# Alberi sintattici

Un albero si dice *etichettato* se a ogni nodo è associato un elemento (*etichetta*, o *label*) preso da un qualche insieme prefissato.

**Definizione.** L'*albero sintattico* di una formula proposizionale $P$ è l'albero binario etichettato finito tale che:

(1) La radice è etichettata con $P$

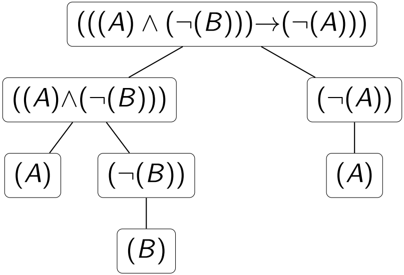
(2) Se un nodo è etichettato con una formula $Q$, allora:

- se $Q$ è una formula atomica, allora tale nodo è una foglia
- se $Q$ è $(\neg R)$, allora tale nodo ha un unico successore immediato, etichettato con $R$
- se $Q$ è $(R \Box S)$, dove $\Box$ è un connettivo binario, allora tale nodo ha due successori immediati, uno etichettato con $R$, l'altro etichettato con $S$

Questa definizione descrive un algoritmo che permette di costruire l'albero sintattico di una formula, partendo dalla radice e applicando la clausola (2) ai nodi via via costruiti.

# Esempio

L'albero sintattico di $(((A) \wedge (\neg(B))) \to (\neg(A)))$ è:




L'algoritmo di costruzione termina sempre, perché l'etichetta di ogni nodo (esclusa la radice) è di lunghezza inferiore all'etichetta del predecessore immediato.

# Osservazioni

Sia $P$ una formula, e sia $T$ il suo albero sintattico.

- Le etichette dei nodi di $T$ sono le *sottoformule* di $P$
- Le etichette delle foglie di $T$ sono le formule atomiche che occorrono in $P$
- **Per poter applicare l'algoritmo che produce l'albero sintattico, occorre saper riconoscere qual è il connettivo principale di una formula**

**Esempio.** Qual è il connettivo principale di

$$(((\neg(((A) \to (B))\vee(C))) \to ((A)\wedge((B) \to (C))))\wedge(\neg((A) \leftrightarrow ((A)\vee(B)))))?$$

C'è un algoritmo per saperlo, che si basa sulla posizione delle parentesi.

# Contatore di parentesi

Nella costruzione di una formula, ogni parentesi ( è introdotta contemporaneamente alla sua chiusura ), che si trova da qualche parte alla sua destra.

**Esempio.** Nella formula

$$(((\neg(A)) \to (B)) \wedge (\neg(B)))$$

qual è la parentesi associata alla seconda ( da sinistra?

Per saperlo si può utilizzare un contatore che:

- all'inizio segna 0
- parte dalla parentesi in questione e scorre la stringa verso destra
- ogni volta che incontra una parentesi (, compresa quella iniziale, aggiunge 1
- ogni volta che incontra una parentesi ) sottrae 1
- la prima parentesi su cui il contatore torna a 0 è la parentesi cercata

# Esempio

$$
\begin{array}{ccccccccccccccccccccc}
( & ( & ( & \neg & ( & A & ) & ) & \rightarrow & ( & B & ) & ) & \wedge & ( & \neg & ( & B & ) & ) & ) \\
 & 1 & 2 & & 3 & & 2 & 1 & & 2 & & 1 & 0 & & & & & & & &
\end{array}
$$

# Individuazione del connettivo principale

Sia $P$ una formula non atomica.

- Il primo simbolo è sempre una parentesi ( : la sua chiusura è l'ultimo simbolo della formula
- Il secondo simbolo può essere:
  (1) il connettivo $\neg$; oppure
  (2) un'altra parentesi (

# Individuazione del connettivo principale

- Nel caso (1), il connettivo principale è quel connettivo $\neg$, e la sottoformula principale a cui è applicato è tutta la stringa alla sua destra escluso l'ultimo simbolo )

  **Esempio.** Nella formula

  $$(\neg(\neg(A) \rightarrow (B)))$$

  il connettivo principale è il primo $\neg$. La sottoformula principale è $(\neg(A) \rightarrow (B))$.

# Individuazione del connettivo principale

- Nel caso (2) si deve anzitutto individuare la parentesi ) che chiude la parentesi ( in seconda posizione della stringa. Il simbolo che la segue è il connettivo principale; esso è applicato alle due sottoformule principali costituite da tutto ciò che precede quel connettivo, salvo la prima parentesi (, e da tutto ciò che segue quel connettivo, salvo l'ultima parentesi ).

  **Esempio.** Nella formula

  $$(((B) \to (A)) \vee (\neg(B)))$$

  il connettivo principale è $\vee$, che è applicato alle due sottoformule principali $((B) \to (A))$ e $(\neg(B))$

# Nota sull'algoritmo

L'algoritmo di costruzione dell'albero sintattico può essere applicato a qualunque stringa $P$, non solo alle stringhe che sono formule ben formate.

- Se la stringa $P$ non è una formula, prima o poi l'algoritmo si blocca. L'albero costruito fino a quel momento non è un albero sintattico, perché le foglie non sono etichettate da una formula atomica.
- Se $P$ è una formula, l'algoritmo lavora fino a produrre l'albero sintattico di $P$; questa situazione si riconosce perché tutte le foglie sono etichettate da una formula atomica.

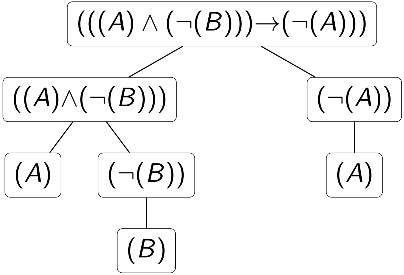
# Nota sull'algoritmo

Quindi l'algoritmo di costruzione dell'albero sintattico:

- Determina se una stringa è una formula ben formata, cioè un elemento di $Prop(L)$
- In caso affermativo, forrnisce l'albero sintattico della formula
- Fornisce l'altezza di una formula, che coincide con l'altezza dell'albero sintattico

# Esempio

L'albero sintattico di $(((A) \wedge (\neg(B))) \to (\neg(A)))$ è:




Quindi $ht[(((A) \wedge (\neg(B))) \to (\neg(A)))] = 3$.

# Convenzioni sulle parentesi

La definizione data di costruzione di una formula comporta l'uso di una quantità enorme di parentesi. L'uso di tutte queste parentesi permette di descrivere (abbastanza) facilmente l'algoritmo di costruzione dell'albero sintattico, ma rende più difficile e fastidiosa la lettura umana di una formula.

Si conviene allora che, nella pratica, si possano eliminare (o aggiungere) parentesi a patto che la formula di cui si tratta e quindi la sua costruzione — con tutte le sua parentesi! — sia sempre ricavabile. Talora si possono usare coppie di parentesi di altra forma, per es. [ e ], per evidenziare meglio visivamente le parentesi appaiate.

# Convenzioni sulle parentesi

Quindi, tipicamente:

- si eliminano le parentesi attorno alle sottoformule atomiche e si elimina la coppia di parentesi più esterna nella formula
- si eliminano le coppie di parentesi attorno alle sottoformule quando queste siano coerenti con la seguente lista gerarchica dei connettivi:
  - $\neg$
  - $\vee, \wedge$
  - $\rightarrow, \leftrightarrow$
- è anche permesso scrivere sottoformule del tipo $A_1 \vee A_2 \vee \ldots \vee A_k$ o $A_1 \wedge A_2 \wedge \ldots \wedge A_k$ nel caso in cui siano associate da sinistra a destra; per esempio, $A_1 \vee A_2 \vee A_3$ significa $(A_1 \vee A_2) \vee A_3$, che a sua volta significa $(((A_1) \vee (A_2)) \vee (A_3))$.

# Esempi

- Non si possono ulteriormente togliere parentesi da $\neg(A \vee B)$, in quanto, per la convenzione sulla priorità dei connettivi, $\neg A \vee B$ è lo stesso di $(\neg A) \vee B$ — che a sua volta è l'abbreviazione di $((\neg(A)) \vee (B))$
- Non sono ammesse espressioni quali $A \vee B \wedge C$ perché, avendo $\vee, \wedge$ uguale priorità, tale espressione potrebbe leggersi come la formula $(A \vee B) \wedge C$ o come la formula $A \vee (B \wedge C)$.

Rispettando tutte queste convenzioni è sempre possibile ricostruire la formula originaria, con tutte le sue parentesi, riprercorrendo la sua costruzione.

# Esempio

Per reintrodurre le parentesi in

$$A \land \neg B \to \neg A$$

- $(A) \land \neg(B) \to \neg(A)$
- $(A) \land (\neg(B)) \to (\neg(A))$
- $((A) \land (\neg(B))) \to (\neg(A))$
- $(((A) \land (\neg(B))) \to (\neg(A)))$

# Esempio

Per reintrodurre le parentesi in

$$A \to \neg(B \wedge \neg\neg C)$$

- $(A) \to \neg((B) \wedge \neg\neg(C))$
- $(A) \to \neg((B) \wedge \neg(\neg(C)))$
- $(A) \to \neg((B) \wedge (\neg(\neg(C))))$
- $(A) \to (\neg((B) \wedge (\neg(\neg(C)))))$
- $((A) \to (\neg((B) \wedge (\neg(\neg(C))))))$

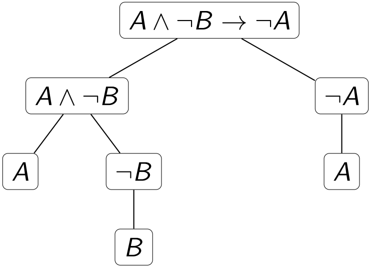
# Ancora sull'albero sintattico

La costruzione dell'albero sintattico si può fare direttamente sulle espressioni in cui siano state omesse le parentesi secondo le convenzioni adottate:

Il connettivo principale è il connettivo di priorità più bassa tra quelli non contenuti in alcuna coppia di parentesi, o nella coppia più esterna; in caso di disgiunzioni o congiunzioni di più sottoformule, il connettivo principale è quello più a destra.
Di conseguenza si determinano le sottoformule principali.

# Esempio

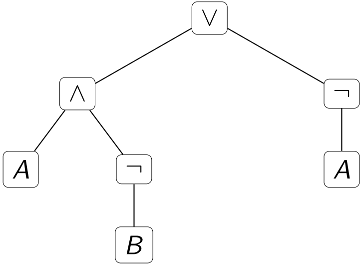
Per costruire l'albero sintattico di $A \wedge \neg B \to \neg A$:



# Ancora sull'albero sintattico

L'albero sintattico può essere semplificato usando per i nodi che non sono foglie solo il connettivo principale della sottoformula che dovrebbe costituire la sua etichetta:

**Esempio.**




La formula di cui questo è l'albero sintattico si può ricostruire partendo dalle formule atomiche che etichettano le foglie e salendo lungo l'albero, applicando ogni volta il connettivo che etichetta il nodo precedente:

$$(A \wedge \neg B) \vee \neg A$$

# Esercizio 1

Verificare quali delle seguenti stringhe sono formule (secondo la definizione, senza la convenzione sulle parentesi) utilizzando l'algoritmo dell'albero sintattico.

1. $((A) \to ((B) \vee (\neg(C))))$
2. $((\neg(A)) \wedge (B)) \vee (C))$
3. $(\neg\neg((A) \to (B)))$
4. $((((A) \to (B)) \wedge (A)) \to (B))$
5. $(((\neg(A)) \wedge (B)) \vee (C))$
6. $(\neg(\neg A))$
7. $((A) \wedge (B) \wedge (C))$

**Risposta parziale.**
1, 4, 5.

# Esercizio 2

Reintrodurre le parentesi nelle seguenti stringhe in modo da ottenere, ove possibile, delle formule.

1. $A \to B \vee \neg C$
2. $A \wedge (\to C \vee A)$
3. $(A \to B) \wedge A \to B$
4. $A \to B \wedge A \to B$
5. $A \vee B \wedge C \to \neg A$
6. $A \wedge B \wedge C \vee \neg C$
7. $\neg\neg A$
8. $\neg A \wedge B \vee C$
9. $A \vee \neg B \to \neg A \vee B$

# Esercizio 2, svolgimento

1. - $(A) \to (B \vee \neg C)$
   - $(A) \to ((B) \vee (\neg C))$
   - $(A) \to ((B) \vee (\neg(C)))$
   - $((A) \to ((B) \vee (\neg(C))))$
2. - $(A) \wedge (\to C \vee A)$
   - Se la sottostringa $(\to C \vee A)$ fosse una formula, il suo connettivo principale sarebbe $\to$. Tuttavia in tale sottostringa $\to$ è preceduto da (.
     La stringa $(A) \wedge (\to C \vee A)$ non è una formula.
3. - $((A \to B) \wedge A) \to (B)$
   - $(((A) \to (B)) \wedge (A)) \to (B)$
   - $((((A) \to (B)) \wedge (A)) \to (B))$
4. La stringa, senza parentesi, contiene due occorrenze di $\to$.
   La stringa $A \to B \wedge A \to B$ non è una formula.
5. - $(A \vee B \wedge C) \to (\neg A)$
   - La sottostringa $(A \vee B \wedge C)$ contiene occorrenze di $\vee$ e $\wedge$.
     La stringa $A \vee B \wedge C \to \neg A$ non è una formula.

# Esercizio 2, svolgimento (cont.)

6. La stringa, senza parentesi, contiene occorrente di $\wedge$ e $\vee$. La stringa $A \wedge B \wedge C \vee \neg C$ non è una formula.
7. - $\neg(\neg A)$
   - $\neg(\neg(A))$
   - $(\neg(\neg(A)))$
8. La stringa, senza parentesi, contiene occorrenze di $\wedge$ e $\vee$. La stringa $\neg A \wedge B \vee C$ non è una formula.
9. - $(A \vee \neg B) \to (\neg A \vee B)$
   - $((A) \vee (\neg B)) \to ((\neg A) \vee (B))$
   - $((A) \vee (\neg(B))) \to ((\neg(A)) \vee (B))$
   - $(((A) \vee (\neg(B))) \to ((\neg(A)) \vee (B)))$

# Esercizio 3

Costruire l'albero sintattico delle seguenti formule e calcolarne l'altezza.

1. $\neg A \to \neg[(B \leftrightarrow \neg(A \vee C)) \vee \neg A]$
2. $(A \leftrightarrow \neg B) \to [\neg(A \wedge \neg B) \to \neg(C \wedge \neg D)]$

**Risposta parziale.**

1. L'altezza della formula è 6.
2. L'altezza della formula è 5.

# Interpretazioni

Sia $L$ un insieme di lettere proposizionali.

**Definizione**

Un'*interpretazione* è una funzione

$$i : L \to \{0,1\}$$

Cioè, un'interpretazione assegna un valore di verità a ognuna delle lettere proposizionali.

# Interpretazioni

## Esempio

Sia $L = \{A, B\}$.
L'interpretazione $i : L \to \{0, 1\}$ definita da

$$i(A) = 0, \qquad i(B) = 1$$

assegna ad $A$ il valore di verità *falso*, e a $B$ il valore di verità *vero*.

L'interpretazione $i' : L \to \{0, 1\}$ definita da

$$i'(A) = i'(B) = 1$$

assegna sia ad $A$ sia a $B$ il valore di verità *vero*.