



SAPIENZA
UNIVERSITÀ DI ROMA

“SAPIENZA” UNIVERSITÀ DI ROMA
INGEGNERIA DELL'INFORMAZIONE,
INFORMATICA E STATISTICA
DIPARTIMENTO DI INFORMATICA

Linguaggi di Programmazione

Author
Simone Bianco

2 ottobre 2023

Indice

Informazioni e Contatti	1
1 Induzione strutturale	2
1.1 Algebre induttive	2
1.2 Strutture dati induttive	7
1.3 Sintassi astratta	11

Informazioni e Contatti

Appunti e riassunti personali raccolti in ambito del corso di *Linguaggi di Programmazione* offerto dal corso di laurea in Informatica dell'Università degli Studi di Roma "La Sapienza".

Ulteriori informazioni ed appunti possono essere trovati al seguente link:

<https://github.com/Exyss/university-notes>. Chiunque si senta libero di segnalare incorrettezze, migliorie o richieste tramite il sistema di Issues fornito da GitHub stesso o contattando in privato l'autore :

- Email: bianco.simone@outlook.it
- LinkedIn: [Simone Bianco](#)

Gli appunti sono in continuo aggiornamento, pertanto, previa segnalazione, si prega di controllare se le modifiche siano già state apportate nella versione più recente.

Prerequisiti consigliati per lo studio:

Apprendimento del materiale relativo al corso *Algebra*.

Licence:

These documents are distributed under the [GNU Free Documentation License](#), a form of copyleft intended for use on a manual, textbook or other documents. Material licensed under the current version of the license can be used for any purpose, as long as the use meets certain conditions:

- All previous authors of the work must be **attributed**.
- All changes to the work must be **logged**.
- All derivative works must be **licensed under the same license**.
- The full text of the license, unmodified invariant sections as defined by the author if any, and any other added warranty disclaimers (such as a general disclaimer alerting readers that the document may not be accurate for example) and copyright notices from previous versions must be maintained.
- Technical measures such as DRM may not be used to control or obstruct distribution or editing of the document.

1

Induzione strutturale

1.1 Algebre induttive

Definizione 1: Assiomi di Peano

L'insieme dei numeri naturali \mathbb{N} è definito secondo i seguenti **assiomi di Peano**:

1. $0 \in \mathbb{N}$
2. $n \in \mathbb{N} \implies \text{succ}(n) \in \mathbb{N}$, dove $\text{succ} : \mathbb{N} \rightarrow \mathbb{N}$ è la funzione successore
3. $\forall n, m \in \mathbb{N}, \text{succ}(n) = \text{succ}(m) \implies n = m$, ossia succ è iniettiva
4. $\nexists n \in \mathbb{N} \mid \text{succ}(n) = 0$
5. $\forall S \subseteq \mathbb{N} \mid (0 \in S \wedge (n \in S \implies \text{succ}(n) \in S)) \implies S = \mathbb{N}$

Proposizione 1: Numeri naturali di Von Neumann

I numeri naturali data da Von Neumann, indicati con \mathcal{N} , definiti come:

$$0_{\mathcal{N}} := \{\}$$

$$1_{\mathcal{N}} := \{\{\}\}$$

$$2_{\mathcal{N}} := \{\{\}, \{\{\}\}\}$$

$$3_{\mathcal{N}} := \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}\}$$

...

dove $\text{succ}_{\mathcal{N}} : \mathcal{N} \rightarrow \mathcal{N} : n \mapsto n \cup \{n\}$, soddisfano gli assiomi di Peano

Dimostrazione.

1. $0_{\mathcal{N}} \in \mathcal{N}$ per definizione stessa di \mathcal{N}
2. $n \in \mathcal{N} \implies \text{succ}_{\mathcal{N}}(n) \in \mathcal{N}$ per definizione stessa di $\text{succ}_{\mathcal{N}}$
3. Siano $n, m \in \mathcal{N}$ tali che $n \neq m$. In tal caso, ne segue automaticamente che:

$$n \neq m \implies n \cup \{n\} \neq m \cup \{m\} \iff \text{succ}_{\mathcal{N}}(n) \neq \text{succ}_{\mathcal{N}}(m)$$

Per contro-nominale, dunque, otteniamo che:

$$\text{succ}_{\mathcal{N}}(n) = \text{succ}_{\mathcal{N}}(m) \implies n = m$$

4. Supponiamo per assurdo che $\exists n \in \mathbb{N} \mid \text{succ}_{\mathcal{N}}(n) = 0_{\mathcal{N}}$. In tal caso, avremmo che:

$$\text{succ}(n) = 0_{\mathcal{N}} \iff n \cup \{n\} = 0_{\mathcal{N}} \iff n \cup \{n\} = \{\}$$

ma ciò risulta assurdo poiché implicherebbe che l'insieme $\{\}$ contenga degli elementi. Di conseguenza, l'unica possibilità è che $\nexists n \in \mathbb{N} \mid \text{succ}_{\mathcal{N}}(n) = 0_{\mathcal{N}}$

5. Supponiamo per assurdo che $\exists S \subseteq \mathcal{N} \mid (0_{\mathcal{N}} \in S \wedge (n \in S \implies \text{succ}_{\mathcal{N}}(n) \in S)) \wedge S \neq \mathcal{N}$. Consideriamo quindi $\mathcal{N} - S = \{n_1, \dots, n_k\}$. Per via del secondo assioma, ogni elemento di $\mathcal{N} - S$ deve avere un proprio successore e un proprio predecessore in \mathcal{N} .

Poiché per ipotesi si ha che $n \in S \implies \text{succ}_{\mathcal{N}}(n) \in S$, ne segue che tutti i predecessori degli elementi in $\mathcal{N} - S$ non possano essere in S , poiché altrimenti tali elementi sarebbero in S . Inoltre, poiché $\text{succ}_{\mathcal{N}}$ è iniettiva, ne segue che i successori degli elementi in $\mathcal{N} - S$ non possano essere in S , poiché esiste già un predecessore in S per ogni elemento in S .

Di conseguenza, ogni predecessore ed ogni successore degli elementi di $\mathcal{N} - S$ deve essere in $\mathcal{N} - S$ stesso. Consideriamo quindi (per comodità) la seguente catena di successori in $\mathcal{N} - S$:

$$n_1 \rightarrow n_2 \rightarrow \dots \rightarrow n_k \rightarrow n_1$$

Notiamo a questo punto che:

$$\text{succ}_{\mathcal{N}}^k(n_1) = n_1 \implies n_1 \in n_1$$

contraddicendo gli assiomi insiemistici per cui un insieme non possa essere contenuto in se stesso. Di conseguenza, l'unica possibilità è che $S = \mathcal{N}$

□

Principio 1: Principio di induzione

Sia P una proprietà che vale per $n = 0$. Dato $n \in \mathbb{N}$, se si verifica che la veridicità di P per n implica che P sia vera anche per $n + 1$, allora P vale per tutto \mathbb{N} . In simboli, abbiamo che:

$$\forall P ((P(0) \wedge (P(n) \implies P(n+1)))) \implies \forall m \in \mathbb{N} P(m)$$

Osservazione 1

Il quinto assioma di Peano è equivalente al principio di induzione, poiché basta considerare $S \subseteq \mathbb{N}$ come l'insieme degli elementi per cui vale la proprietà desiderata

Osservazione 2

Dato $k \in \mathbb{N}$, il principio di induzione può essere utilizzato per dimostrare che una proprietà P valga $\forall n \in \mathbb{N} \mid n \geq k$. In altre parole, non è necessario che il principio valga per tutti i naturali a partire da 0.

Dimostrazione.

- Definendo una proprietà Q tale che $P(n) = Q(n - k)$, si ha che:

$$\forall n - k \in \mathbb{N} \quad Q(n - k) \iff P(n)$$

dunque applicare il principio di induzione per P partendo da k equivale ad applicare il principio di induzione per Q partendo da 0, rispettando quindi il quinto assioma di Peano

□

Definizione 2: Insieme unità

Definiamo come **insieme unità**, indicato con $\mathbb{1}$, un insieme tale che $|\mathbb{1}| = 1$.

Definizione 3: Funzione unaria

Definiamo una funzione $f : \mathbb{1} \rightarrow S$, dunque avente $\mathbb{1}$ come dominio, come **funzione unaria**. Inoltre, per comodità, indichiamo $f(x)$ direttamente con f .

Esempio:

- Data la funzione zero : $\mathbb{1} \rightarrow \mathbb{N} : x \mapsto 0$, indichiamo zero(x) direttamente come zero

Osservazione 3

Una funzione unaria è sempre **iniettiva** in quanto esiste un solo elemento nel dominio.

Definizione 4: Algebra

Definiamo come **algebra** (o struttura algebrica) una n -upla $(A, \gamma_1, \dots, \gamma_n)$ dove A è un insieme non vuoto, detto **dominio**, e $\gamma_1, \dots, \gamma_n$ sono delle operazioni definite su A stesso

Esempi:

- La coppia $(\mathbb{N}, \text{succ})$ è un'algebra
- La coppia $(\mathbb{N}, \text{zero})$ è un'algebra

Definizione 5: Algebra induttiva e Costruttori

Definiamo l'algebra $(A, \gamma_1, \dots, \gamma_n)$ come **induttiva** se:

- $\gamma_1, \dots, \gamma_n$ sono iniettive
- $\forall i \neq j \quad \text{im}(\gamma_i) \cap \text{im}(\gamma_j) = \emptyset$, ossia le immagini delle operazioni sono due a due disgiunte
- $\forall S \subseteq A \quad (\forall i \in [1, n], a_1, \dots, a_k \in S \quad \gamma_i(a_1, \dots, a_k) \in S) \implies S = A$, ossia è soddisfatto il principio di induzione per ogni operazione

Inoltre, definiamo $\gamma_1, \dots, \gamma_n$ come **costruttori di A** .

Esempi:

- L'algebra $(\mathbb{N}, +)$ non è un'algebra induttiva poiché $+: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ non è iniettiva
- L'algebra $(\mathbb{N}, \text{succ}, \text{zero})$ è un'algebra induttiva poiché:
 - succ risulta essere iniettiva grazie al secondo assioma di Peano, mentre zero risulta essere iniettiva poiché funzione unaria
 - $\text{im}(\text{succ}) \cap \text{im}(\text{zero}) = (\mathbb{N} - \{0\}) \cap \{0\} = \emptyset$
 - Sia $S \subseteq \mathbb{N}$ tale che $\forall x \in S \quad \text{succ}(x) \in S$ e $\text{zero} \in S$. Preso $x \in \mathbb{N}$, possiamo esprimere x come $x = \text{succ}(\text{succ}(\dots(\text{zero})))$.

Di conseguenza, poiché S è chiuso per succ e zero , otteniamo che:

- * $\text{zero} \in S \implies \text{succ}(\text{zero}) \in S$
- * $\text{succ}(\text{zero}) \in S \implies \text{succ}(\text{succ}(\text{zero})) \in S$
- * ...
- * $\text{succ}(\dots(\text{zero})) \in S \implies x = \text{succ}(\text{succ}(\dots(\text{zero}))) \in S$

Di conseguenza, otteniamo che $A \subseteq S$ e dunque che $S = A$

Osservazione 4

Equivalentemente, la terza condizione necessaria delle algebre induttive può essere considerata come

$$\nexists S \subsetneq A \mid (S, \gamma_1, \dots, \gamma_n) \text{ è algebra induttiva}$$

Definizione 6: Omomorfismo

Date due strutture algebriche $(A, \gamma_1, \dots, \gamma_k)$ e $(B, \delta_1, \dots, \delta_k)$ dello stesso tipo, definiamo $f : A \rightarrow B$ come **omomorfismo** se

$$\forall a_1, \dots, a_n \in A, i \in [1, k] \quad f(\gamma_i(a_1, \dots, a_k)) = \delta_i(f(a_1), \dots, f(a_k))$$

Esempi:

- Date le due algebre $(\mathbb{N}, \text{succ}, +)$ e $(\mathcal{N}, \text{succ}_{\mathcal{N}}, +_{\mathcal{N}})$, affinché la funzione $f : \mathbb{N} \rightarrow \mathcal{N}$ sia un omomorfismo è necessario che:

$$f(\text{succ}(n)) = \text{succ}_{\mathcal{N}}(f(n)) \quad f(n + m) = f(n) +_{\mathcal{N}} f(m)$$

- Date le due algebre $(\mathbb{R}, +)$ e $(\mathbb{R}_{>0}, \cdot)$, la funzione $\exp : \mathbb{R} \rightarrow \mathbb{R}_{>0} : x \mapsto e^x$ è un omomorfismo:

$$\exp(x + y) = e^{x+y} = e^x e^y = \exp(x) \exp(y)$$

Definizione 7: Isomorfismo

Definiamo come **isomorfismo** un omomorfismo biiettivo

Osservazione 5

Data una funzione $f : A \rightarrow B$, si ha che:

$$f \text{ è biettiva} \iff \exists f^{-1} : B \rightarrow A$$

(*dimostrazione omessa*)

Osservazione 6

Data una funzione $f : A \rightarrow B$, si ha che:

$$f \text{ è un isomorfismo} \iff f^{-1} \text{ è un isomorfismo}$$

(*dimostrazione omessa*)

Esempio:

- Date le due algebre $(\mathbb{R}, +)$ e $(\mathbb{R}_{>0}, \cdot)$, la funzione $\exp : \mathbb{R} \rightarrow \mathbb{R}_{>0} : x \mapsto e^x$ è un isomorfismo, poiché:
 - \exp è un omomorfismo
 - $\exists \ln : \mathbb{R}_{>0} \rightarrow \mathbb{R} \mid \ln(\exp(x)) = x$, dunque f è biettiva.

1.2 Strutture dati induttive

Definizione 8: Insieme delle liste finite

Definiamo $\text{List}\langle T \rangle$ come l'insieme delle liste finite di elementi di T

Esempio:

- Dato $\text{List}\langle \text{Int} \rangle$, si ha che $[3 \rightarrow 5 \rightarrow 1] \in \text{List}\langle \text{Int} \rangle$

Proposizione 2: Algebra induttiva delle liste finite

La tripla $(\text{List}\langle T \rangle, \text{empty}, \text{cons})$, dove:

- $\text{empty} : \mathbb{1} \rightarrow \text{List}\langle T \rangle : x \mapsto []$ è la funzione unaria che restituisce la **lista vuota**
- $\text{cons} : \text{List}\langle T \rangle \times T \rightarrow \text{List}\langle T \rangle : x, ([x_1 \rightarrow \dots \rightarrow x_n]) \mapsto [x \rightarrow x_1 \rightarrow \dots \rightarrow x_n]$ è la funzione di **costruzione delle liste**

è un'algebra induttiva

Dimostrazione.

1. La funzione empty risulta essere iniettiva poiché unaria.

Dati $\ell_1, \ell_2 \in \text{List}\langle T \rangle$ e $x_1, x_2 \in T$, supponiamo che:

$$\text{cons}(y_1, \ell_1) = \text{cons}(y_2, \ell_2) = [x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n]$$

Per definizione stessa di cons , si ha che:

$$\begin{aligned} \text{cons}(y_1, \ell_1) &= \text{cons}(y_2, \ell_2) = [x \rightarrow x_1 \rightarrow \dots \rightarrow x_n] \\ \implies y_1 &= y_2 = x, \ell_1 = \ell_2 = [x_1 \rightarrow \dots \rightarrow x_n] \end{aligned}$$

dunque anche cons risulta iniettiva

2. $\text{im}(\text{empty}) \cap \text{im}(\text{cons}) = \{[]\} \cap (\text{List}\langle T \rangle - \{[]\}) = \emptyset$

3. Sia $S \subseteq \text{List}\langle T \rangle$ tale che $\forall x \in T, \ell \in \text{List}\langle T \rangle \text{ } \text{cons}(x, \ell) \in S$ e $\text{empty} \in S$.

Preso $\ell := [x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n] \in \text{List}\langle T \rangle$, possiamo esprimere ℓ come

$$\ell = \text{cons}(x_1, \text{cons}(x_2, \dots \text{cons}(x_n, \text{empty})))$$

Di conseguenza, poiché S è chiuso per cons e empty e poiché $\text{empty} \in S$, otteniamo che ogni valore della catena sia contenuto in S , implicando che $x \in S$ e quindi che $\text{List}\langle T \rangle \subseteq S$, concludendo che $S = \text{List}\langle T \rangle$

□

Osservazione 7

La tripla $(\text{List}\langle T \rangle_\infty, \text{empty}, \text{cons})$, dove $\text{List}\langle T \rangle_\infty$ è l'insieme delle liste infinite di elementi di T **non è un'algebra induttiva**, poiché $\text{List}\langle T \rangle \subsetneq \text{List}\langle T \rangle_\infty$ e poiché $(\text{List}\langle T \rangle, \text{empty}, \text{cons})$ è un'algebra induttiva

Osservazione 8

Tramite i costruttori di un'algebra induttiva è possibile definire le ulteriori operazioni "aggiuntive" di tale algebra

Esempio:

- Data l'algebra induttiva $(\text{List}\langle T \rangle, \text{empty}, \text{cons})$, definiamo la seguente operazione

$$\text{concat} : \text{List}\langle T \rangle \times \text{List}\langle T \rangle \rightarrow \text{List}\langle T \rangle$$

dove:

$$\begin{cases} \text{concat}(\text{empty}, \ell) = \ell \\ \text{concat}(\text{cons}(n, \ell), \ell') = \text{cons}(n, \text{concat}(\ell, \ell')) \end{cases}$$

- Ad esempio, in $\text{List}\langle \text{Int} \rangle$, abbiamo che:

$$\begin{aligned} \text{concat}([1 \rightarrow 5], [7 \rightarrow 2]) &= \text{concat}(\text{cons}(1, [5], [7 \rightarrow 2])) = \text{cons}(1, \text{concat}([5], [7 \rightarrow 2])) = \\ &= \text{cons}(1, \text{concat}(\text{cons}(5, \text{empty}), [7 \rightarrow 2])) = \text{cons}(1, \text{cons}(5, \text{concat}(\text{empty}, [7 \rightarrow 2]))) = \\ &= \text{cons}(1, \text{cons}(5, [7 \rightarrow 2])) = \text{cons}(1, [5 \rightarrow 7 \rightarrow 2]) = [1 \rightarrow 5 \rightarrow 7 \rightarrow 2] \end{aligned}$$

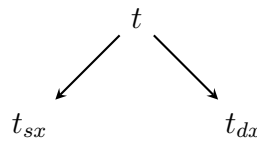
Definizione 9: Insieme degli alberi binari finiti

Definiamo **BinTree** come l'insieme degli alberi binari finiti

Proposizione 3: Algebra induttiva degli alberi binari finiti

La tripla $(\text{BinTree}, \text{leaf}, \text{branch})$, dove:

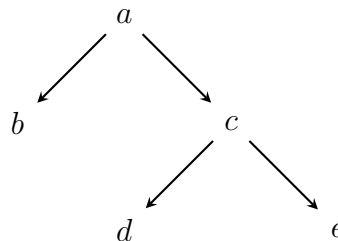
- $\text{leaf} : \mathbb{1} \rightarrow \text{BinTree} : x \mapsto \circ$ è la funzione unaria che restituisce una **foglia**
- $\text{branch} : \text{BinTree} \times \text{BinTree} \rightarrow \text{BinTree} : (t_{sx}, t_{dx}) \mapsto t$ è la funzione di **costruzione dei rami**, ossia tale che



è un'algebra induttiva
(*dimostrazione omessa*)

Esempio:

- Il seguente albero



corrisponde a:

$$a = \text{branch}(\text{leaf}, \text{branch}(\text{leaf}, \text{leaf}))$$

Definizione 10: Induzione strutturale

Definiamo come **induzione strutturale** il metodo dimostrativo generalizzante il principio di induzione e basato sulle proprietà di un'algebra induttiva.

In particolare, viene ipotizzato che una proprietà P valga per ogni argomento di ogni costruttore dell'algebra e tramite il terzo assioma viene dimostrato che tale proprietà valga per tutti gli elementi dell'algebra stessa

Teorema 1: Relazione tra nodi e foglie

Dato $t \in \text{BinTree}$ avente n foglie, il numero di nodi di t è pari a $2n - 1$

Dimostrazione per induzione strutturale.

- Definiamo l'operazione

$$\text{leaves} : \text{BinTree} \rightarrow \mathbb{N} : t \mapsto \text{Numero di foglie in } t$$

dove:

$$\begin{cases} \text{leaves}(\text{leaf}) = 1 \\ \text{leaves}(\text{branch}(b_1, b_2)) = \text{leaves}(b_1) + \text{leaves}(b_2) \end{cases}$$

- Dato $t \in \text{BinTree}$, sia k il numero di nodi di t e sia $n = \text{leaves}(t)$

Caso base. Se $t = \text{leaf}$, allora t è composto da $k = 1$ nodi e $n = \text{leaves}(\text{leaf}) = 1$ foglie. Difatti, si ha che:

$$k = 1 = 2n - 1$$

Ipotesi induttiva. Ogni argomento t' di ogni costruttore possiede $k' = 2\text{leaves}(t') - 1$ nodi

Passo induttivo. Se $t \neq \text{leaf}$, allora $\exists t_1, t_2 \in \text{BinTree} \mid t = \text{branch}(t_1, t_2)$ dove t_1 e t_2 possiedono rispettivamente k_1 e k_2 nodi. Inoltre, si ha che:

$$k = k_1 + k_2 + 1$$

In quanto t_1 e t_2 sono argomenti del costruttore branch , per ipotesi induttiva si ha che:

$$\begin{aligned} k &= k_1 + k_2 + 1 = 2\text{leaves}(t_1) - 1 + 2\text{leaves}(t_2) - 1 + 1 = 2(\text{leaves}(t_1) + \text{leaves}(t_2)) - 1 = \\ &= 2(\text{leaves}(\text{branch}(t_1, t_2))) - 1 = 2(\text{leaves}(t)) - 1 \end{aligned}$$

□

1.3 Sintassi astratta

Definizione 11: Linguaggio

Definiamo come **linguaggio** un insieme di stringhe

Definizione 12: Grammatica

Definiamo come **grammatica** un insieme di regole, dette **termini**, che definiscono come poter manipolare le stringhe di un linguaggio.

La **forma di Backus-Naur** è una notazione utilizzata per descrivere grammatiche ed è definita come:

$$\langle \text{symbol} \rangle ::= _ \text{expression} _$$

dove:

- $\langle \text{symbol} \rangle$ è un simbolo non-terminale espresso dalla grammatica
- L'operatore $::=$ indica che ciò che si trova alla sua sinistra possa essere sostituito con ciò che si trova alla sua destra
- $_ \text{expression} _$ consiste in una o più sequenze di simboli terminali o non-terminali dove ogni sequenza è separata da una barra verticale (ossia $|$) indicante una scelta possibile per l'operatore $::=$

Esempio:

- Consideriamo il linguaggio L espresso dalla grammatica:

$$M, N ::= 0 \mid 1 \mid \dots \mid M + N \mid M * N$$

Tale grammatica indica che i simboli non-terminali M e N possono essere sostituiti con:

- Un numero naturale
- Un'espressione $M + N$ o $M * N$ dove M e N sono due ulteriori simboli terminali o non-terminali
- Ad esempio, abbiamo che la stringa " $5 + 7$ " sia ben definita dalla grammatica, mentre la stringa " $5 + +$ " non lo sia

Definizione 13: Sintassi astratta

La **sintassi astratta** di un linguaggio è una definizione induttiva di un insieme T di termini, permettendo di definire strutture algebriche senza dover necessariamente definire concretamente le sue operazioni

Esempio:

- Consideriamo ancora il linguaggio L definito dalla grammatica

$$M, N ::= 0 \mid 1 \mid \dots \mid M + N \mid M * N$$

- Definiamo quindi la funzione $\text{eval} : L \rightarrow \mathbb{N}$ in grado di valutare le espressioni del linguaggio:

$$\text{eval}("0") = 0$$

$$\text{eval}("1") = 1$$

...

$$\text{eval}("M + N") = \text{eval}("M") + \text{eval}("N")$$

$$\text{eval}("M * N") = \text{eval}("M") + \text{eval}("N")$$

- Notiamo quindi che la grammatica definisca in modo astratto (ma concretamente tramite eval) le seguenti operazioni:

$$0 : \mathbb{1} \rightarrow \mathbb{N} : x \mapsto 0$$

$$1 : \mathbb{1} \rightarrow \mathbb{N} : x \mapsto 1$$

...

$$\text{plus} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} : (m, n) \mapsto m + n$$

$$\text{times} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} : (m, n) \mapsto m \cdot n$$

- Notiamo però che le operazioni plus e times non risultano essere né iniettive né con immagini disgiunte. Di conseguenza, la funzione eval non ci permette di definire un'algebra induttiva.
- Tuttavia, per tale linguaggio è comunque possibile definire (in qualche modo, ad esempio fissando una precedenza per le operazioni rompendo proprietà come l'associatività e la commutatività) una funzione che possa descrivere un'algebra induttiva.

Teorema 2: Algebra dei termini

Dato un linguaggio L rappresentato da una sintassi astratta per termini definiti in T , esiste sempre un'algebra induttiva (T, α)

(*dimostrazione omessa*)