"Sapienza" Università di Roma
Ingegneria dell'Informazione,
Informatica e Statistica
Dipartimento di Informatica

# Linguaggi di Programmazione

*Author*
Simone Bianco

7 ottobre 2023

# Indice

# 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

# Struttura e Rappresentazione

## 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 $\text{succ} : \mathbb{N} \to \mathbb{N}$ 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. $\forall S \subseteq \mathbb{N} \mid (0 \in S \land (n \in S \implies \text{succ}(n) \in S)) \implies S = \mathbb{N}$

**Proposizione 1: Numeri naturali di Von Neumann**

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

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

$$1_{\mathcal{N}} := \{\,\{\}\,\}$$
$$2_{\mathcal{N}} := \{\,\{\}, \{\{\}\}\,\}$$
$$3_{\mathcal{N}} := \{\,\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}\,\}$$
$$\ldots$$

dove $\text{succ}_{\mathcal{N}} : \mathcal{N} \to \mathcal{N} : n \mapsto n \cup \{n\}$, 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:

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

   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 $\exists S \subseteq \mathcal{N} \mid (0_{\mathcal{N}} \in S \land (n \in S \implies \text{succ}_{\mathcal{N}}(n) \in S)) \land S \neq \mathcal{N}$. 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 $\mathcal{N} - S$:

$$n_1 \to n_2 \to \dots \to n_k \to n_1$$

   Notiamo a questo punto che:

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

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

$\square$

**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:

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

  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** l'insieme $\mathbb{1} = \{()\}$, ossia l'insieme composto da una zerupla

**Definizione 3: Funzione nullaria**

Definiamo una funzione $f : \mathbb{1} \to S$, dunque avente $\mathbb{1}$ come dominio, come **funzione nullaria** (o funzione costante).

Inoltre, per comodità, indichiamo $f(x)$ direttamente con $f$, poiché $x = ()$

**Esempio:**

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

**Osservazione 3**

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

**Definizione 4: Segnatura di una funzione**

Data una funzione $f$ definiamo $f : D \to C$ come **segnatura di** $f$ dove $D$ è il **dominio di** $f$ e $C$ è il **codominio di** $f$

**Definizione 5: Algebra**

Definiamo come **algebra** (o struttura algebrica) una n-upla $(A, \gamma_1, \ldots, \gamma_n)$ dove $A$ è un insieme non vuoto, detto **dominio**, e $\gamma_1, \ldots, \gamma_n$ 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 6: Segnatura di un'algebra**

Data un'algebra $(A, \gamma_1, \ldots, \gamma_n)$, definiamo come **segnatura dell'algebra** l'insieme delle segnature delle operazioni definite su essa

**Definizione 7: Segnature equivalenti**

Date due algebre $(A, \gamma_1, \ldots, \gamma_n)$ e $(B, \delta_1, \ldots, \delta_n)$, definiamo le segnature di tali algebre come **equivalenti** se per ogni operazione $\gamma$ definita su $A$ esiste un'operazione $\delta$ definita su $B$ per cui invertendo $B$ con $A$ all'interno della segnatura di $\delta$ si ottiene la segnatura di $\gamma$

**Esempio:**

- Date le due algebre $(\mathbb{N}, \text{zero}, \text{succ})$ e $(\mathcal{N}, \text{zero}_\mathcal{N}, \text{succ}_\mathcal{N})$, le segnature di tali algebre sono equivalenti poiché:
  - La segnatura di $\text{zero} : \mathbb{1} \to \mathbb{N}$ è equivalente alla segnatura di $\text{zero}_\mathcal{N} : \mathbb{1} \to \mathcal{N}$
  - La segnatura di $\text{succ} : \mathbb{N} \to \mathbb{N}$ è equivalente alla segnatura di $\text{succ}_\mathcal{N} : \mathcal{N} \to \mathcal{N}$

**Definizione 8: Algebra induttiva e Costruttori**

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

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

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

**Esempi:**

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

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

    * $\mathrm{zero} \in S \implies \mathrm{succ}(\mathrm{zero}) \in S$
    * $\mathrm{succ}(\mathrm{zero}) \in S \implies \mathrm{succ}(\mathrm{succ}(\mathrm{zero})) \in S$
    * $\ldots$
    * $\mathrm{succ}(...(\mathrm{zero})) \in S \implies x = \mathrm{succ}(\mathrm{succ}(...(\mathrm{zero}))) \in S$

    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

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

**Definizione 9: Omomorfismo**

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

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

**Esempi:**

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

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

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

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

**Definizione 10: Isomorfismo**

Definiamo come **isomorfismo** un omomorfismo biettivo. Inoltre, definiamo due algebre $(A, \gamma_1, \ldots, \gamma_n)$, $(B, \delta_1, \ldots, \delta_n)$ come **isomorfe**, indicato con $A \cong B$, se esiste un isomorfismo tra loro.

**Osservazione 5**

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

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

(*dimostrazione omessa*)

**Osservazione 6**

Data una funzione $f : A \to B$, 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 $\exp : \mathbb{R} \to \mathbb{R}_{>0} : x \mapsto e^x$ è un isomorfismo, poiché:
  - exp è un omomorfismo
  - $\exists \ln : \mathbb{R}_{>0} \to \mathbb{R} \mid \ln(\exp(x)) = x$, dunque $f$ è biettiva.

**Lemma 1**

Data un'algebra induttiva $(A, \gamma_1, \ldots, \gamma_n)$, per ogni algebra $(B, \delta_1, \ldots, \delta_n)$ con la stessa segnatura di $A$ si ha che

$$\exists! \text{ omomorfismo } f : A \to B$$

**Nota:** l'algebra di $B$ non deve necessariamente essere induttiva

(*dimostrazione omessa*)

**Teorema 1: Lemma di Lambek (versione ridotta)**

Date due algebre induttive $(A, \gamma_1, \ldots, \gamma_n)$ e $(B, \delta_1, \ldots, \delta_n)$ si ha che $A \cong B$

*Dimostrazione.*

- Per il lemma precedente, si ha che:

$$\exists! \text{ omomorfismo } f : A \to B$$

$$\exists! \text{ omomorfismo } g : B \to A$$

- Consideriamo quindi la funzione $g \circ f : A \to Axg(f(x))$ e verifichiamo che essa sia un omomorfismo

$$g \circ f(x+y) = g(f(x+y)) = g(f(x)+f(y)) = g(f(x))+g(f(y)) = g \circ f(x)+g \circ f(y)$$

- Tuttavia, per ogni algebra esiste sempre l'isomorfismo identità $\text{id} : A \to A : x \mapsto x$ e poiché per il lemma precedente esiste necessariamente un unico omomorfismo tra $A$ e $A$, ne segue necessariamente che $g \circ f = \text{id}$
- Di conseguenza, si ha che

$$g \circ f = \text{id} \iff g = f^{-1} \implies g, f \text{ biettive} \implies g, f \text{ isomorfismi} \implies A \cong B$$

□

**Esempio:**

- Date le due algebre induttive $(\mathbb{N}, \text{zero}, \text{succ})$ e $(\mathcal{N}, \text{zero}_{\mathcal{N}}, \text{succ}_{\mathcal{N}})$ sono isomorfe tra loro poiché aventi la stessa segnatura algebrica
- Difatti, come già dimostrato, $\mathbb{N}$ e $\mathcal{N}$ sono solamente due modi diversi per rappresentare lo stesso identico concetto algebrico

## 1.2 Strutture dati induttive

**Definizione 11: Insieme delle liste finite**

Definiamo `List<T>` come l'insieme delle liste finite di elementi di $T$

**Esempio:**

- Dato `List<Int>`, si ha che $[3 \to 5 \to 1] \in$ `List<Int>`

**Proposizione 2: Algebra induttiva delle liste finite**

La tripla $(\texttt{List<T>}, \text{empty}, \text{cons})$, dove:

- $\text{empty} : \mathbb{1} \to \texttt{List<T>} : x \mapsto [\ ]$ è la funzione nullaria che restituisce la **lista vuota**
- $\text{cons} : \texttt{List<T>} \times T \to \texttt{List<T>} : x, ([x_1 \to \ldots \to x_n]) \mapsto [x \to x_1 \to \ldots x_n]$ è la funzione di **costruzione delle liste**

è un'algebra induttiva

*Dimostrazione.*

1. La funzione empty risulta essere iniettiva poiché nullaria.

   Dati $\ell_1, \ell_2 \in \texttt{List<T>}$ e $x_1, x_2 \in T$, supponiamo che:

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

   Per definizione stessa di cons, si ha che:

   $$\text{cons}(y_1, \ell_1) = \text{cons}(y_2, \ell_2) = [x \to x_1 \to \ldots \to x_n]$$

   $$\implies y_1 = y_2 = x, \ell_1 = \ell_2 = [x_1 \to \ldots \to x_n]$$

   dunque anche cons risulta iniettiva

2. $\text{im}(\text{empty}) \cap \text{im}(\text{cons}) = \{[\ ]\} \cap (\texttt{List<T>} - \{[\ ]\}) = \varnothing$

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

   Preso $\ell := [x_1 \to x_2 \to \ldots \to x_n] \in \texttt{List<T>}$, possiamo esprimere $\ell$ come

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

   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 $\texttt{List<T>} \subseteq S$, concludendo che $S = \texttt{List<T>}$

□

**Osservazione 7**

La tripla $(\texttt{List<T>}_\infty, \text{empty}, \text{cons})$, dove $\texttt{List<T>}_\infty$ è l'insieme delle liste infinite di elementi di $T$ **non è un'algebra induttiva**, poiché $\texttt{List<T>} \subsetneq \texttt{List<T>}_\infty$ e poiché $(\texttt{List<T>}, \text{empty}, \text{cons})$ 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 $(\texttt{List<T>}, \text{empty}, \text{cons})$, definiamo la seguente operazione

$$\text{concat} : \texttt{List<T>} \times \texttt{List<T>} \to \texttt{List<T>}$$

  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 $\texttt{List<Int>}$, abbiamo che:

$$\text{concat}([1 \to 5], [7 \to 2]) = \text{concat}(\text{cons}(1, [5], [7 \to 2])) = \text{cons}(1, \text{concat}([5], [7 \to 2])) =$$

$$\text{cons}(1, \text{concat}(\text{cons}(5, \text{empty}), [7 \to 2])) = \text{cons}(1, \text{cons}(5, \text{concat}(\text{empty}, [7 \to 2]))) =$$

$$\text{cons}(1, \text{cons}(5, [7 \to 2])) = \text{cons}(1, [5 \to 7 \to 2]) = [1 \to 5 \to 7 \to 2]$$

**Definizione 12: Insieme degli alberi binari finiti**

Definiamo $\texttt{BinTree}$ come l'insieme degli alberi binari finiti

**Proposizione 3: Algebra induttiva degli alberi binari finiti**

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

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

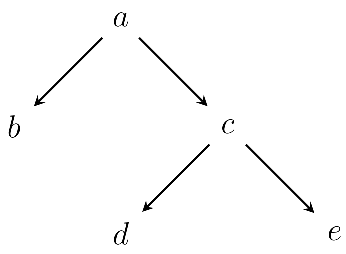
$$\begin{tikzcd} & t \arrow[dl] \arrow[dr] & \\ t_{sx} & & t_{dx} \end{tikzcd}$$

è un'algebra induttiva

(*dimostrazione omessa*)

**Esempio:**

- Il seguente albero




corrisponde a:

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

**Definizione 13: 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 2: Relazione tra nodi e foglie**

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

*Dimostrazione per induzione strutturale.*

- Definiamo l'operazione

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

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 $t \in \texttt{BinTree}$, 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 $n = \textbf{leaves}(\text{leaf}) = 1$ foglie. Difatti, si ha che:

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

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

*Passo induttivo.* Se $t \neq \text{leaf}$, allora $\exists t_1, t_2 \in \texttt{BinTree} \mid t = \text{branch}(\text{t}_1, \text{t}_2)$ 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:

$$k = k_1 + k_2 + 1 = 2\text{leaves}(t_1) - 1 + 2\text{leaves}(t_2) - 1 + 1 = 2(\text{leaves}(t_1) + \text{leaves}(t_2)) - 1 =$$

$$= 2(\text{leaves}(\text{branch}(t_1, t_2))) - 1 = 2(\text{leaves}(t)) - 1$$

□

## 1.3 Sintassi astratta

**Definizione 14: Linguaggio**

Definiamo come **linguaggio** un insieme di stringhe

**Definizione 15: 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:

`<symbol>` ::= `_expression_`

dove:

- `<symbol>` è una 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
- `<_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 16: 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 $\mathrm{eval} : L \to \mathbb{N}$ in grado di valutare le espressioni del linguaggio:

$$\begin{aligned}
\mathrm{eval}(\texttt{"0"}) &= 0 \\
\mathrm{eval}(\texttt{"1"}) &= 1 \\
&\dots \\
\mathrm{eval}(\texttt{"M + N"}) &= \mathrm{eval}(\texttt{"M"}) + \mathrm{eval}(\text{"}N\text{"}) \\
\mathrm{eval}(\texttt{"M * N"}) &= \mathrm{eval}(\texttt{"M"}) + \mathrm{eval}(\text{"}N\text{"})
\end{aligned}$$

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

$$\begin{aligned}
0 &: \mathbb{1} \to \mathbb{N} : x \mapsto 0 \\
1 &: \mathbb{1} \to \mathbb{N} : x \mapsto 1 \\
&\dots \\
\mathrm{plus} &: \mathbb{N} \times \mathbb{N} \to \mathbb{N} : (m, n) \mapsto m + n \\
\mathrm{times} &: \mathbb{N} \times \mathbb{N} \to \mathbb{N} : (m, n) \mapsto m \cdot n
\end{aligned}$$

- 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 3: Algebra dei termini**

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

(*dimostrazione omessa*)

# 2

# Paradigma funzionale

## 2.1 $Exp$: un semplice linguaggio funzionale

**Definizione 17: Il linguaggio $Exp$**

Definiamo come $Exp$ il linguaggio rappresentato dalla seguente grammatica:

$$M, N ::= k \ \mid x \ \mid M + N \ \mid let\ x = M \text{ in } N$$

dove:

- $k \in Val = \{0, 1, \ldots\}$ ossia è una costante
- $x \in Var = \{x, y, z, \ldots\}$ ossia è una variabile
- $+ : Exp \times Exp \to$ la quale somma le due espressioni
- $let : Var \times Exp \times Exp \to Exp$ la quale assegna alla variabile $x$ l'espressione $M$ all'interno della valutazione di $N$. Inoltre, $x$ prende il nome di variabile locale all'interno di $N$

**Esempi:**

- L'espressione $let\ x = 3 \text{ in } x + 1$ indica che la variabile $x$ assuma valore 3 all'interno della valutazione di $x + 1$. Di conseguenza, il risultato della valutazione dell'espressione è 4
- L'espressione $let\ x = 3 \text{ in } 7$ viene valutata come 7
- L'espressione $let\ y = 9 \text{ in } (let\ x = (let\ y = 2 \text{ in } y + 1) \text{ in } x + y)$ viene valutata come 12 (si consiglia di cercare di capire come le clausole interne sovrascrivano i valori delle clausole esterne. Se ciò risultasse complesso, più avanti verranno forniti strumenti matematici per valutare in modo corretto le clausole *let* annidate)

**Definizione 18: Scope di una variabile**

Data un'espressione e una variabile $x$, definiamo come **scope di** $x$ la porzione la porzione dell'espressione all'interno della quale una variabile può essere riferita, ossia per cui ne è definito il valore.

Una variabile il cui valore non è assegnato in una porzione dell'espressione viene detta **variabile libera**

**Definizione 19: Variabile libera**

Data un'espressione $expr \in Exp$, definiamo $x \in expr$ come **libera** se $x$ non ha un valore assegnato durate la valutazione di $expr$.

**Esempio:**

- L'espressione $let\ x = (let\ y = 2 \text{ in } y + 1) \text{ in } x + y$ non è coerente con la grammatica di $Exp$, poiché $y$ non è definito durante la valutazione di $x + y$. Di conseguenza, non è possibile valutare tale espressione.

**Proposizione 4: Calcolo delle variabili libere**

Dato il linguaggio $Exp$, la funzione

$$\text{free} : Exp \to \mathcal{P}(Var)$$

restituisce l'insieme di tutte le **variabili libere** di un'espressione dove:

$$\begin{cases} \text{free}(k) = \varnothing \\ \text{free}(x) = \{x\} \\ \text{free}(M + N) = \text{free}(M) \cup \text{free}(N) \\ \text{free}(let\ x = M \text{ in } N) = \text{free}(M) \cup (\text{free}(N) - \{x\}) \end{cases}$$

**Nota**: $\mathcal{P}(Var)$ è l'insieme delle parti di $Var$, ossia l'insieme contenente tutti i suoi sottoinsiemi possibili

**Esempio:**

- Riprendendo l'esempio precedente, notiamo che:

$$\text{free}(let\ x = (let\ y = 2 \text{ in } y + 1) \text{ in } x + y) =$$

$$= \text{free}(let\ y = 2 \text{ in } y + 1) \cup (\text{free}(x + y) - \{x\}) =$$
$$= \text{free}(let\ y = 2 \text{ in } y + 1) \cup ((\text{free}(x) \cup \text{free}(y)) - \{x\}) =$$
$$= \text{free}(let\ y = 2 \text{ in } y + 1) \cup ((\{x\} \cup \{y\}) - \{x\}) =$$
$$= \text{free}(let\ y = 2 \text{ in } y + 1) \cup \{y\} =$$

$$= (\text{free}(2) \cup (\text{free}(y+1) - \{y\})) \cup \{y\} =$$
$$= ((\text{free}(y)) - \{y\}) \cup \{y\} =$$
$$= \{y\}$$

dunque l'espressione è invalutabile

**Definizione 20: Insieme degli ambienti**

Dato il linguaggio *Exp*, definiamo come **insieme degli ambienti di** *Exp*, indicato con *Env*, l'insieme delle funzioni parziali (ossia non necessariamente definite su tutto il dominio) che associano ogni variabile al proprio valore:

$$Env = \{f \mid f : Var \xrightarrow{fin} v\}$$

**Definizione 21: Concatenazione di ambienti**

Dato il linguaggio *Exp*, definiamo l'operazione di **concatenazione di ambienti**, ossia:

$$\cdot : Env \times Env \to Env$$

dove:

$$(E_1 E_2)(x) = \begin{cases} E_2(x) & \text{se } x \in dom(E_1) \\ E_1(x) & \text{altrimenti} \end{cases}$$

**Nota**: tale operazione può essere interpretata come una sovrascrittura in $E_1$ di tutte le variabili definite in $E_2$

**Esempio:**

- Dati gli ambienti $E_1 = \{(x, 4), (y, 3)\}$ e $E_2 = \{(x, 5)\}$, si ha che

$$(E_1 E_2)(x) = 5$$

$$(E_1 E_2)(y) = 3$$

**Proposizione 5: Regola di inferenza**

Data la proposizione:

$$\text{Premessa 1} \wedge \ldots \wedge \text{Premessa n} \implies \text{Conclusione}$$

definiamo come **regola di inferenza** la notazione alternativa:

$$\frac{\text{Premessa 1} \quad \ldots \quad \text{Premessa n}}{\text{Conclusione}}$$

**Definizione 22: Giudizio operazionale**

Data la relazione di *semantica operazionale*, ossia:

$$\leadsto \subseteq Env \times Exp \times Val$$

definiamo come **giudizio operazionale** la tripla $(E, M, v) \in \leadsto$ descritta dalla notazione

$$E \vdash M \leadsto v$$

la quale viene letta come "nell'ambiente $E$, $M$ viene valutato come $v$".

Tale relazione gode del seguente insieme di regole:

- $E \vdash k \leadsto k$
- Se $E(x) = v$, ossia se $x$ è definita in $E$, allora $E \vdash x \leadsto v$
- Se $u = v + v'$, allora:
$$\frac{E \vdash M \leadsto v \quad E \vdash N \leadsto v'}{E \vdash M + N \leadsto u}$$
- Vale che:
$$\frac{E \vdash M \leadsto v \quad E\{(x, v)\} \vdash N \leadsto v'}{E \vdash let\ x = M \text{ in } N \leadsto v'}$$

**Esempio:**

- L'espressione $let\ y = k$ in $(let\ x = k'$ in $x + y)$ segue la regola:

$$\cfrac{E \vdash k \leadsto k \qquad \cfrac{E\{y, k\} \vdash k' \leadsto k' \qquad \cfrac{E\{(y,k)\}\{x, k'\} \vdash x \leadsto v' \quad E\{(y,k),(x,k')\} \vdash y \leadsto v''}{E\{(y,k)\}\{x, k'\} \vdash x + y \leadsto v}}{E\{(y,k)\} \vdash let\ x = k' \text{ in } x + y \leadsto v}}{E \vdash let\ y = k \text{ in } (let\ x = k' \text{ in } x + y) \leadsto v}$$

  dove $v = v' + v''$

- Per valutare l'intera espressione, in questo caso, ci basta in realtà valutare i termini "più in alto":
  - $E\{(y,k)\}\{x, k'\} \vdash x \leadsto v' \implies v' = k'$
  - $E\{(y,k)\}\{x, k'\} \vdash y \leadsto v'' \implies v'' =$
- Di conseguenza, essendo l'output $v$ e poiché dall'inferenza otteniamo che $v = v' + v''$, ne segue automaticamente che l'espressione totale venga valutata in $v = k' + k''$