

# Capitolo 1

## Induzione strutturale

### Introduzione

Sia  $A$  un insieme, sia  $\triangleleft$  una relazione binaria definita su  $A$  ( $\triangleleft : A \times A$ ).

**Def.**  $\triangleleft$  è b.f. (ben fondata) se  $\nexists \dots \triangleleft a_i \triangleleft \dots \triangleleft a_1 \triangleleft a_0$   
ovvero non esistono catene infinite discendenti.  
( $A, \triangleleft$ ) è un insieme ben fondato se  $\triangleleft$  è b.f.

Sia  $\trianglelefteq$  la chiusura riflessiva e transitiva di  $\triangleleft$  b.f. su  $A$ .

**Lemma 1.**  $\trianglelefteq$  non è mai b.f.

*Dimostrazione.* Sia  $a_i \in A$ .  $\exists \dots a_i \trianglelefteq a_i \trianglelefteq \dots$  □

**Def.** Sia  $a \in A$ .  $a$  è *minimale* in  $A$  rispetto a  $\triangleleft$  se  $\forall b \triangleleft a, b \notin A$ .

**Lemma 2.**  $\triangleleft$  è ben fondato su  $A$  se e solo se ogni  $B \subseteq A$  ha un elemento minimale rispetto a  $\triangleleft$ .

*Dimostrazione.* •  $\Rightarrow$ ) Da  $B \subseteq A$  e  $(A, \triangleleft)$  b.f. si ha che non esiste  $\dots \triangleleft b_i \triangleleft \dots \triangleleft b_0$ , quindi  $\exists b_n$  t.c.  $b_n \triangleleft \dots \triangleleft b_i \triangleleft \dots \triangleleft b_0$  ovvero  $b_n$  è minimale in  $B$  rispetto a  $\triangleleft$ .

- $\Leftarrow$ ) Per assurdo. Supponiamo che esista  $\dots \triangleleft a_i \triangleleft \dots \triangleleft a_0$ , ovvero  $\neg(\triangleleft \text{ b.f. })$ . Allora l'insieme  $B = \{a_i | i \in \mathbb{N}\}$  (insieme degli elementi della sequenza) non ha un minimale, cosa che contraddice l'ipotesi. Quindi una tale sequenza non esiste, ovvero  $\triangleleft$  b.f. □

## 1.1 Principio di induzione noetheriana

**Teorema 1** (Principio di induzione noetheriana (prima forma)). *Sia  $\mathcal{P}$  una proprietà su  $(A, \triangleleft)$  b.f. .*

$$\forall a \in A. \mathcal{P}(a) \Leftrightarrow \forall a \in A. ([\forall b \triangleleft a. \mathcal{P}(b)] \Rightarrow \mathcal{P}(a))$$

*Dimostrazione.* •  $\Rightarrow$ ) Ovvio.

•  $\Leftarrow$ ) Per assurdo.

Supponiamo

$$\forall a \in A. ([\forall b \triangleleft a. \mathcal{P}(b)] \Leftarrow \mathcal{P}(a)) \quad (1.1)$$

e

$$\exists c \in A. \neg \mathcal{P}(c) \quad (1.2)$$

Sia  $C = \{c \in A \mid \neg \mathcal{P}(c)\} \subseteq A$ .

Per il lemma 2,  $\exists \hat{c}$  minimale di  $C$  rispetto a  $\triangleleft$ , allora  $\neg \mathcal{P}(\hat{c})$

Per  $\hat{c}$  minimale,  $\forall b \triangleleft \hat{c}. b \notin C$ , allora  $\mathcal{P}(b)$ , ovvero per (1.1)  $\mathcal{P}(\hat{c})$

□

**Def.**  $\text{base}_A = \{a \in A \mid a \text{ è minimale in } A \text{ rispetto a } \triangleleft\}$ .

Osserviamo che  $\forall a \in \text{base}_A, \forall b \in A. b \not\triangleleft a$ .

**Teorema 2** (Principio di induzione noetheriana - seconda forma). *Sia  $\mathcal{P}$  una proprietà su  $(A, \triangleleft)$  b.f. .*

$$\forall a \in A. \mathcal{P}(a) \Leftrightarrow \left( \begin{array}{c} \forall a \in \text{base}_A. \mathcal{P}(a) \\ \wedge \\ \forall a \in (A \setminus \text{base}_A). ([\forall b \triangleleft a. \mathcal{P}(b)] \Rightarrow \mathcal{P}(a)) \end{array} \right)$$

**Teorema 3** (Induzione matematica). *Sia  $A = \mathbb{N}$ .*

*Sia  $n \triangleleft m \Leftrightarrow m = n + 1 \quad n, m \in \mathbb{N}$ .*

*Osserviamo che  $\triangleleft$  è ben fondata:  $0 \triangleleft 1 \triangleleft 2 \triangleleft \dots$ .*

*Osserviamo  $\text{base}_{\mathbb{N}} = \{0\}$ .*

$$\forall m \in \mathbb{N}. \mathcal{P}(m) \Leftrightarrow \left( \begin{array}{c} \mathcal{P}(0) \\ \wedge \\ \forall m > 0. (\mathcal{P}(m-1) \Rightarrow \mathcal{P}(m)) \end{array} \right)$$

**Def.**  $a^{\triangleleft} = \{b \in A \mid b \triangleleft a\}$  con  $(A, \triangleleft)$  b.f. .

**Def.** Sia  $f : A \rightarrow B$ , sia  $A' \subseteq A$ .  $f|_{A'} = \{(a, f(a)) \mid a \in A'\}$ .

**Teorema 4** (di ricorsione / delle definizioni noetheriane). *Sia  $(A, \triangleleft)$  b.f.*

$$\forall a \in A, \forall h : a^{\triangleleft} \rightarrow B. F(a, h) \in B$$

*( $F$  è detta operatore di composizione). Allora*

$$\exists! f : A \rightarrow B \text{ t.c. } \forall a \in A. f(a) = F(a, f|_{a^{\triangleleft}})$$

*Esempio.* Sia  $A = \mathbb{N}$ , sia  $n \triangleleft m \Leftrightarrow m = n + 1$ .

$$\text{Fact}(n) = \begin{cases} 1 & \text{se } n = 0 \\ n \cdot \text{Fact}(n-1) & \text{se } n > 0 \end{cases}$$

In questo caso,  $f = \text{Fact}$  e  $F$  è la moltiplicazione.

$$f(n) = F(n, f(n-1)) = n \cdot f(n-1)$$

## Capitolo 2

# Definizione di Linguaggio

TODO

**Composizionalità:** la proprietà per cui ciascuna stringa deve essere funzione solo dei componenti della stringa stessa.

**Modularità:** la proprietà per cui, aggiungendo dei costrutti ad un linguaggio  $L$ , non devo ridefinire la semantica dello stesso.

- posso quindi definire un linguaggio in maniera incrementale, ovvero, partendo da un nucleo centrale, posso aggiungere dello zucchero sintattico senza modificare il nucleo di partenza.

## Capitolo 3

# Semantica

- **Operazionale:** COME eseguire un programma  $P$ .
- **Denotazionale:** CHE COSA si ottiene dall'esecuzione di  $P$ .
- **Assiomatica:** verifica SE il programma é corretto rispetto ad una data proprietà.

**Def.** dato un programma  $P$ :

$$\dots z := 2; y := z; y := y + 1; z := y. \dots \quad (3.1)$$

il **supporto a tempo di esecuzione** é definito come:

$$[z = 0, y = 0] \equiv \rho(z) = 0, \rho(y) = 0 \quad (3.2)$$

TODO

### 3.1 Semantica operativa

**Def.** Un **sistema di transizione**  $T_G$  é un grafo t.c.

$$T_G = (P \times \rho, \longrightarrow, I, F)$$

dove

- $P$  sono i programmi (ad es. assegnazioni di (3.1))
- $\rho$  sono gli ambienti del supporto a tempo di esecuzione (ad es. (3.2))
- $\longrightarrow \subseteq (P, \rho) \times (P, \rho)$
- $I \subseteq (P, \rho)$  iniziali
- $F \subseteq (P, \rho)$  finali

**Def.** La **semantica operativa** é una funzione

$$\mathcal{E}_{op} : L(G) \times \rho_{init} \longrightarrow T_G$$

### 3.2 Semantica denotazionale

**Def.** La **semantica denotazionale** é definita come

$$\mathcal{E}_{den} : L(G) \times \rho_{init} \longrightarrow \rho_{fin}$$

Rappresenta solo il risultato dell'esecuzione del programma, ovvero l'insieme degli ambienti finali.

### 3.3 Semantica assiomatica

TODO