Logica Matematica
5.3 – Formule

Docenti: Alessandro Andretta e Luca Motto Ros

Dipartimento di Matematica
Università di Torino

Formule atomiche

Fissiamo un linguaggio del prim'ordine L.

Definizione

Una **formula atomica** (nel linguaggio L) è una stringa della forma

$$(R(t_1, \dots, t_n))$$

dove R è un simbolo di predicato n-ario in L e $t_1, \dots, t_n$ sono termini (nel linguaggio L), oppure della forma

$$(t_1 = t_2)$$

dove t_1, t_2 sono termini (nel linguaggio L).

Attenzione! Anche in questo caso le virgole non sono necessarie, ma possono aiutare nella lettura della formula.

Formule atomiche, equazioni e disequazioni

Consideriamo il linguaggio $L = \{<, +, \cdot, 1, 0\}$ dove $<$ è un simbolo di relazione binario, $+$ e $\cdot$ sono simboli di funzione binari, e $1, 0$ sono simboli di costante. Abbiamo visto che il termine t

$$+(\cdot(x, x), 1)$$

corrisponde (utilizzando la notazione infissa) al polinomio

$$x^2 + 1$$

La formula atomica

$$(+(\cdot(x, x), 1) = 0)$$

eprime allora nel linguaggio L l'equazione

$$x^2 + 1 = 0,$$

mentre

$$(< (+(\cdot(x, x), 1), 0))$$

esprime, utilizzando la notazione infissa per $<$, la disequazione $x^2 + 1 < 0$.

Per riconoscere se una data stringa è una formula atomica, si procede come segue:

Algoritmo per il riconoscimento di formule atomiche

- Il primo e l'ultimo simbolo della stringa devono essere una parentesi sinistra e una parentesi destra, rispettivamente.
- Se il secondo simbolo della stringa è un simbolo di relazione n-ario $R \in L$, allora la formula atomica deve essere del tipo $(R(t_1, \ldots, t_n))$, dove $n = \mathrm{ar}(R)$: quindi si controlla che il terzo simbolo sia una parentesi sinistra e il penultimo simbolo sia una parentesi destra, si analizza la stringa compresa tra queste parentesi per individuare i termini $t_1, \ldots, t_n$ (l'algoritmo è lo stesso di quello utilizzato nel caso dei termini), e infine se ne costruisce l'albero sintattico per controllare che siano termini ben formati.
- Se il secondo simbolo della stringa è una variabile, una costante o un simbolo di funzione, allora la formula deve essere del tipo $(t_1 = t_2)$: si cerca allora il simbolo di uguaglianza (ce ne deve essere solo uno!), si individuano i termini t_1 e t_2, e se ne costruisce l'albero sintattico per controllare che siano termini ben formati.

Nei restanti casi, la stringa data non era una formula atomica.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(P(f(x)c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(P(f(x)c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(P(f(x)c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(P(f(x)c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(P(f(x)c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(P(f(x)c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(P(f(x)c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(P(f(x)c))$$

è una formula atomica oppure no.

Dall'analisi fatta, risulta che la stringa è del tipo $(P(t_1 t_2))$, dove t_1 è $f(x)$ e t_2 è c: poiché questi ultimo sono termini ben formati, la stringa è una formula atomica. Reintroducendo le virgole, tale formula atomica si scrive

$$(P(f(x), c)).$$

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(f(f(x)) = f(c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(f(f(x)) = f(c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$({\color{red}f}(f(x)) = f(c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(f(f(x)) = f(c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(f(f(x)) = f(c))$$

è una formula atomica oppure no.

Esempio

Sia $L = \{P, f, c\}$ con P simbolo di relazione binario, f simbolo di funzione unario e c simbolo di costante. Verifichiamo se la stringa

$$(f(f(x)) = f(c))$$

è una formula atomica oppure no.

Dall'analisi fatta, risulta che la stringa è del tipo $(t_1 = t_2)$, dove t_1 è $f(f(x))$ e t_2 è $f(c)$: poiché questi ultimo sono termini ben formati, la stringa è una formula atomica.

Formule del prim'ordine

L'insieme delle **formule** del linguaggio L (o, più brevemente, L-**formule**) è definito *ricorsivamente* dalle clausole:

- una formula atomica è una formula;
- se φ è una formula, allora anche $(\neg\varphi)$ è una formula,
- se φ e ψ sono formule, allora anche $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ e $(\varphi \leftrightarrow \psi)$ sono formule;
- se φ è una formula e x è una variabile, allora anche $(\exists x\varphi)$ e $(\forall x\varphi)$ sono formule. In questo caso, φ viene detta **raggio d'azione** del quantificatore $\exists x$ o $\forall x$.

Useremo le lettere greche φ, ψ, e χ, variamente decorate, per le formule.

Tecnicamente, bisognerebbe di nuovo dare una definizione per ricorsione degli insiemi Fml_n per $n \in \mathbb{N}$: Fml_0 è l'insieme delle formule atomiche e Fml_{n+1} è l'unione di Fml_n con l'insieme delle formule che si ottengono applicando una delle regole qui sopra a formule in Fml_n. L'insieme delle formule è allora $\mathsf{Fml} = \bigcup_{n\in\mathbb{N}} \mathsf{Fml}_n$. L'**altezza** $\mathrm{ht}(\varphi)$ di una formula $\varphi \in \mathsf{Fml}$ è definita nella maniera usuale.

La **costante logica principale** di una L-formula (non atomica) φ è l'ultima costante logica introdotta per creare φ in accordo con la definizione ricorsiva data. Più precisamente:

1. se φ è della forma $(\neg\psi)$, allora $\neg$ è la costante logica principale di φ, mentre ψ viene detta sottoformula principale di φ;
2. se φ è della forma $(\psi \square \chi)$, dove $\square$ è uno dei connettivi binari $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, allora $\square$ è la costante logica principale di φ, mentre ψ e χ sono le sottoformule principali di φ;
3. infine, se φ è della forma $(\exists x\, \psi)$ oppure della forma $(\forall x\, \psi)$, allora $\exists$ e $\forall$ sono, rispettivamente, la costante logica principale di φ, mentre ψ viene detta sottoformula principale di φ.

Nei casi 1 e 2 parliamo anche di **connettivo principale** di φ, nel caso 3 parliamo invece di **quantificatore principale** di φ.

Diciamo che una formula φ è una **negazione**, **congiunzione**, **disgiunzione**, **implicazione**, **bi-implicazione**, **formula esistenziale** oppure **formula universale** quando la sua costante logica principale è $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\exists$ o $\forall$, rispettivamente.

Per individuare la costante logica principale di una L-formula φ si usa (l'ovvia variante del)l'algoritmo visto per la logica proposizionale.

Il primo e l'ultimo simbolo della stringa devono essere una parentesi sinistra e una parentesi destra, rispettivamente. Consideriamo il secondo simbolo della stringa.

- Se il secondo simbolo è $\neg$ oppure una parentesi sinistra $($, allora si procede come visto per la logica proposizionale: nel primo caso la costante logica principale è proprio $\neg$, mentre nel secondo caso la costante logica principale è uno dei connettivi binari $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, e precisamente quello che segue la parentesi destra che chiude la parentesi sinistra in esame.
- Altrimenti, il secondo simbolo è $\exists$ oppure $\forall$: in questo caso, tale quantificatore è proprio la costante logica principale di φ. Esso dovrà necessariamente essere seguito da una variabile, e ciò che segue tale variabile (esclusa l'ultima parentesi di chiusura) è la sottoformula principale di φ.

Albero sintattico di una formula

Costruzione dell'albero sintattico di una formula del prim'ordine

1. Si etichetta la radice con la formula data.
2. Sia φ la formula che compare nell'etichetta di un nodo:
   - se φ è una formula atomica (ben formata) non si aggiunge alcun successore al nodo, che diventerà una foglia dell'albero;
   - se φ è una negazione, ovvero è del tipo $(\neg\psi)$, si aggiunge un solo successore al nodo e lo si etichetta con ψ;
   - se la costante logica principale di φ è un connettivo binario $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, ovvero φ è del tipo $(\psi \square \chi)$ con $\square$ connettivo binario, allora si aggiungono due successori al nodo etichettandoli con ψ e χ, rispettivamente;
   - se la costante logica principale di φ è un quantificatore, ovvero φ è del tipo $(\exists x\, \psi)$ oppure $(\forall x\, \psi)$, allora si aggiunge un solo successore al nodo etichettandolo con ψ.

Esempio

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \land ((P(x)) \to (f(x) = c))))$$

è

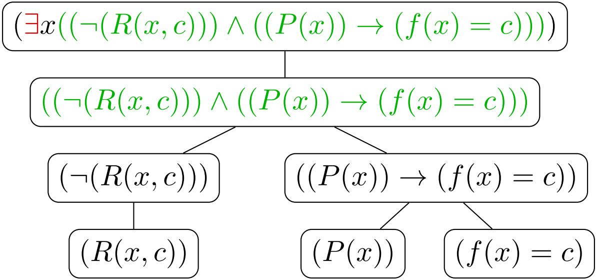
$$\begin{tikzcd}
 & & \boxed{(\exists x((\neg(R(x,c))) \land ((P(x)) \to (f(x) = c))))} \arrow[d, dash] & & \\
 & & \boxed{((\neg(R(x,c))) \land ((P(x)) \to (f(x) = c)))} \arrow[dl, dash] \arrow[dr, dash] & & \\
 & \boxed{(\neg(R(x,c)))} \arrow[d, dash] & & \boxed{((P(x)) \to (f(x) = c))} \arrow[dl, dash] \arrow[dr, dash] & \\
 & \boxed{(R(x,c))} & \boxed{(P(x))} & & \boxed{(f(x) = c)}
\end{tikzcd}$$

Esempio

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \wedge ((P(x)) \to (f(x) = c))))$$

è



L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \wedge ((P(x)) \to (f(x) = c))))$$

è

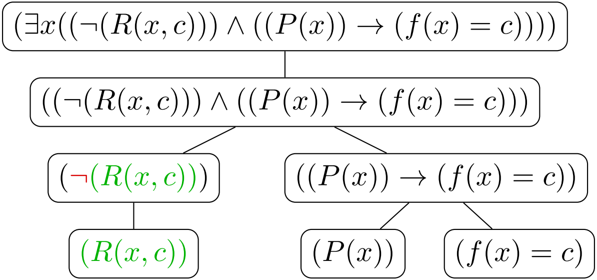
- $(\exists x((\neg(R(x,c))) \wedge ((P(x)) \to (f(x) = c))))$
  - $((\neg(R(x,c))) \wedge ((P(x)) \to (f(x) = c)))$
    - $(\neg(R(x,c)))$
      - $(R(x,c))$
    - $((P(x)) \to (f(x) = c))$
      - $(P(x))$
      - $(f(x) = c)$

Esempio

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \wedge ((P(x)) \to (f(x)=c))))$$

è

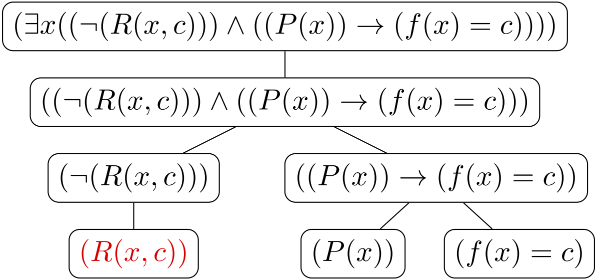


Esempio

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \wedge ((P(x)) \rightarrow (f(x) = c))))$$

è

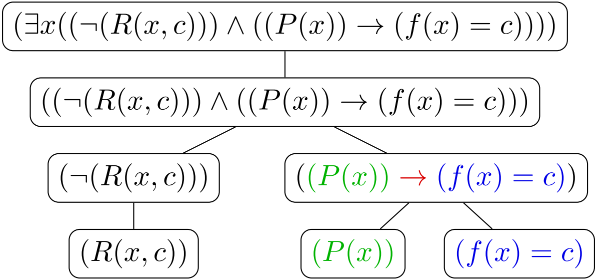


Esempio

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \wedge ((P(x)) \to (f(x) = c))))$$

è

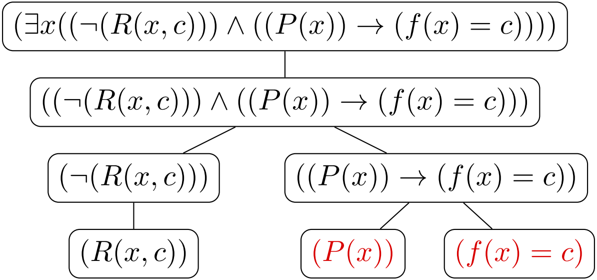


Esempio

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \land ((P(x)) \to (f(x) = c))))$$

è



Esempio

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \land ((P(x)) \to (f(x) = c))))$$

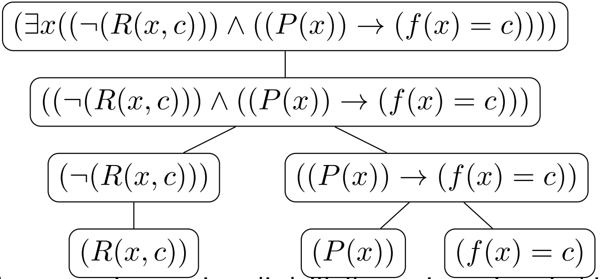
è

- $(\exists x((\neg(R(x,c))) \land ((P(x)) \to (f(x) = c))))$
  - $((\neg(R(x,c))) \land ((P(x)) \to (f(x) = c)))$
    - $(\neg(R(x,c)))$
      - $(R(x,c))$
    - $((P(x)) \to (f(x) = c))$
      - $(P(x))$
      - $(f(x) = c)$

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \wedge ((P(x)) \to (f(x)=c))))$$

è



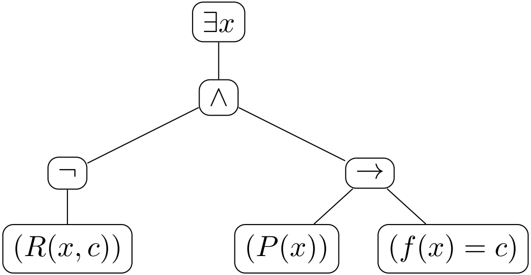
Le formula che compaiono nei nodi dell'albero sintattico si chiamano **sottoformule** della formula data.
Anche per le formule del prim'ordine vale la regola che l'altezza della formula è uguale all'altezza dell'albero diminuita di una unità.

Esempio

L'albero sintattico di

$$(\exists x((\neg(R(x,c))) \wedge ((P(x)) \rightarrow (f(x) = c))))$$

è



Le formula che compaiono nei nodi dell'albero sintattico si chiamano **sottoformule** della formula data.
Anche per le formule del prim'ordine vale la regola che l'altezza della formula è uguale all'altezza dell'albero diminuita di una unità.

Esercizio

Calcolare l'albero sintattico della formula

$$((\exists x(\forall y\,((P(x,y)) \to (Q(x))))) \to ((\forall z(R(z))) \vee (S(z)))).$$