Sintassi
4.1 – Sintassi della logica proposizionale

Docenti: Alessandro Andretta e Luca Motto Ros

Dipartimento di Matematica
Università di Torino

Sintassi

Fissiamo un insieme L non vuoto i cui elementi si dicono **lettere proposizionali**. Le lettere proposizionali sono indicate dalle prime lettere dell'alfabeto $\mathrm{A}, \mathrm{B}, \mathrm{C}, \ldots$ (eventualmente con pedici e apici, come $\mathrm{A}_0, \mathrm{A}_1, \ldots$ o $\mathrm{A}', \mathrm{A}'', \ldots$). L'insieme $\mathrm{Prop}(L)$ delle **proposizioni** (o **formule proposizionali**) su L è un sottoinsieme di

$$(L \cup \{(,), \neg, \vee, \wedge, \to, \leftrightarrow\})^*,$$

l'insieme di tutte le stringhe finite di simboli che sono elementi di L oppure connettivi o parentesi. $\mathrm{Prop}(L)$ è definito dalle clausole

- Se $\mathrm{A} \in L$ allora $(\mathrm{A}) \in \mathrm{Prop}(L)$.
- Se $\mathrm{P} \in \mathrm{Prop}(L)$ allora $(\neg \mathrm{P}) \in \mathrm{Prop}(L)$.
- Se $\square$ è un connettivo binario (ovvero $\square \in \{\vee, \wedge, \to, \leftrightarrow\}$), e se $\mathrm{P}, \mathrm{Q} \in \mathrm{Prop}(L)$ allora $(\mathrm{P} \square \mathrm{Q}) \in \mathrm{Prop}(L)$.

Le clausole della definizione sono anche regole di costruzione: s'intende che ogni proposizione si ottiene applicando un numero finito di volte le clausole della definizione.

Formalmente

Definizione

Per $n \in \mathbb{N}$ definiamo $\mathrm{Prop}_n(L)$ un sottoinsieme di $(L \cup \{(,), \neg, \vee, \wedge, \to, \leftrightarrow\})^*$ per ricorsione mediante le clausole

$$\begin{aligned} \mathrm{Prop}_0(L) &= \{(\mathrm{A}) \mid \mathrm{A} \in L\} \\ \mathrm{Prop}_{n+1}(L) &= \mathrm{Prop}_n(L) \cup \{(\neg \mathrm{P}) \mid \mathrm{P} \in \mathrm{Prop}_n(L)\} \cup \\ &\quad \cup \{(\mathrm{P} \,\square\, \mathrm{Q}) \mid \mathrm{P}, \mathrm{Q} \in \mathrm{Prop}_n(L),\ \square \in \{\vee, \wedge, \to, \leftrightarrow\}\}\,. \end{aligned}$$

Quindi $\mathrm{Prop}_0(L) \subseteq \mathrm{Prop}_1(L) \subseteq \ldots$ e

$$\mathrm{Prop}(L) = \bigcup_{n \in \mathbb{N}} \mathrm{Prop}_n(L)$$

Le lettere $\mathrm{P}, \mathrm{Q}, \mathrm{R}, \ldots$ (eventualmente con pedici e apici) variano su elementi di $\mathrm{Prop}(L)$.

Le proposizioni della forma (A) (per qualche $A \in L$) si dicono **proposizioni atomiche**.

Se una proposizione è invece della forma $(\neg \mathrm{P})$ o della forma $(\mathrm{P} \square \mathrm{Q})$, $\neg$ e $\square$ sono rispettivamente il suo **connettivo principale**, e P e Q le **sottoproposizioni immediate**.

Una proposizione non atomica P viene detta **negazione**, **congiunzione**, **disgiunzione**, **implicazione** oppure **bi-implicazione** quando il suo connettivo principale è $\neg$, $\wedge$, $\vee$, $\rightarrow$ o $\leftrightarrow$, rispettivamente.

La **lunghezza** di una proposizione P è la lunghezza di P vista come stringa,

$$\mathrm{lh}\colon (L \cup \{(,), \neg, \vee, \wedge, \to, \leftrightarrow\})^* \to \mathbb{N}$$

mentre l'**altezza** di una proposizione $\mathrm{ht}(\mathrm{P})$ è definita da

$$\mathrm{ht}\colon \mathrm{Prop}(L) \to \mathbb{N}, \qquad \mathrm{ht}(\mathrm{P}) = \min\{n \in \mathbb{N} \mid \mathrm{P} \in \mathrm{Prop}_n(L)\},$$

Per esempio se $\mathrm{P} = (\mathrm{A})$, allora $\mathrm{lh}(\mathrm{P}) = 3$ e $\mathrm{ht}(\mathrm{P}) = 0$, mentre se $\mathrm{P} = ((\neg(\mathrm{A})) \wedge ((\mathrm{B}) \to (\neg(\mathrm{A}))))$, allora $\mathrm{lh}(\mathrm{P}) = 21$ e $\mathrm{ht}(\mathrm{P}) = 3$.

La lunghezza e l'altezza di una proposizione si dicono **misure di complessità** e ci permettono di dimostrare fatti sulle proposizioni per induzione strutturale sull'insieme di stringhe $(L \cup \{(,), \neg, \vee, \wedge, \to, \leftrightarrow\})^*$ (usando lh) oppure sull'insieme $\mathrm{Prop}(L)$ delle proposizioni (usando ht).

Proposizione

Per ogni $\mathrm{P} \in \mathrm{Prop}(L)$

- P *inizia con una parentesi sinistra "(", termina con una parentesi destra ")",*
- *ogni parentesi sinistra ha una sua parentesi di chiusura destra,*
- $\mathrm{lh}(\mathrm{P})$ *è divisibile per* 3.

Dimostrazione. (Per induzione strutturale sull'altezza di P)

Se $\mathrm{P} \in \mathrm{Prop}_0(L)$ (ossia $\mathrm{ht}(\mathrm{P}) = 0$), allora $\mathrm{P} = (\mathrm{A})$ per qualche $\mathrm{A} \in L$ e il risultato segue immediatamente.

Supponiamo ora che il risultato valga per ogni proposizione in $\mathrm{Prop}_n(L)$ (ossia ogni P tale che $\mathrm{ht}(\mathrm{P}) = n$) e dimostriamo che allora vale anche per ogni proposizione in $\mathrm{Prop}_{n+1}(L)$ (ossia ogni P tale che $\mathrm{ht}(\mathrm{P}) = n + 1$). Fissiamo dunque una generica $\mathrm{P} \in \mathrm{Prop}_{n+1}(L)$.

- Se $\mathrm{P} \in \mathrm{Prop}_n(L)$ il risultato segue immediatamente dall'ipotesi induttiva.

(continua)

Dimostrazione.

- Se $\mathrm{P} = (\mathrm{Q} \square \mathrm{R})$ con $\mathrm{Q}, \mathrm{R} \in \mathrm{Prop}_n(L)$, allora chiaramente P inizia con una parentesi sinistra " (" e termina con una parentesi destra ")". Inoltre, per ipotesi induttiva ogni parentesi sinistra che compare in Q ha la sua parentesi di chiusura destra in Q stesso, e la stessa cosa vale per R; poiché l'unica parentesi sinistra di P che compare fuori da Q e R è quella iniziale, la cui parentesi destra di chiusura è la parentesi finale, si ha che il medesimo risultato vale per P. Infine, se $\mathrm{lh}(\mathrm{Q}) = 3i$ e $\mathrm{lh}(R) = 3j$, allora $\mathrm{lh}(\mathrm{P}) = 1 + 3i + 1 + 3j + 1 = 3(i + j + 1)$.
- Se $\mathrm{P} = (\neg \mathrm{Q})$ con $\mathrm{Q} \in \mathrm{Prop}_n(L)$, allora chiaramente P inizia con una parentesi sinistra " (" e termina con una parentesi destra ")". Per ipotesi induttiva, ogni parentesi sinistra che compare in Q ha la sua parentesi destra di chiusura in Q stesso; poiché l'unica parentesi sinistra di P che compare fuori da Q è quella iniziale, la cui parentesi destra di chiusura è la parentesi finale, si ha che il medesimo risultato vale per P. Infine, se $\mathrm{lh}(\mathrm{Q}) = 3i$, allora $\mathrm{lh}(\mathrm{P}) = 1 + 1 + 3i + 1 = 3(i + 1)$. □

Esempi

Siano $\mathrm{A}, \mathrm{B} \in L$.

1. $\mathrm{A} \wedge \mathrm{B}$ è una proposizione? No, perché ogni proposizione contiene almeno una parentesi.
2. $)\mathrm{A}($ è una proposizione? No, come non lo sono A o $)\mathrm{A})$, perché ogni proposizione inizia con una parentesi $($ e termina con una parentesi $)$.
3. $((\mathrm{A}) \to (\mathrm{B}))$ è una proposizione? Sì, perché ottenuta dalle (A) e (B) con una applicazione della clausola induttiva relativa a $\to$.
4. $(\neg((\mathrm{A}) \to (\mathrm{B})))$ è una proposizione? Sì, perché ottenuta da (A) e (B) con una prima applicazione della clausola induttiva relativa a $\to$ e una seconda applicazione della clausola relativa a $\neg$.
5. $((\mathrm{A})$ è una proposizione? No, perché c'è una parentesi sinistra che non ha la sua parentesi destra di chiusura.
6. (AB) è una proposizione? No, perché non è atomica e non contiene nessun connettivo.

Una proposizione è una lista di simboli, ma è anche passibile di una rappresentazione con una diversa struttura. A ogni proposizione è associato un **albero di costruzione**, o **albero sintattico**, che è un albero etichettato finito binario.

Un **albero finito** è un insieme X parzialmente ordinato, cioè con una relazione binaria $\preceq$ definita su X, con le seguenti proprietà:

- $\preceq$ è una relazione di ordine su X tale che per ogni $x \in X$ l'insieme $\{y \in X \mid y \preceq x\}$ dei predecessori di x è finito e linearmente ordinato da $\preceq$. Gli elementi di X vengono detti **nodi**. Se $x \preceq y$, si dice che y è un **successore**, o un **discendente** di x. I nodi a tali che non esiste $b \neq a$ per cui $a \preceq b$ si chiamano **foglie**.
- Esiste un nodo minimo rispetto a $\preceq$, ovvero un nodo r tale che $r \preceq x$ per ogni nodo di X. Tale nodo si chiama **radice**.

L'albero si dice **binario** se

- Ogni nodo che non sia una foglia ha uno o al massimo due successori immediati, dove si dice che b è un **successore immediato** di a se $a \preceq b$, $a \neq b$ e non esiste un c tale che $a \preceq c \preceq b$, con $c \neq a$ e $c \neq b$.

La rappresentazione usuale di un albero binario è di questo tipo, dove la radice è in alto e l'albero cresce verso il basso:

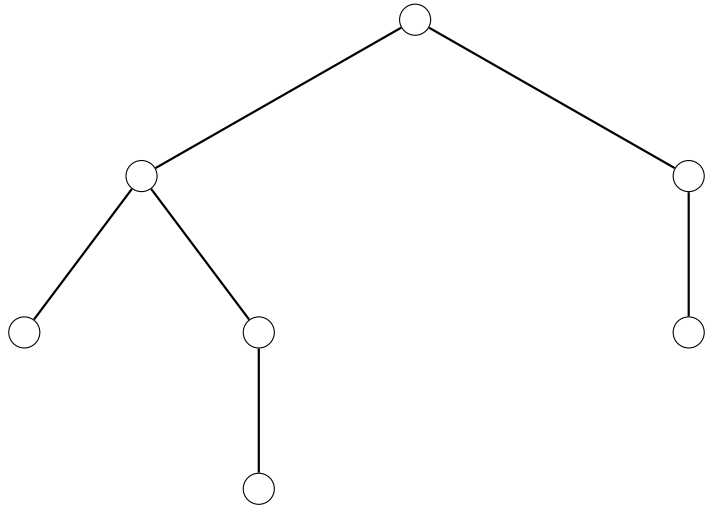


Figura: Esempio di rappresentazione di albero binario. I nodi sono disposti su quattro righe. Nell prima riga si trova la radice (un unico nodo). Nella seconda riga ci sono due nodi connessi con la radice. Nella terza riga ci sono tre nodi: due nodi connessi con il nodo di sinistra della riga precedente, un nodo connesso con il nodo di destra della riga precedente. Nella quarta riga c'è un solo nodo connesso con il secondo nodo della riga precedente.

Un **ramo** è un insieme totalmente ordinato di nodi che va dalla radice a una foglia. La sua lunghezza è il numero di nodi che vi appartengono. L'**altezza** dell'albero è la massima lunghezza dei suoi rami.

Un albero si dice etichettato se ad ogni nodo è associato un elemento di qualche insieme prefissato, che si chiama etichetta (*label*). Le etichette si possono sovrapporre ed identificare con i nodi.

Costruzione dell'albero sintattico

Algoritmo per costruire l'albero sintattico

L'albero sintattico di una proposizione P è definito in questo modo:

1. la radice è (etichettata con) la proposizione data;
2. ogni nodo ha nessuno, uno o due successori immediati a seconda che la proposizione etichetta del nodo sia atomica, della forma $(\neg \mathrm{Q})$ (ovvero il suo connettivo principale è $\neg$), o della forma $(\mathrm{Q} \,\square\, \mathrm{R})$ (ovvero il suo connettivo principale è un connettivo binario $\square$), rispettivamente. Nel secondo caso il successore è etichettato con Q, nel terzo caso i due successori sono etichettati rispettivamente con Q e con R.

Questo descrive un algoritmo che, partendo dalla radice etichettata con la proposizione iniziale, permette di costruire l'albero sintattico applicando ripetutamente la condizione 2 ai nodi costruiti al passo precedente.

Esempio

L'albero sintattico di $(((\mathrm{A}) \wedge (\neg(\mathrm{B}))) \rightarrow (\neg(\mathrm{A})))$ è il seguente:

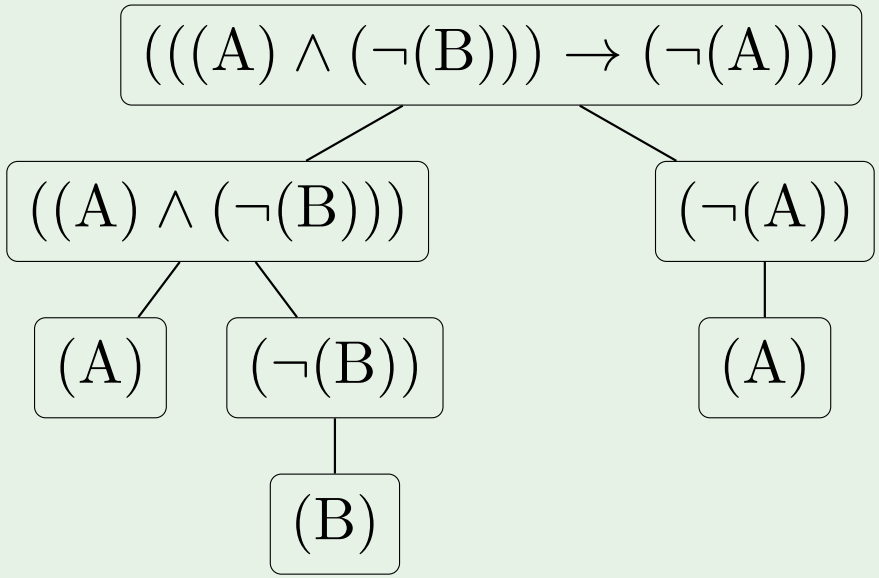


Figura: Albero sintattico di $(((\mathrm{A}) \wedge (\neg(\mathrm{B}))) \rightarrow (\neg(\mathrm{A})))$

L'albero è completo perché tutte le sue foglie sono etichettate con proposizioni atomiche (quindi l'algoritmo termina).

- Le etichette dei nodi dell'albero di costruzione di una proposizione sono le sue **sottoproposizioni**.
- Si dice che un simbolo **occorre** in una proposizione se è un elemento della stringa (che è la proposizione). Le **occorrenze** di un simbolo in una proposizione sono i vari posti della stringa in cui il simbolo si presenta, e il **numero di occorrenze** di un simbolo è il numero di volte che compare nella stringa.
- Le lettere che compaiono nelle (proposizioni atomiche nelle) foglie sono le lettere che occorrono nella proposizione.

Algoritmo per costruire l'albero sintattico

L'albero sintattico di una proposizione P è definito in questo modo:

1. la radice è (etichettata con) la proposizione data;
2. ogni nodo ha nessuno, uno o due successori immediati a seconda che la proposizione etichetta del nodo sia atomica, della forma $(\neg \mathrm{Q})$ (ovvero il suo connettivo principale è $\neg$), o della forma $(\mathrm{Q} \,\square\, \mathrm{R})$ (ovvero il suo connettivo principale è un connettivo binario $\square$), rispettivamente. Nel secondo caso il successore è etichettato con Q, nel terzo caso i due successori sono etichettati rispettivamente con Q e con R.

Per eseguire lo step 2 è necessario avere un metodo (= algoritmo) che permetta di determinare, se esiste, qual è il connettivo principale della proposizione nell'etichetta del nodo in esame: ad esempio, come si fa a determinare il connettivo principale della proposizione seguente?

$$(((\neg(((\mathrm{A}) \to (\mathrm{B})) \vee (\mathrm{C}))) \to ((\mathrm{A}) \wedge ((\mathrm{B}) \to (\mathrm{C})))) \wedge (\neg((\mathrm{A}) \leftrightarrow ((\mathrm{A}) \vee (\mathrm{B})))))$$

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 0

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 1

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 2

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 2

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 3

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 3

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 2

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 1

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 1

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 2

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 2

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 1

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Contatore di parentesi

Consideriamo la stringa

$$(((\neg(\mathrm{A})) \to (\mathrm{B})) \wedge (\neg(\mathrm{B})))$$

e supponiamo di voler trovare la parentesi che chiude la parentesi sinistra evidenziata in azzurro. Possiamo allora utilizzare un contatore che scorre la lista da sinistra verso destra partendo dalla parentesi cui siamo interessati, e scatta di $+1$ quando incontra una parentesi sinistra, di -1 quando incontra una parentesi destra. La prima parentesi su cui il contatore torna a 0 è proprio la parentesi di chiusura cercata. Nel nostro esempio:

Contatore: 0

Le parentesi sono essenziali per individuare il connettivo principale di una proposizione, e quindi per costruire il suo albero sintattico.

Come si individua il connettivo principale?

Sia P una proposizione non atomica. Il primo simbolo di P sarà sempre (, mentre il secondo simbolo sarà o $\neg$ oppure un'altra parentesi (.

- Nel primo caso il connettivo principale è proprio $\neg$ e ciò che lo segue *esclusa l'ultima parentesi*) è la sua sottoproposizione principale.

  Esempio: $(\neg(\neg(\mathrm{A}) \to (\mathrm{B})))$.

- Nel secondo caso, occorre trovare innanzitutto la parentesi che chiude quella che si trova al secondo posto: il simbolo che lo segue è il connettivo principale e le due sottoproposizioni principali sono tutto ciò che precede e segue tale connettivo *escluse le parentesi iniziali e finali*.

  Esempio: $(((\mathrm{B}) \to (\mathrm{A})) \vee (\neg(\mathrm{B})))$

Algoritmo per costruire l'albero sintattico

L'albero sintattico di una proposizione P è definito in questo modo:

1. la radice è (etichettata con) la proposizione data;
2. ogni nodo ha nessuno, uno o due successori immediati a seconda che la proposizione etichetta del nodo sia atomica, della forma $(\neg \mathrm{Q})$ (ovvero il suo connettivo principale è $\neg$), o della forma $(\mathrm{Q} \square \mathrm{R})$ (ovvero il suo connettivo principale è un connettivo binario $\square$), rispettivamente. Nel secondo caso il successore è etichettato con Q, nel terzo caso i due successori sono etichettati rispettivamente con Q e con R.

Attenzione! Se si raggiunge un nodo etichettato con una stringa che non è una formula atomica e non è nemmeno della forma $(\neg \mathrm{Q})$ o $(\mathrm{Q} \square \mathrm{R})$, l'algoritmo termina immediatamente e si conclude che P NON era una proposizione ben formata (ovvero $\mathrm{P} \notin \mathrm{Prop}(L)$).

Se invece questo non accade mai, dopo un certo numero di passi tutti i nodi privi di successori saranno etichettati con proposizioni atomiche: l'algoritmo quindi termina e quello costruito è l'albero sintattico di P.

Dunque l'algoritmo per la costruzione dell'albero sintattico permette di

- determinare se la stringa P che viene data come input è una proposizione ben formata (ovvero un elemento di $\mathrm{Prop}(L)$) oppure no;
- nel caso in cui $\mathrm{P} \in \mathrm{Prop}(L)$, individuarne tutte le sottoproposizioni e l'ordine con cui le regole di costruzione delle proposizioni sono state applicate ad esse per costruire P.

Inoltre, l'albero sintattico permette anche di determinare l'**altezza** della proposizione P: infatti, si dimostra facilmente per induzione strutturale che $\mathrm{ht}(\mathrm{P})$ coincide con l'altezza del suo albero di costruzione *diminuita di una unità*, ovvero con la massima lunghezza di un cammino che congiunga la radice a un nodo terminale.

Esempio

Sia P la proposizione $(((\mathrm{A}) \wedge (\neg(\mathrm{B}))) \to (\neg(\mathrm{A})))$. Abbiamo visto che il suo albero sintattico è il seguente:

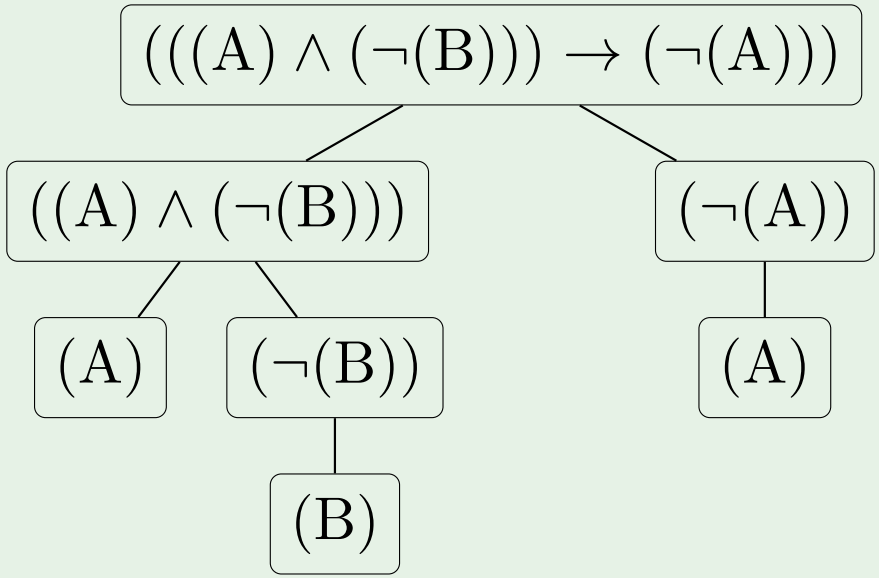


Figura: Albero sintattico della proposizione $(((\mathrm{A}) \wedge (\neg(\mathrm{B}))) \to (\neg(\mathrm{A})))$

Siccome l'altezza dell'albero è 4, l'altezza della proposizione $\mathrm{ht}(\mathrm{P})$ è $4 - 1$, ovvero 3.

Esercizio

Verificare quali delle seguenti stringhe sono proposizioni e quali no, costruendone l'albero sintattico e spiegando dove eventualmente la costruzione fallisce e per quale ragione:

- $((\mathrm{A}) \to ((\mathrm{B}) \vee (\neg(\mathrm{C}))))$
- $((\neg(\mathrm{A})) \wedge (\mathrm{B})) \vee (\mathrm{C}))$
- $(\neg\neg((\mathrm{A}) \to (\mathrm{B})))$
- $((((\mathrm{A}) \to (\mathrm{B})) \wedge (\mathrm{A})) \to (\mathrm{B}))$
- $(((\neg(\mathrm{A})) \wedge (\mathrm{B})) \vee (\mathrm{C}))$
- $(\neg(\neg\mathrm{A}))$
- $((\mathrm{A}) \wedge (\mathrm{B}) \wedge (\mathrm{C}))$.

Convenzioni sulle parentesi

Alcune parentesi sono sovrabbondanti: ad esempio, quelle della coppia più esterna e quelle nelle proposizioni atomiche, dove sono usate sia per uniformità sia per sottolineare la differenza tra una lettera come elemento dell'alfabeto e la lettera come proposizione.

Per comodità di scrittura e lettura riduciamo il numero di parentesi con le convenzioni seguenti.

Convenzioni sulle parentesi

- Non si scrivono le parentesi nelle proposizioni atomiche e non si scrivono le parentesi più esterne.
- Si eliminano alcune coppie di parentesi intorno ad alcune sottoproposizioni, utilizzando il criterio di priorità tra connettivi dato dalla seguente graduatoria:

$$\neg$$

$$\wedge \quad \vee$$

$$\rightarrow$$

$$\leftrightarrow$$

- Per occorrenze multiple dello stesso connettivo si prende in esame l'ultima, quella più a destra; questo significa che per formule composte con uno stesso connettivo ripetuto si conviene l'associazione a destra, cioè ad esempio con $\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C}$ si intende $\mathrm{A} \rightarrow (\mathrm{B} \rightarrow \mathrm{C})$, e con $\mathrm{A} \wedge \mathrm{B} \wedge \mathrm{C}$ si intende $\mathrm{A} \wedge (\mathrm{B} \wedge \mathrm{C})$.

Non tutte le parentesi si possono eliminare da una proposizione!

Esempio

Non si possono eliminare le parentesi da $\neg(\mathrm{A} \vee \mathrm{B})$. Togliendole si avrebbe infatti $\neg\mathrm{A} \vee \mathrm{B}$ che, per la priorità tra connettivi che abbiamo convenuto, rappresenta a formula $(\neg\mathrm{A}) \vee \mathrm{B}$.

Similmente, la scrittura $\mathrm{A} \wedge \mathrm{B} \vee \mathrm{C}$ non ha una lettura univoca e quindi è considerata priva di significato: le convenzioni sulle parentesi date non permette di capire se si intenda considerare la formula $(\mathrm{A} \wedge \mathrm{B}) \vee \mathrm{C}$ oppure $\mathrm{A} \wedge (\mathrm{B} \vee \mathrm{C})$ (poiché $\wedge$ e $\vee$ hanno la stessa priorità!).

Riassumendo: si possono eliminare parentesi fin tanto che il significato dell'espressione risultante resta univocamente determinato ed identico a quello della proposizione originale (ovvero alla versione formale con tutte le parentesi). Questo vuol dire che, utilizzando le convenzioni date, si devono poter reintrodurre le parentesi senza ambiguità, fino a ricostruire la proposizione originale.

Esempi

Data $\mathrm{A} \wedge \neg \mathrm{B} \rightarrow \neg \mathrm{A}$, la reintroduzione delle parentesi avviene attraverso questa successione di passi:

1. $(\mathrm{A}) \wedge \neg(\mathrm{B}) \rightarrow \neg(\mathrm{A})$
2. $(\mathrm{A}) \wedge \neg(\mathrm{B}) \rightarrow (\neg(\mathrm{A}))$
3. $(\mathrm{A}) \wedge (\neg(\mathrm{B})) \rightarrow (\neg(\mathrm{A}))$
4. $((\mathrm{A}) \wedge (\neg(\mathrm{B}))) \rightarrow (\neg(\mathrm{A}))$
5. $(((\mathrm{A}) \wedge (\neg(\mathrm{B}))) \rightarrow (\neg(\mathrm{A})))$.

I passi 2 e 3 si possono naturalmente fare in parallelo.

Reintrodurre le parentesi in $\mathrm{A} \to \neg(\mathrm{B} \wedge \neg\neg\mathrm{C})$

1. $(\mathrm{A}) \to \neg((\mathrm{B}) \wedge \neg\neg(\mathrm{C}))$
2. $(\mathrm{A}) \to \neg((\mathrm{B}) \wedge \neg(\neg(\mathrm{C})))$
3. $(\mathrm{A}) \to \neg((\mathrm{B}) \wedge (\neg(\neg(\mathrm{C}))))$
4. $(\mathrm{A}) \to (\neg((\mathrm{B}) \wedge (\neg(\neg(\mathrm{C})))))$
5. $((\mathrm{A}) \to (\neg((\mathrm{B}) \wedge (\neg(\neg(\mathrm{C}))))))$

oppure, per rendere più chiara la lettura

1. $\mathrm{A} \to \neg(\mathrm{B} \wedge \neg(\neg\mathrm{C}))$
2. $\mathrm{A} \to \neg(\mathrm{B} \wedge (\neg(\neg\mathrm{C})))$
3. $\mathrm{A} \to (\neg(\mathrm{B} \wedge (\neg(\neg\mathrm{C}))))$
4. $(\mathrm{A} \to (\neg(\mathrm{B} \wedge (\neg(\neg\mathrm{C})))))$

rimettendo infine le parentesi intorno alle lettere

5. $((\mathrm{A}) \to (\neg((\mathrm{B}) \wedge (\neg(\neg(\mathrm{C}))))))$

L'albero sintattico si può costruire direttamente anche per le espressioni in cui sono state omesse le parentesi seguendo le convenzioni stabilite.

Il connettivo principale è sempre quello di priorità più bassa tra quelli non contenuti in alcuna coppia di parentesi; se ve n'è più di uno con queste caratteristiche, il connettivo principale è quello più a sinistra tra di essi.

Esempio

L'albero sintattico di $\mathrm{A} \wedge \neg \mathrm{B} \to \neg \mathrm{A}$ è il seguente, essendo $\to$ il connettivo principale:

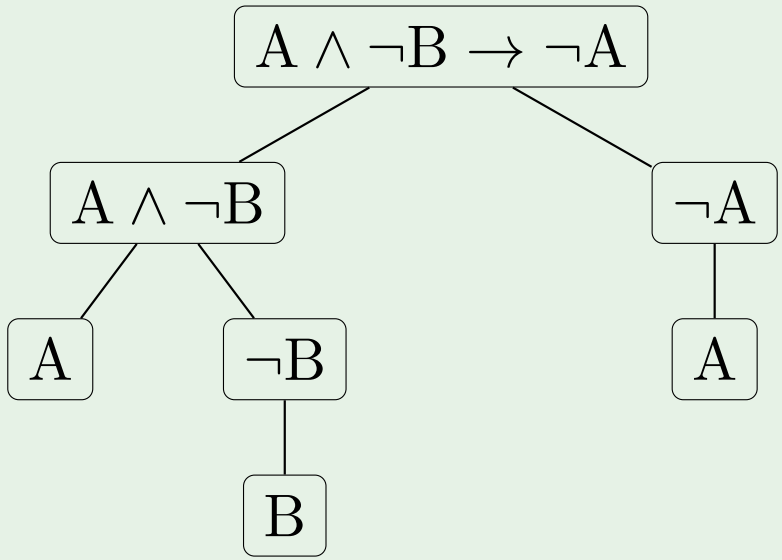
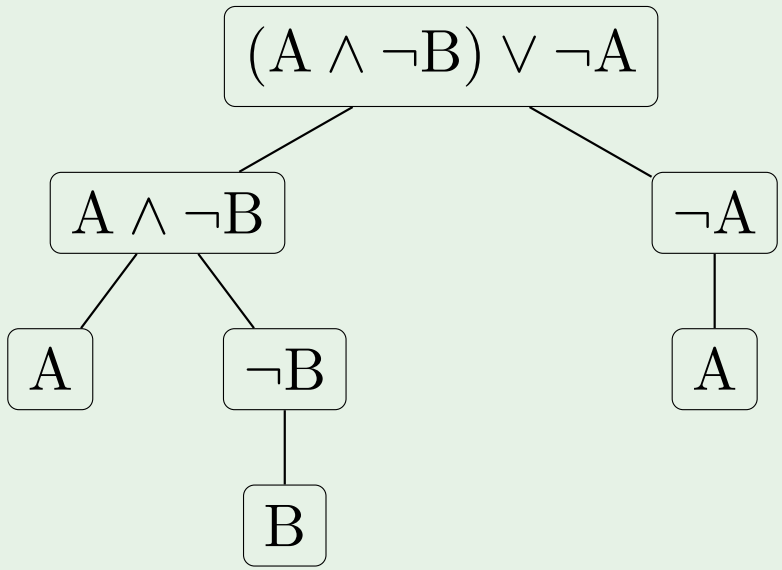


Figura: Albero sintattico della proposizione $\mathrm{A} \wedge \neg \mathrm{B} \to \neg \mathrm{A}$

L'albero sintattico può essere ulteriormente semplificato etichettando le foglie con le formule atomiche, e tutti gli altri nodi con il solo connettivo che serve per costruire la (sotto)proposizione corrispondente a partire dalle (sotto)proposizioni nei suoi successori immediati.

Esempio

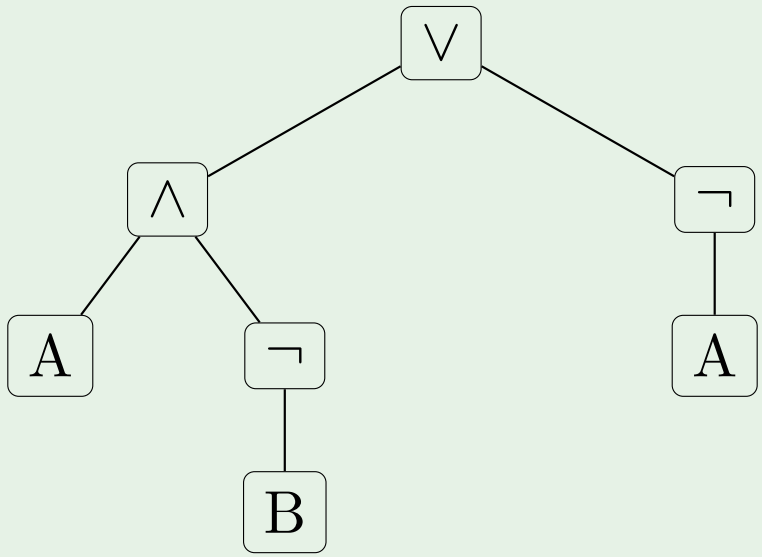
L'albero sintattico di $(\mathrm{A} \wedge \neg \mathrm{B}) \vee \neg \mathrm{A}$



L'albero sintattico può essere ulteriormente semplificato etichettando le foglie con le formule atomiche, e tutti gli altri nodi con il solo connettivo che serve per costruire la (sotto)proposizione corrispondente a partire dalle (sotto)proposizioni nei suoi successori immediati.

Esempio

L'albero sintattico di $(\mathrm{A} \wedge \neg \mathrm{B}) \vee \neg \mathrm{A}$ si può disegnare anche così:



Per semplificare la lettura di alcune formule, nella pratica ometteremo spesso molte parentesi (ma non tutte) e useremo anche le parentesi quadre

$$[\quad\quad]$$

al posto di quelle tonde.

Esempio

Per esempio scriveremo espressioni come

$$(\mathrm{A} \wedge \neg \mathrm{B}) \to [\neg \mathrm{C} \leftrightarrow (\mathrm{B} \vee \neg \mathrm{D})].$$

Tale espressione non è altro che una "semplificazione" della proposizione che si ottiene sostituendo le parentesi quadre con le parentesi tonde e reintroducendo tutte le parentesi nella maniera opportuna, ovvero della proposizione

$$(((\mathrm{A}) \wedge (\neg(\mathrm{B}))) \to ((\neg(\mathrm{C})) \leftrightarrow ((\mathrm{B}) \vee (\neg(\mathrm{D}))))).$$

Esercizio

Reintrodurre le parentesi nelle seguenti stringhe in modo da ottenere, se possibile, proposizioni, o se no spiegare il perché.

- $\mathrm{A} \to \mathrm{B} \vee \neg \mathrm{C}$
- $\mathrm{A} \wedge (\to \mathrm{C} \vee \mathrm{A})$
- $(\mathrm{A} \to \mathrm{B}) \wedge \mathrm{A} \to \mathrm{B}$
- $\mathrm{A} \to \mathrm{B} \wedge \mathrm{A} \to \mathrm{B}$
- $\mathrm{A} \vee \mathrm{B} \wedge \mathrm{C} \to \neg \mathrm{A}$
- $\mathrm{A} \wedge \mathrm{B} \wedge \mathrm{C} \vee \neg \mathrm{C}$
- $\neg\neg \mathrm{A}$
- $\neg \mathrm{A} \wedge \mathrm{B} \vee \mathrm{C}$
- $\mathrm{A} \vee \neg \mathrm{B} \to \neg \mathrm{A} \vee \mathrm{B}$

Esercizio

Costruire l'albero sintattico delle seguenti formule (in cui sono state omesse alcune parentesi secondo le convenzioni adottate).

- $\neg \mathrm{A} \rightarrow \neg[(\mathrm{B} \leftrightarrow \neg(\mathrm{A} \vee \mathrm{C})) \vee \neg \mathrm{A}]$
- $(\mathrm{A} \leftrightarrow \neg \mathrm{B}) \rightarrow [\neg(\mathrm{A} \wedge \neg \mathrm{B}) \rightarrow \neg(\mathrm{C} \wedge \neg \mathrm{D})]$

Calcolare anche l'altezza di ciascun albero e l'altezza della formula a cui è associato.