mossa dipende solo dalla proposizione in considerazione e non dalle altre presenti in altri nodi.

Si può sfruttare questa circostanza (oltre che come si è fatto nella dimostrazione della terminazione) per formulare utili regole euristiche, come quella di prendere in esame prima le proposizioni che si limitano ad allungare i rami e non introducono diramazioni.

Riassumendo

Corollario 7.2.5. *Per ogni A,*
A è soddisfacibile se e solo se l'albero di refutazione con radice A non si chiude

mentre, nello spirito del controesempio,

Corollario 7.2.6. *Per ogni A,*
A è una tautologia se e solo se l'albero di refutazione con radice $\neg A$ si chiude.

Per la nozione di conseguenza logica, serve infine il

Corollario 7.2.7. *Per ogni A e B,*
$\models A \to B$ se e solo se l'albero di refutazione con radice $\neg(A \to B)$, o con radice $A \land \neg B$, si chiude.

Si noti che è indifferente avere nella radice $\neg(A \to B)$ oppure l'equivalente $A \land \neg B$ perché in entrambi i casi l'applicazione delle regole per la negazione di un condizionale o per la congiunzione portano ad aggiungere alla radice

$$\begin{array}{c} \downarrow \\ A \\ \downarrow \\ \neg B \end{array}$$

dopo di che si continua lavorando solo su A e su $\neg B$ e loro sottoproposizioni. Si può addirittura partire con

$$\begin{array}{c} A \\ \downarrow \\ \neg B \end{array}$$

se interessa la domanda $A \models B$.

Tuttavia, salvo questa eccezione iniziale, il metodo non prevede, e così le prove di esame, che nel corso della costruzione dell'albero una proposizione venga rimpiazzata da una logicamente equivalente; nei nodi dell'albero devono comparire solo sottoproposizioni o negazioni di sottoproposizioni della radice.

7.3 Forme normali

Gli alberi di refutazione permettono di ottenere altre informazioni sulle proposizioni a cui si applicano. Se A è una proposizione soddisfacibile, e quindi l'albero di refutazione con radice A non si chiude, una forma normale disgiuntiva di A si può ottenere nel seguente modo: per ogni ramo terminato e non chiuso, si faccia la congiunzione di tutti i letterali che sono nodi del ramo, quindi si faccia la disgiunzione di queste congiunzioni. Le proprietà dimostrate della correttezza e della completezza garantiscono che questa disgiunzione è proprio equivalente a A (esercizio).

Esempio

$$
\begin{array}{ccc}
 & \neg(p \lor \neg q) \lor q \lor \neg(p \to q) & \\
\swarrow & \downarrow & \searrow \\
\neg(p \lor \neg q) & q & \neg(p \to q) \\
\downarrow & & \downarrow \\
\neg p & & p \\
\downarrow & & \downarrow \\
\neg\neg q & & \neg q \\
\downarrow & & \\
q\ . & &
\end{array}
$$

L'albero non è chiuso e la forma normale disgiuntiva della radice è $(\neg p \land q) \lor q \lor (p \land \neg q)$; i tre modelli dati dai tre rami non chiusi sono

$$
\begin{array}{l}
i_1(p) = 0, \quad i_1(q) = 1, \\
i_2(q) = 1, \\
i_3(p) = 1, \quad i_3(q) = 0
\end{array}
$$

dove il secondo sta per due interpretazioni, di cui una però coincide con la prima; rami diversi non danno necessariamente interpretazioni diverse. La proposizione non è una tautologia in quanto manca l'interpretazione $i(p) = i(q) = 0$ tra i suoi modelli.

Se l'albero per A si chiude, si sa che A è una contraddizione e una forma normale disgiuntiva si scrive direttamente.

Dall'albero di A non si legge invece direttamente la forma normale congiuntiva di A; per ottenere questa, una via indiretta ma comunque veloce è la seguente: si mette nella radice $\neg A$, si sviluppa l'albero per $\neg A$ e si trova una forma normale disgiuntiva di $\neg A$. Quindi si nega questa premettendo una negazione, e si applicano le leggi di De Morgan.

Poiché l'albero terminato e non chiuso permette di leggere i modelli della radice, per verificare che A è una tautologia si può anche sviluppare l'albero con radice A, e controllare che ci siano alla fine 2^n interpretazioni associate ai rami non chiusi, se A ha n lettere. Ma se la domanda è se A sia una tautologia, è più conveniente impostare l'albero con $\neg A$, perché se la risposta è positiva essa arriva dalla chiusura dell'albero, in generale più in fretta dello sviluppo integrale dell'albero con radice A e del conteggio dei modelli.

7.4 Esercizi

1. Verificare con gli alberi di refutazione le leggi logiche del paragrafo 3.2.3.

2. Verificare con gli alberi di refutazione se le seguenti proposizioni sono tautologie, e se no indicare i controesempi:

   $(p \vee q) \wedge (r \to \neg p) \to (r \to q)$

   $((p \to \neg p) \wedge (q \to p)) \to \neg q$

   $(p \wedge q) \vee (\neg p \wedge q) \vee q$

   $(p \wedge q) \vee (\neg p \wedge q) \vee \neg q$.

3. Verificare con gli alberi di refutazione se le seguenti proposizioni sono insoddisfacibili:

   $((p \vee q) \wedge (\neg p \vee q) \wedge \neg q) \to q$

   $(p \to \neg q) \wedge (\neg p \vee q)$

   $(p \to \neg q) \wedge \neg p \wedge q$

   $(p \to \neg q) \wedge p \wedge q$

   $(p \vee \neg q \vee r) \wedge \neg r \wedge (\neg p \vee q) \wedge \neg p$.

4. Trovare con gli alberi di refutazione la forma normale disgiuntiva e i modelli delle seguenti proposizioni:

   $p \wedge q \to (p \to q)$

   $p \wedge q \to (p \to q \wedge r)$

   $(p \to (q \vee (p \wedge r))) \wedge (\neg p \wedge (q \to p))$.

5. Con gli alberi di refutazione trovare la forma normale congiuntiva delle seguenti proposizioni:

   $p \wedge q \to (p \to q \wedge r)$

   $(p \vee q \to r) \wedge \neg p \to (p \vee r)$

   $(p \to (q \vee (p \wedge r))) \wedge (\neg p \wedge (q \to p))$.