

Elementi di Logica Matematica

Nicola Gambino

A.A. 2009/10

Nota

Questo documento contiene le dispense del corso di Elementi di Logica Matematica offerto dalla Facoltà di Scienze MM.FF.NN. dell'Università degli Studi di Palermo per l'anno accademico 2009/10.

I capitoli 1 e 2 sono tratti da [vD08, End01], mentre i capitoli 3 e 4 sono tratti da [Joh87, MvO]. A volte ho fatto ricorso anche a materiale tratto da [BBJ07, End01, DS96]. Nulla, quindi, salvo la scelta del materiale, alcuni aspetti della presentazione e qualche esercizio è da considerarsi originale. Errori e imprecisioni rimangono comunque di mia responsabilità. Ringrazio gli studenti dell'anno accademico 2008/09 che hanno contribuito a ridurre il numero con le loro segnalazioni. Ulteriori segnalazioni (da inviarsi per e-mail all'indirizzo ngambino@math.unipa.it) sono benvenute.

Palermo, 24/02/2010

Indice

1	Logica proposizionale	1
1	Le formule proposizionali	1
2	Il calcolo della deduzione naturale	3
3	Valutazioni e tavole di verità	8
4	Il teorema di validità	12
5	Il teorema di completezza	16
6	Esercizi	20
2	Logica del primo ordine	23
1	Linguaggi del primo ordine	23
2	Il calcolo della deduzione naturale	26
3	Strutture	29
4	Il teorema di validità	31
5	Il teorema di completezza e alcune sue applicazioni	34
6	Definibilità	36
7	Esercizi	37
3	Teoria della calcolabilità	41
1	Funzioni ricorsive	41
2	Funzioni calcolabili	43
3	La tesi di Church	44
4	Esempio di una funzione non ricorsiva	48
5	Insiemi ricorsivi e ricorsivamente enumerabili	49
6	Definibilità in PA	52
7	Esercizi	53
4	Teoria degli insiemi	55
1	La teoria degli insiemi di Zermelo-Fraenkel	55
2	Esempi della codifica della matematica in ZF	59
3	Induzione insiemistica	61
4	Ordinali	63
5	L'assioma della scelta	64
6	Cardinali	66
7	Esercizi	68

Capitolo 1

Logica proposizionale

1 Le formule proposizionali

Sia $P = \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$ un insieme numerabile. L'insieme Frm_P delle *formule proposizionali* generato da P è il più piccolo insieme X che soddisfa le seguenti proprietà:

- (i) se $p \in P$ allora $p \in X$,
- (ii) $\top \in X$ e $\perp \in X$,
- (iii) se $\varphi, \psi \in X$ allora $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \Rightarrow \psi) \in X$.

Esempio 1.1. Se $p, q, r \in P$ allora

$$(p \Rightarrow (q \wedge r)), \quad ((p \vee q) \wedge (p \vee r)), \quad (p \Rightarrow (p \Rightarrow \perp))$$

sono formule proposizionali.

Per questo capitolo considereremo l'insieme P come fissato. Diremo semplicemente formula anziché formula proposizionale, visto che non utilizzeremo altri tipi di formule. Si noti che $P \subseteq \text{Frm}_P$. Gli elementi di P saranno detti *formule atomiche*. Le formule \top e \perp saranno chiamate il *vero* e il *falso*, rispettivamente. Date due formule φ e ψ , chiameremo $(\varphi \wedge \psi)$ la *coniunzione* di φ e ψ , $(\varphi \vee \psi)$ la *disgiunzione* di φ e ψ , e $(\varphi \Rightarrow \psi)$ l'*implicazione* da φ a ψ . È conveniente introdurre alcune abbreviazioni e convenzioni. Innanzitutto, definiamo

$$\begin{aligned}(\varphi \Leftrightarrow \psi) &=_{\text{def}} ((\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)), \\(\neg \varphi) &=_{\text{def}} (\varphi \Rightarrow \perp).\end{aligned}$$

Chiameremo $(\neg \varphi)$ la *negazione* di φ . Nel seguito, utilizzeremo le seguenti convenzioni, in maniera da semplificare la scrittura e la lettura delle formule.

- Le parentesi più esterne verranno omesse. Per esempio,

$$\varphi \wedge \psi \quad \text{abbrevia} \quad (\varphi \wedge \psi).$$

- Il simbolo di negazione va inteso come applicato solo alla formula che lo segue immediatamente. Per esempio,

$$\neg\varphi \wedge \psi \quad \text{abbrevia} \quad (\neg\varphi) \wedge \psi.$$

- Salve rimanendo le convenzioni precedenti, i simboli di congiunzione e disgiunzione vanno intesi come applicati solo alle formule piú vicine tra cui compaiono. Per esempio,

$$\varphi_1 \wedge \psi_1 \Rightarrow \neg\varphi_2 \vee \psi_2 \quad \text{abbrevia} \quad (\varphi_1 \wedge \psi_1) \Rightarrow (\neg\varphi_2 \vee \psi_2).$$

- Quando lo stesso connettivo è utilizzato ripetutamente, sarà inteso come applicato da destra a sinistra. Per esempio,

$$\begin{aligned} \varphi \wedge \psi \wedge \chi & \quad \text{abbrevia} \quad \varphi \wedge (\psi \wedge \chi), \\ \varphi \Rightarrow \psi \Rightarrow \chi & \quad \text{abbrevia} \quad \varphi \Rightarrow (\psi \Rightarrow \chi). \end{aligned}$$

In base a queste convenzioni, le formule dell'Esempio 1.1 possono essere riscritte come

$$p \Rightarrow (q \wedge r), \quad (p \vee q) \wedge (p \vee r), \quad p \Rightarrow p \Rightarrow \perp.$$

Nel caso non si sia sicuri di rispettare le convenzioni è sempre possibile non eliminare parentesi.

Dalla definizione dell'insieme delle formule proposizionali segue che esso soddisfa un principio di induzione analogo al principio di induzione per l'insieme dei numeri naturali.

Principio di induzione per le formule proposizionali. Per dimostrare che vale $\mathcal{S}(\varphi)$ per ogni formula φ , è sufficiente dimostrare le seguenti asserzioni.

- *Primo caso base.* Per ogni $p \in P$, vale $\mathcal{S}(p)$.
- *Secondo caso base.* Valgono $\mathcal{S}(\top)$ e $\mathcal{S}(\perp)$.
- *Passi induttivi.* Se valgono $\mathcal{S}(\varphi)$ e $\mathcal{S}(\psi)$ allora valgono anche $\mathcal{S}(\varphi \wedge \psi)$, $\mathcal{S}(\varphi \vee \psi)$, $\mathcal{S}(\varphi \Rightarrow \psi)$.

Per esempio, il principio di induzione per le formule proposizionali può essere utilizzato per dimostrare che ogni formula contiene un numero pari di parentesi. Abbiamo anche un analogo al principio di definizione per ricorsione di funzioni per i numeri naturali.

Principio di definizione per ricorsione per le formule proposizionali. Sia A un insieme. Si supponga di avere i seguenti dati.

- Una funzione $f : P \rightarrow A$,
- Elementi $\top_A \in A$, $\perp_A \in A$,

- Funzioni

$$\begin{array}{lll}
 A \times A & \rightarrow & A \\
 (a, b) & \mapsto & a \wedge_A b,
 \end{array}
 \quad
 \begin{array}{lll}
 A \times A & \rightarrow & A \\
 (a, b) & \mapsto & a \vee_A b,
 \end{array}
 \quad
 \begin{array}{lll}
 A \times A & \rightarrow & A \\
 (a, b) & \mapsto & a \Rightarrow_A b.
 \end{array}$$

Allora esiste una unica funzione $v : \text{Frm}_P \rightarrow A$ che soddisfa le seguenti proprietà:

$$\begin{aligned}
 v(p) &= f(p) \\
 v(\top) &= \top_A \\
 v(\perp) &= \perp_A \\
 v(\varphi \wedge \psi) &= v(\varphi) \wedge_A v(\psi), \\
 v(\varphi \vee \psi) &= v(\varphi) \vee_A v(\psi), \\
 v(\varphi \Rightarrow \psi) &= v(\varphi) \Rightarrow_A v(\psi),
 \end{aligned}$$

ove p è una formula atomica, mentre φ e ψ sono formule arbitrarie. Utilizzeremo il principio di definizione per ricorsione nella Sezione 3, in cui introdurremo il concetto di valutazione.

2 Il calcolo della deduzione naturale

Sia Γ un insieme di formule e φ una formula. Vogliamo definire che cosa significhi per φ essere *derivabile* da Γ . L'idea che vogliamo rendere precisa è che l'insieme Γ costituisce un insieme di ipotesi da cui segue la conclusione φ , una situazione che indicheremo scrivendo $\Gamma \vdash \varphi$. Se $\Gamma = \emptyset$, scriveremo $\vdash \varphi$ anziché $\emptyset \vdash \varphi$. Esistono diversi, ma essenzialmente equivalenti, sistemi per assiomatizzare questa nozione. Noi seguiremo il sistema della *deduzione naturale*, che cattura in maniera intuitiva e precisa le leggi utilizzate comunemente nel ragionamento matematico. Per esempio, il sistema della deduzione naturale ci permette di dimostrare $\{\varphi, \varphi \Rightarrow \psi, \varphi \Rightarrow \chi\} \vdash \psi \wedge \chi$ tramite la costruzione del seguente albero di deduzione:

$$\frac{\frac{\varphi \Rightarrow \psi}{\psi} \Rightarrow_E \quad \frac{\varphi \Rightarrow \chi}{\chi} \Rightarrow_E}{\psi \wedge \chi} \wedge_I$$

Le regole della deduzione naturale sono regole per costruire alberi di deduzione come il precedente. Nel dare le regole della deduzione naturale, distingueremo (quando è appropriato farlo) tra regole di introduzione e regole di eliminazione.

Regole per la congiunzione. Per la congiunzione, abbiamo una regola di introduzione e due regole di eliminazione:

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} \wedge_I$$

$$\frac{\varphi \wedge \psi}{\varphi} \wedge_{E,1} \quad \frac{\varphi \wedge \psi}{\psi} \wedge_{E,2}$$

La regola di introduzione ci dice che se possiamo dedurre sia φ e ψ , allora possiamo dedurre anche la loro congiunzione $\varphi \wedge \psi$. Le regole di eliminazione ci dicono che se possiamo dedurre $\varphi \wedge \psi$, allora possiamo dedurre sia φ che ψ .

Esempio 2.1.

(i) Per dimostrare $\{\varphi \wedge \psi\} \vdash \psi \wedge \varphi$ possiamo costruire l'albero

$$\frac{\frac{\varphi \wedge \psi}{\psi} \wedge_{E,2} \quad \frac{\varphi \wedge \psi}{\varphi} \wedge_{E,1}}{\psi \wedge \varphi} \wedge_I$$

(ii) Per dimostrare $\{(\varphi \wedge \psi) \wedge \chi\} \vdash \varphi \wedge (\psi \wedge \chi)$ costruiamo l'albero

$$\frac{\frac{\frac{(\varphi \wedge \psi) \wedge \chi}{\varphi \wedge \psi} \wedge_{E,1} \quad \frac{\varphi \wedge \psi}{\psi} \wedge_{E,2}}{\varphi} \wedge_{E,1} \quad \frac{\frac{(\varphi \wedge \psi) \wedge \chi}{\chi} \wedge_{E,2}}{\psi \wedge \chi} \wedge_I}{\varphi \wedge (\psi \wedge \chi)} \wedge_I$$

Regole per l'implicazione. Per l'implicazione, abbiamo una regola di introduzione e una regola di eliminazione. La regola di introduzione è

$$\frac{\begin{array}{c} [\varphi]_n \\ \vdots \\ \psi \end{array}}{\varphi \Rightarrow \psi} \Rightarrow_{I,n}$$

La regola di eliminazione è

$$\frac{\varphi \Rightarrow \psi \quad \varphi}{\psi} \Rightarrow_E$$

La regola di eliminazione, nota anche come *modus ponens*, ci dice che se sappiamo dedurre $\varphi \Rightarrow \psi$ e φ , allora possiamo dedurre ψ . La regola di introduzione dice che se dall'ipotesi φ sappiamo dedurre ψ , allora possiamo dedurre $\varphi \Rightarrow \psi$ senza più assumere φ . Quest'ultimo aspetto è indicato mettendo la formula φ tra parentesi quadre etichettate con un indice, dato da un numero $n \in \mathbb{N}$, e ripetendo lo stesso indice nel passaggio in cui l'ipotesi φ viene scaricata. È importante ricordare che, una volta che un'ipotesi viene scaricata, essa non può più essere utilizzata, a meno che non venga nuovamente assunta.

Esempio 2.2.

(i) Per dimostrare $\{\varphi \Rightarrow \psi, \varphi \Rightarrow \chi\} \vdash \varphi \Rightarrow (\psi \wedge \chi)$, costruiamo l'albero

$$\frac{\frac{\varphi \Rightarrow \psi \quad [\varphi]_1}{\psi} \Rightarrow_E \quad \frac{\varphi \Rightarrow \chi \quad [\varphi]_1}{\chi} \Rightarrow_E}{\frac{\psi \wedge \chi}{\varphi \Rightarrow (\psi \wedge \chi)} \Rightarrow_{I,1}}$$

(ii) Per dimostrare $\vdash (\varphi \Rightarrow \chi) \Rightarrow ((\varphi \wedge \psi) \Rightarrow \chi)$, costruiamo l'albero

$$\frac{\frac{\frac{[\varphi \Rightarrow \chi]_2}{\chi} \Rightarrow_{I,1} \quad \frac{\frac{[\varphi \wedge \psi]_1}{\varphi} \wedge_{E,1}}{\varphi \Rightarrow \chi} \Rightarrow_{E,1}}{(\varphi \wedge \psi) \Rightarrow \chi} \Rightarrow_{I,1}}{(\varphi \Rightarrow \chi) \Rightarrow ((\varphi \wedge \psi) \Rightarrow \chi)} \Rightarrow_{I,2}$$

(iii) Per dimostrare $\vdash \varphi \Rightarrow \neg\neg\varphi$, costruiamo l'albero

$$\frac{\frac{\frac{[\neg\varphi]_1}{\perp} \Rightarrow_{E,1} \quad [\varphi]_2}{\neg\neg\varphi} \Rightarrow_{E,2}}{\varphi \Rightarrow \neg\neg\varphi} \Rightarrow_{E,2}$$

L'applicazione della regola di eliminazione dell'implicazione nel primo passaggio è giustificata dalla definizione della negazione data nella Sezione 1. Infatti, questo passaggio è identico al seguente:

$$\frac{\varphi \Rightarrow \perp \quad \varphi}{\perp} \Rightarrow_E$$

Quest'ultimo è chiaramente un'istanza della regola di eliminazione per l'implicazione.

Osservazione 2.3. È possibile applicare la regola di introduzione dell'implicazione scaricando una formula che non è stata assunta. In questo caso, il passaggio non richiede che venga indicato alcun indice. Per esempio, la seguente derivazione è da considerarsi corretta:

$$\frac{\frac{[\varphi]_1}{\psi \Rightarrow \varphi} \Rightarrow_I}{\varphi \Rightarrow (\psi \Rightarrow \varphi)} \Rightarrow_{I,1}$$

Questa deduzione è corretta in quanto avremmo potuto introdurre passaggi ridondanti che coinvolgono la formula non assunta, ma scaricata. Per esempio, potremmo derivare $\varphi \Rightarrow (\psi \Rightarrow \varphi)$ nel modo seguente:

$$\frac{\frac{\frac{[\varphi]_2}{\varphi \wedge \psi} \wedge_{E,1} \quad [\psi]_1}{\varphi} \wedge_{E,1}}{\psi \Rightarrow \varphi} \Rightarrow_{I,1}}{\varphi \Rightarrow (\psi \Rightarrow \varphi)} \Rightarrow_{I,2}$$

Regole per la disgiunzione. Abbiamo due regole di introduzione e una regola di eliminazione.

$$\frac{\varphi}{\varphi \vee \psi} \vee_{I,1} \qquad \frac{\psi}{\varphi \vee \psi} \vee_{I,2}$$

$$\frac{\varphi \vee \psi \quad \begin{array}{c} [\varphi]_n \\ \vdots \\ \chi \end{array} \quad \begin{array}{c} [\psi]_m \\ \vdots \\ \chi \end{array}}{\chi} \vee_{E,n,m}$$

Le regole di introduzione ci dicono che se possiamo dedurre φ o ψ , allora possiamo dedurre anche $\varphi \vee \psi$. La regola di eliminazione ci dice che se sappiamo dedurre $\varphi \vee \psi$, sappiamo dedurre χ assumendo anche φ , sappiamo dedurre χ assumendo anche ψ , allora possiamo dedurre χ , senza piú assumere nè φ nè ψ . Come per la regola di eliminazione per l'implicazione, utilizziamo parentesi quadre e indici per segnalare ipotesi che vengono eliminate.

Esempio 2.4. Per dimostrare $\{\varphi \wedge (\psi \vee \chi)\} \vdash (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$, costruiamo l'albero

$$\frac{\frac{\varphi \wedge (\psi \vee \chi)}{\psi \vee \chi} \wedge_{E,2} \quad \frac{\frac{\frac{\varphi \wedge (\psi \vee \chi)}{\varphi} \wedge_{E,1} \quad [\psi]_1 \wedge_I}{\varphi \wedge \psi} \wedge_I \quad \frac{\frac{\varphi \wedge (\psi \vee \chi)}{\varphi} \wedge_{E,1} \quad [\chi]_2 \wedge_I}{\varphi \wedge \chi} \wedge_I}{(\varphi \wedge \psi) \vee (\varphi \wedge \chi)} \vee_{I,1}}{(\varphi \wedge \psi) \vee (\varphi \wedge \chi)} \vee_{E,1,2}$$

Regole per le dimostrazioni per assurdo. La regola per le dimostrazioni per assurdo è

$$\frac{\begin{array}{c} [\neg\varphi]_n \\ \vdots \\ \perp \end{array}}{\varphi} \text{RAA}_n$$

Questa regola, nota anche come *reductio ad absurdum*, ci dice che se sappiamo dedurre il falso dall'ipotesi $\neg\varphi$, allora possiamo dedurre φ , senza piú bisogno dell'ipotesi $\neg\varphi$. Ancora una volta, utilizziamo parentesi quadre e indici per indicare che l'ipotesi $\neg\varphi$ viene scaricata nell'applicazione della regola.

Esempio 2.5. Dimostriamo la *legge della doppia negazione*, ovvero $\vdash \neg\neg\varphi \Rightarrow \varphi$.

$$\frac{\frac{\frac{[\neg\neg\varphi]_2 \quad [\neg\varphi]_1}{\perp} \text{RAA}_1}{\varphi} \Rightarrow_E}{\neg\neg\varphi \Rightarrow \varphi} \Rightarrow_{I,2}$$

Si noti che l'applicazione della regola di eliminazione dell'implicazione, nel primo passaggio, è giustificata dalla definizione, data nella Sezione 1, della negazione. In particolare, quel passaggio può essere riscritto equivalentemente come

$$\frac{\neg\varphi \Rightarrow \perp \quad \neg\varphi}{\perp} \Rightarrow E$$

che è chiaramente un'istanza della regola di eliminazione dell'implicazione.

Regole per il vero e per il falso. Per il vero, abbiamo solo una regola di introduzione, mentre per il falso abbiamo solo una regola di eliminazione.

$$\frac{}{\top} \top_I \qquad \frac{\perp}{\varphi} \perp_E$$

La regola di introduzione per il vero ci dice semplicemente che, senza alcuna premessa, possiamo dedurre il vero. La regola di eliminazione per il falso, nota anche come *ex falso quodlibet*, ci dice che se le nostre ipotesi ci permettono di dedurre il falso, allora ci permettono di dedurre qualsiasi formula. Infine, abbiamo la regola più semplice:

$$\frac{\varphi}{\varphi}$$

che esprime semplicemente che ogni formula permette di derivare se stessa.

Definizione 2.6.

- Sia Γ un insieme di formule e sia φ una formula. Diremo che φ è *derivabile* da Γ se esiste un'albero di derivazione con conclusione φ e le cui ipotesi non scaricate sono elementi di Γ . Scriveremo $\Gamma \vdash \varphi$ per indicare che φ è derivabile da Γ .
- Sia φ una formula. Diremo che φ è un *teorema* se φ è derivabile dall'insieme vuoto. Scriveremo $\vdash \varphi$ per indicare che φ è un teorema.

Proposizione 2.7. Le seguenti formule sono teoremi.

(i) Leggi di idempotenza:

$$\begin{aligned} \varphi \wedge \varphi &\Leftrightarrow \varphi, \\ \varphi \vee \varphi &\Leftrightarrow \varphi. \end{aligned}$$

(ii) Leggi associative:

$$\begin{aligned} (\varphi \wedge \psi) \wedge \chi &\Leftrightarrow \varphi \wedge (\psi \wedge \chi), \\ (\varphi \vee \psi) \vee \chi &\Leftrightarrow \varphi \vee (\psi \vee \chi). \end{aligned}$$

(iii) Leggi distributive:

$$\begin{aligned} \varphi \vee (\psi \wedge \chi) &\Leftrightarrow (\varphi \vee \psi) \wedge (\varphi \vee \chi), \\ \varphi \wedge (\psi \vee \chi) &\Leftrightarrow (\varphi \wedge \psi) \vee (\varphi \wedge \chi). \end{aligned}$$

(iv) Legge della doppia negazione: $\neg\neg\varphi \Leftrightarrow \varphi$.

(v) Legge del terzo escluso: $\varphi \vee \neg\varphi$.

(vi) Leggi di De Morgan:

$$\begin{aligned}\neg(\varphi \vee \psi) &\Leftrightarrow \neg\varphi \wedge \neg\psi, \\ \neg(\varphi \wedge \psi) &\Leftrightarrow \neg\varphi \vee \neg\psi.\end{aligned}$$

(vii) Riduzione dell'implicazione: $(\varphi \Rightarrow \psi) \Leftrightarrow (\neg\varphi \vee \psi)$.

Dimostrazione. Esercizio. □

3 Valutazioni e tavole di verità

Le regole del calcolo di deduzione naturale ci permettono di stabilire che una formula φ segue da un insieme di ipotesi Γ . Cominciamo ora ad affrontare il problema di come sia possibile stabilire che una formula φ non sia derivabile da un insieme di ipotesi Γ . Questo tipo di problema è apparentemente difficile da risolvere, visto che dire che φ non è derivabile da Γ significa dire che le regole del calcolo di deduzione naturale non permettono di costruire nessun albero di derivazione con ipotesi non scaricate in Γ e conclusione φ . La nozione di valutazione, che introdurremo nella Definizione 3.1, ci permetterà di studiare il problema in maniera efficace.

Chiameremo $\{0, 1\}$ l'insieme dei *valori di verità booleani*. Useremo 1 per verità e 0 per falsità. Dati $x, y \in \{0, 1\}$, definiamo

$$\begin{aligned}x \wedge y &=_{\text{def}} \begin{cases} 1 & \text{se } x = 1 \text{ e } y = 1, \\ 0 & \text{altrimenti.} \end{cases} \\ x \vee y &=_{\text{def}} \begin{cases} 1 & \text{se } x = 1 \text{ o } y = 1, \\ 0 & \text{altrimenti.} \end{cases} \\ x \Rightarrow y &=_{\text{def}} \begin{cases} 0 & \text{se } x = 1 \text{ e } y = 0, \\ 1 & \text{altrimenti.} \end{cases}\end{aligned}$$

Infine, per $x \in \{0, 1\}$ definiamo $\neg x =_{\text{def}} x \Rightarrow 0$. È facile verificare che:

$$\neg x = \begin{cases} 1 & \text{se } x = 0, \\ 0 & \text{se } x = 1. \end{cases}$$

Definizione 3.1. Una *valutazione* è una funzione $v : \text{Frm}_P \rightarrow \{0, 1\}$ con le seguenti proprietà:

- $v(\perp) = 0$ e $v(\top) = 1$
- $v(\varphi * \psi) = v(\varphi) * v(\psi)$, ove $*$ $\in \{\wedge, \vee, \Rightarrow\}$.

Dal principio di definizione per ricorsione per le formule proposizionali, si ha che per ogni funzione $f : P \rightarrow \{0, 1\}$ esiste una unica valutazione v tale che $v(p) = f(p)$ per ogni $p \in P$. In altre parole, per definire una valutazione è sufficiente definire i valori di verità delle formule atomiche. Inoltre, se due valutazioni coincidono sulle formule proposizionali, allora coincidono su tutte le formule.

Definizione 3.2.

- Sia v una valutazione, φ una formula. Diremo che v *soddisfa* φ se $v(\varphi) = 1$.
- Sia Γ un insieme di formule e φ una formula. Diremo che Γ *implica semanticamente* φ se ogni valutazione che soddisfa tutte le formule in Γ soddisfa anche φ . Scriveremo $\Gamma \models \varphi$ per indicare che Γ implica semanticamente φ .
- Sia φ una formula. Diremo che φ è una *tautologia* se φ è soddisfatta da ogni valutazione. Scriveremo $\models \varphi$ per indicare che φ è una tautologia.

I valori di una valutazione possono essere descritti anche tramite le cosiddette tavole di verità per la logica proposizionale. Uno dei vantaggi delle tavole di verità è che esse permettono di calcolare i valori di una valutazione in maniera completamente algoritmica, anche se talvolta laboriosa. Daremo ora le tavole di verità per ciascun connettivo.

Tavola di verità per la congiunzione.

φ	ψ	$\varphi \wedge \psi$
1	1	1
1	0	0
0	1	0
0	0	0

Quindi, una valutazione soddisfa $\varphi \wedge \psi$ se e solo se soddisfa sia φ che ψ .

Tavola di verità per la disgiunzione.

φ	ψ	$\varphi \vee \psi$
1	1	1
1	0	1
0	1	1
0	0	0

Quindi, una valutazione soddisfa $\varphi \vee \psi$ se e solo se soddisfa almeno una tra φ e ψ .

Tavola di verità per l'implicazione.

φ	ψ	$\varphi \Rightarrow \psi$
1	1	1
1	0	0
0	1	1
0	0	1

Questa tavola di verità è la più complessa da giustificare. Il punto più delicato riguarda i casi in cui $v(\varphi) = 0$, ovvero in cui la premessa dell'implicazione non è soddisfatta. In attesa di ritornare su questo punto nell'Osservazione 3.6, si consideri la seguente affermazione:

“Se $0 = 1$, allora $2 = 3$ ”.

È opportuno dichiarare questa implicazione vera, visto che abbiamo il seguente, perfettamente legittimo, ragionamento:

“Se $0 = 1$, allora $0 + 2 = 1 + 2$, da cui segue $2 = 3$ ”.

La convenzione di dichiarare $v(\varphi \Rightarrow \psi) = 1$ ogniqualvolta $v(\varphi) = 0$ intende catturare l'idea che da premesse false è possibile, secondo ragionamenti legittimi, ottenere qualsiasi conclusione, anche conclusioni che sono a loro volta false.

Esempio 3.3 (Tavola di verità della negazione). Possiamo calcolare la tavola di verità per la negazione, che è completamente determinata dalla tavola di verità dell'implicazione.

φ	\perp	$\varphi \Rightarrow \perp$
1	0	0
0	0	1

Possiamo riassumere questa tavola come segue:

φ	$\neg\varphi$
1	0
0	1

Quindi, una valutazione soddisfa $\neg\varphi$ se e solo se non soddisfa φ .

Esempio 3.4. Possiamo combinare le tavole di verità date finora per ottenere le tavole di verità di formule complesse. Per esempio, ricordando che abbiamo definito

$$(\varphi \Leftrightarrow \psi) =_{\text{def}} (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi),$$

possiamo costruire la seguente tavola di verità:

φ	ψ	$\varphi \Rightarrow \psi$	$\psi \Rightarrow \varphi$	$(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$
1	1	1	1	1
1	0	0	1	0
0	1	1	0	0
0	0	1	1	1

Possiamo riassumere questa tavola nel modo seguente:

φ	ψ	$\varphi \Leftrightarrow \psi$
1	1	1
1	0	0
0	1	0
0	0	1

Quindi, per ogni valutazione v , si ha che $v(\varphi \Leftrightarrow \psi) = 1$ se e solo se $v(\varphi) = v(\psi)$.

Esempio 3.5. Le tavole di verità ci permettono di verificare quando una formula è una tautologia. Per esempio, verifichiamo che

$$(\varphi \Rightarrow \psi) \Leftrightarrow (\neg\varphi \vee \psi)$$

è una tautologia. In base alle osservazioni fatte nell'Esempio 3.4, è sufficiente verificare che $v(\varphi \Rightarrow \psi) = v(\neg\varphi \vee \psi)$ per ogni valutazione v . A tal fine, calcoliamo la tavola di verità di $\neg\varphi \vee \psi$, che è la seguente:

φ	ψ	$\neg\varphi$	$\neg\varphi \vee \psi$
1	1	0	1
1	0	0	0
0	1	1	1
0	0	1	1

Abbiamo quindi

φ	ψ	$\neg\varphi \vee \psi$	$\varphi \Rightarrow \psi$
1	1	1	1
1	0	0	0
0	1	1	1
0	0	1	1

Il che dimostra che, per ogni valutazione v , si ha $v(\neg\varphi \vee \psi) = v(\varphi \Rightarrow \psi)$, come volevasi dimostrare.

Osservazione 3.6. Per concludere questa sezione, ritorniamo a considerare la giustificazione della tavola di verità dell'implicazione. Si ricordi che la formula

$$(\varphi \Rightarrow \psi) \Leftrightarrow (\neg\varphi \vee \psi)$$

è un teorema. Se vogliamo, come è naturale richiedere, che ogni teorema sia una tautologia, la tavola di verità per $\varphi \Rightarrow \psi$ è completamente determinata dalle tavole di verità per la disgiunzione e per la negazione, entrambe delle quali sono ben motivate. Il fatto che la negazione sia definita in termini dell'implicazione crea un apparente circolo vizioso, ma questo può essere evitato introducendo la negazione come un simbolo primitivo, anziché definito, e aggiungendo regole di deduzione che permettano di dimostrare che $\neg\varphi \Leftrightarrow (\varphi \Rightarrow \perp)$ è un teorema.

4 Il teorema di validità

Teorema 4.1 (Teorema di Validità). Se $\Gamma \vdash \varphi$ allora $\Gamma \vDash \varphi$.

Dimostrazione. Supponiamo che $\Gamma \vdash \varphi$. Questo significa che esiste un albero di deduzione naturale con conclusione φ e le cui premesse non scaricate sono elementi di Γ . Per dimostrare che $\Gamma \vDash \varphi$ procediamo per induzione sull'altezza di quest'albero, distinguendo a seconda di quale sia l'ultima regola applicata. Per il caso base, abbiamo che gli unici alberi di altezza 1, sono

$$\frac{\varphi}{\varphi} \quad \frac{}{\top}$$

Per la prima regola, visto che φ è l'unica ipotesi dell'albero, dobbiamo dimostrare che se $\varphi \in \Gamma$, allora $\Gamma \vDash \varphi$, ma questo è evidente. Per la seconda regola, ogni valutazione è tale che $v(\top) = 1$ e quindi non c'è nulla da dimostrare.

Se l'ultima regola applicata è l'introduzione della congiunzione, l'albero di deduzione ha la forma

$$\frac{\begin{array}{c} \vdots \\ \varphi_1 \end{array} \quad \begin{array}{c} \vdots \\ \varphi_2 \end{array}}{\varphi_1 \wedge \varphi_2} \wedge_I$$

Ipotesi induttive.

- Se Γ_1 contiene tutte le ipotesi del sottoalbero con conclusione φ_1 allora $\Gamma_1 \vDash \varphi_1$.
- Se Γ_2 contiene tutte le ipotesi del sottoalbero con conclusione φ_2 allora $\Gamma_2 \vDash \varphi_2$.

Tesi. Se Γ contiene le tutte ipotesi dell'albero con conclusione $\varphi_1 \wedge \varphi_2$ allora $\Gamma \vDash \varphi_1 \wedge \varphi_2$.

Dimostrazione della tesi. Sia Γ un insieme che contiene tutte le ipotesi dell'albero con conclusione $\varphi_1 \wedge \varphi_2$. Definiamo Γ_1 come l'insieme delle ipotesi del sottoalbero con conclusione φ_1 e Γ_2 come l'insieme delle ipotesi del sottoalbero con conclusione φ_2 . Dalle ipotesi induttive, segue che $\Gamma_1 \vDash \varphi_1$ e $\Gamma_2 \vDash \varphi_2$. Oltre a questo, abbiamo che $\Gamma_1 \cup \Gamma_2 \subseteq \Gamma$. Sia ora v una valutazione che rende valida ogni formula in Γ . In particolare v rende valida ogni formula in Γ_1 e in Γ_2 . Visto che $\Gamma_1 \vDash \varphi_1$ e $\Gamma_2 \vDash \varphi_2$, ne segue che $v(\varphi_1) = 1$ e $v(\varphi_2) = 1$. Da questo possiamo dedurre $v(\varphi_1 \wedge \varphi_2) = 1$, come volevasi dimostrare.

Se l'ultima regola applicata è un'eliminazione della congiunzione, l'albero di deduzione ha una delle seguenti due forme:

$$\frac{\begin{array}{c} \vdots \\ \varphi_1 \wedge \varphi_2 \end{array}}{\varphi_1} \wedge_{E,1} \quad \frac{\begin{array}{c} \vdots \\ \varphi_1 \wedge \varphi_2 \end{array}}{\varphi_2} \wedge_{E,2}$$

Le due regole sono analoghe e quindi trattiamo solo la prima.

Ipotesi induttiva. Se Γ è un insieme che contiene tutte le ipotesi del sottoalbero con conclusione $\varphi_1 \wedge \varphi_2$, allora $\Gamma \models \varphi_1 \wedge \varphi_2$.

Tesi. Se Γ contiene tutte le ipotesi dell'albero con conclusione φ_1 , allora $\Gamma \models \varphi_1$.

Dimostrazione della tesi. Sia allora Γ un insieme che contiene tutte le ipotesi dell'albero con conclusione φ_1 . Chiaramente, Γ contiene tutte le ipotesi del sottoalbero con conclusione $\varphi_1 \wedge \varphi_2$. Dall'ipotesi induttiva, possiamo concludere che $\Gamma \models \varphi_1 \wedge \varphi_2$. Da questo segue immediatamente che $\Gamma \models \varphi_1$, come volevasi dimostrare.

Se l'ultima regola applicata è l'introduzione dell'implicazione, l'albero ha la forma:

$$\frac{\begin{array}{c} [\varphi]_n \\ \vdots \\ \psi \end{array}}{\varphi \Rightarrow \psi} \Rightarrow_{I,n}$$

Ipotesi induttiva. Se Γ contiene tutte le ipotesi dell'albero

$$\begin{array}{c} \varphi \\ \vdots \\ \psi \end{array} \quad (*)$$

allora $\Gamma \models \psi$.

Tesi. Se Γ' contiene tutte le ipotesi dell'albero con conclusione $\varphi \Rightarrow \psi$, allora $\Gamma' \models \varphi \Rightarrow \psi$.

Dimostrazione della tesi. Sia Γ' un insieme che contiene tutte le ipotesi dell'albero con conclusione $\varphi \Rightarrow \psi$. Per assurdo, si supponga che $\Gamma' \not\models (\varphi \Rightarrow \psi)$. Allora, esiste una valutazione v che soddisfa ogni formula in Γ' , ma che non soddisfa $\varphi \Rightarrow \psi$. Questo avviene se e solo se $v(\varphi) = 1$ e $v(\psi) = 0$. Si definisca allora

$$\Gamma =_{\text{def}} \Gamma' \cup \{\varphi\}.$$

Chiaramente, v soddisfa tutte le formule in Γ . Inoltre, Γ contiene tutte le ipotesi dell'albero in (*). Per l'ipotesi induttiva, si ha che $\Gamma \models \psi$ e quindi che $v(\psi) = 1$, una contraddizione.

Se l'ultima regola applicata è l'eliminazione dell'implicazione, l'albero ha la forma:

$$\frac{\begin{array}{c} \vdots \\ \varphi \Rightarrow \psi \end{array} \quad \begin{array}{c} \vdots \\ \varphi \end{array}}{\psi} \Rightarrow_E$$

Ipotesi induttive.

- Se Γ' contiene tutte le ipotesi del sottoalbero con conclusione $\varphi \Rightarrow \psi$, allora $\Gamma' \vDash \varphi \Rightarrow \psi$.
- Se Γ'' contiene tutte le ipotesi del sottoalbero con conclusione φ , allora $\Gamma'' \vDash \varphi$.

Tesi. Se Γ contiene tutte le ipotesi dell'albero con conclusione ψ , allora $\Gamma \vDash \psi$.

Dimostrazione della tesi. Se Γ contiene tutte le ipotesi dell'albero con conclusione ψ , allora contiene anche tutte le ipotesi del sottoalbero con conclusione $\varphi \Rightarrow \psi$ e del sottoalbero con conclusione φ . Dall'ipotesi induttiva, ne segue che $\Gamma \vDash \varphi \Rightarrow \psi$ e $\Gamma \vDash \varphi$. Per verificare che $\Gamma \vDash \psi$, si consideri una valutazione v che soddisfa tutte le formule in Γ . Da $\Gamma \vDash \varphi \Rightarrow \psi$, segue che $v(\varphi \Rightarrow \psi) = 1$, e da $\Gamma \vDash \varphi$ segue che $v(\varphi) = 1$. Deve quindi valere che $v(\psi) = 1$. Se non lo fosse, avremmo

$$v(\varphi \Rightarrow \psi) = (1 \Rightarrow 0) = 0,$$

una contraddizione.

Se l'ultima regola applicata è l'introduzione della disgiunzione, l'albero ha la forma

$$\frac{\begin{array}{c} \vdots \\ \varphi_1 \end{array}}{\varphi_1 \vee \varphi_2} \vee_{I,1} \quad \frac{\begin{array}{c} \vdots \\ \varphi_2 \end{array}}{\varphi_1 \vee \varphi_2} \vee_{I,2}$$

Come nel caso dell'eliminazione della congiunzione, le due regole sono essenzialmente analoghe e quindi trattiamo solo quella di sinistra.

Ipotesi induttiva. Se Γ_1 contiene tutte le ipotesi del sottoalbero con conclusione φ_1 allora $\Gamma_1 \vDash \varphi_1$.

Tesi. Se Γ contiene tutte le ipotesi dell'albero con conclusione $\varphi_1 \vee \varphi_2$, allora $\Gamma \vDash \varphi_1 \vee \varphi_2$.

Dimostrazione della tesi. Chiaramente, se Γ contiene tutte le ipotesi dell'albero con conclusione $\varphi_1 \vee \varphi_2$, allora contiene anche tutte le ipotesi del sottoalbero con conclusione φ_1 . Dall'ipotesi induttiva, possiamo concludere che $\Gamma \vDash \varphi_1$. Da questo segue immediatamente $\Gamma \vDash \varphi_1 \vee \varphi_2$, come volevasi dimostrare.

Se l'ultima regola applicata è l'eliminazione della disgiunzione, l'albero ha la forma

$$\frac{\begin{array}{c} \vdots \\ \varphi_1 \vee \varphi_2 \end{array} \quad \begin{array}{c} [\varphi_1]_n \\ \vdots \\ \chi \end{array} \quad \begin{array}{c} [\varphi_2]_m \\ \vdots \\ \chi \end{array}}{\chi} \vee_{E,n,m}$$

Ipotesi induttive.

- Se Γ contiene tutte le ipotesi del sottoalbero con conclusione $\varphi_1 \vee \varphi_2$, allora $\Gamma \vDash \varphi_1 \vee \varphi_2$.

- Se Γ_1 contiene tutte le ipotesi del sottoalbero

$$\begin{array}{c} \varphi_1 \\ \vdots \\ \chi \end{array}$$

allora $\Gamma_1 \vDash \chi$.

- Se Γ_2 contiene tutte le ipotesi del sottoalbero

$$\begin{array}{c} \varphi_2 \\ \vdots \\ \chi \end{array}$$

allora $\Gamma_2 \vDash \chi$.

Tesi. Se Γ' contiene tutte le ipotesi dell'albero con conclusione χ allora $\Gamma' \vDash \chi$.

Dimostrazione della tesi. Se Γ' contiene tutte le ipotesi dell'albero con conclusione χ , allora contiene tutte le ipotesi del sottoalbero con conclusione $\varphi_1 \vee \varphi_2$ e quindi, per ipotesi induttiva, $\Gamma' \vDash \varphi_1 \vee \varphi_2$. Vogliamo dimostrare che $\Gamma' \vDash \chi$. Per assurdo, supponiamo che esista una valutazione v che soddisfa tutte le formule in Γ' , ma tale che $v(\chi) = 0$. Visto che $\Gamma' \vDash \varphi_1 \vee \varphi_2$, deve valere $v(\varphi_1) = 1$ o $v(\varphi_2) = 1$. Nel primo caso, definiamo

$$\Gamma_1 =_{\text{def}} \Gamma' \cup \{\varphi_1\}$$

Chiaramente, Γ_1 contiene tutte le ipotesi del sottoalbero

$$\begin{array}{c} \varphi_1 \\ \vdots \\ \chi \end{array}$$

Quindi, per ipotesi induttiva $\Gamma_1 \vDash \chi$. Ma sappiamo che v soddisfa ogni formula in Γ_1 e quindi $v(\chi) = 1$, una contraddizione. La dimostrazione nel secondo caso, in cui $v(\varphi_2) = 1$, è analoga.

Se l'ultima regola applicata è l'eliminazione del falso, l'albero ha la forma

$$\begin{array}{c} \vdots \\ \vdots \\ \perp \\ \hline \varphi \end{array}$$

Ipotesi induttiva. Se Γ contiene tutte le ipotesi del sottoalbero con conclusione \perp , allora $\Gamma \vDash \perp$.

Tesi. Se Γ contiene tutte le ipotesi dell'albero con conclusione φ , allora $\Gamma \vDash \varphi$.

Dimostrazione della tesi. Si supponga che Γ contiene tutte le ipotesi dell'albero con conclusione φ . Per assurdo, si supponga che $\Gamma \not\vDash \varphi$. Allora esiste una valutazione v che soddisfa tutte le formule in Γ , ma tale che $v(\varphi) = 0$. Visto che Γ contiene anche tutte le ipotesi del sottoalbero con conclusione \perp , per ipotesi induttiva vale che $\Gamma \vDash \perp$. Ne segue che $v(\perp) = 1$, una contraddizione.

Se l'ultima regola applicata è quella per dimostrazioni per assurdo, l'albero ha la forma

$$\frac{\begin{array}{c} [\neg\varphi]_n \\ \vdots \\ \perp \end{array}}{\varphi} \text{RAA}_n$$

Ipotesi induttiva. Se Γ contiene tutte le ipotesi dell'albero

$$\begin{array}{c} \neg\varphi \\ \vdots \\ \perp \end{array} \quad (*)$$

allora $\Gamma \vDash \perp$.

Tesi. Se Γ' contiene tutte le ipotesi dell'albero con conclusione φ , allora $\Gamma' \vDash \varphi$.

Dimostrazione della tesi. Supponiamo che Γ' contenga tutte le ipotesi dell'albero con conclusione φ . Per assurdo, supponiamo che $\Gamma' \not\vDash \varphi$. Allora, esiste una valutazione v che soddisfa tutte le formule in Γ' , ma tale che $v(\varphi) = 0$. Quindi, $v(\neg\varphi) = 1$. Definiamo allora

$$\Gamma =_{\text{def}} \Gamma' \cup \{\neg\varphi\}.$$

È chiaro che Γ contiene le ipotesi dell'albero in (*) e che v soddisfa le formule in Γ . Per l'ipotesi induttiva, abbiamo $v(\perp) = 1$, una contraddizione.

La dimostrazione del teorema di validità è completa. □

Corollario 4.2. Se φ è un teorema, allora φ è una tautologia.

Dimostrazione. Caso speciale del teorema di validità, con $\Gamma = \emptyset$ □

5 Il teorema di completezza

Definizione 5.1. Sia Γ un insieme di formule. Diremo che Γ è *consistente* se soddisfa una delle seguenti condizioni equivalenti.

- (i) $\Gamma \not\vdash \perp$.
- (ii) Non esiste φ tale che $\Gamma \vdash \varphi$ e $\Gamma \vdash \neg\varphi$.
- (iii) Esiste almeno una formula φ tale che $\Gamma \not\vdash \varphi$.

Se un insieme di formule Γ non è consistente, lo chiameremo *inconsistente*.

Lemma 5.2. Sia Γ un insieme di formule e φ una formula.

- (i) Se $\Gamma \cup \{\neg\varphi\}$ è inconsistente, allora $\Gamma \vdash \varphi$.
- (ii) Se $\Gamma \cup \{\varphi\}$ è inconsistente, allora $\Gamma \vdash \neg\varphi$.

Dimostrazione.

(i) Se $\Gamma \cup \{\neg\varphi\}$ è inconsistente, allora esiste un albero di derivazione

$$\begin{array}{c} \neg\varphi \\ \vdots \\ \perp \end{array}$$

con ipotesi contenute in $\Gamma \cup \{\neg\varphi\}$. Quindi, possiamo estendere l'albero di derivazione con un'applicazione della legge per le dimostrazioni per assurdo:

$$\begin{array}{c} [\neg\varphi]_n \\ \vdots \\ \perp \\ \hline \varphi \\ \text{RAA}_n \end{array}$$

Quindi $\Gamma \vdash \varphi$.

(ii) Se $\Gamma \cup \{\varphi\}$ è inconsistente, allora esiste una derivazione

$$\begin{array}{c} \varphi \\ \vdots \\ \perp \end{array}$$

Possiamo quindi estendere questo albero con un'applicazione della legge per l'introduzione dell'implicazione:

$$\begin{array}{c} [\varphi]_n \\ \vdots \\ \perp \\ \hline \neg\varphi \\ \Rightarrow_{I,n} \end{array}$$

Quindi $\Gamma \vdash \neg\varphi$.

□

Definizione 5.3. Sia Γ un insieme di formule. Diremo che Γ è *massimamente consistente* se Γ è consistente e, per ogni insieme consistente di formule Γ' , se $\Gamma \subseteq \Gamma'$ allora $\Gamma = \Gamma'$.

Lemma 5.4. Sia Γ un insieme di formule massimalmente consistente e φ una formula. Se $\Gamma \vdash \varphi$ allora $\varphi \in \Gamma$.

Dimostrazione. Si supponga che $\Gamma \vdash \varphi$. Per assurdo, si supponga che $\varphi \notin \Gamma$. Visto che Γ è massimamente consistente, si ha che $\Gamma \cup \{\varphi\}$ è inconsistente. Quindi $\Gamma \vdash \neg\varphi$. Da questo segue che Γ è inconsistente, una contraddizione. □

Lemma 5.5. Sia Γ massimalmente consistente e φ una formula.

(i) $\varphi \in \Gamma$ oppure $\neg\varphi \in \Gamma$.

(ii) $\varphi \in \Gamma$ se e solo se $\neg\varphi \notin \Gamma$.

Dimostrazione. Per dimostrare (i), definiamo $\Gamma' =_{\text{def}} \Gamma \cup \{\varphi\}$. Se Γ' è inconsistente, allora il Lemma 5.2 implica che $\Gamma \vdash \neg\varphi$. Applicando il Lemma 5.4 si ha poi che $\neg\varphi \in \Gamma$. Se invece Γ' è consistente, allora deve valere che $\Gamma' = \Gamma$, per la massimalità di Γ , da cui segue che $\varphi \in \Gamma$. Per dimostrare (ii), dimostriamo le due implicazioni separatamente. Se $\varphi \in \Gamma$, allora non può valere che $\neg\varphi \in \Gamma$, visto che in quel caso Γ sarebbe inconsistente. Se $\neg\varphi \notin \Gamma$, allora deve valere che $\varphi \in \Gamma$. Infatti, se non valesse avremmo una contraddizione con l'enunciato in (i). \square

Lemma 5.6. Sia Γ un insieme di formule massimamente consistente. Per ogni φ e ψ , le seguenti asserzioni sono equivalenti.

(i) $(\varphi \Rightarrow \psi) \in \Gamma$.

(ii) Se $\varphi \in \Gamma$ allora $\psi \in \Gamma$.

Dimostrazione. Per dimostrare (i) \Rightarrow (ii), supponiamo $(\varphi \Rightarrow \psi) \in \Gamma$ e $\varphi \in \Gamma$. Ma allora $\Gamma \vdash \varphi \Rightarrow \psi$ e $\Gamma \vdash \varphi$, da cui segue $\Gamma \vdash \psi$. Visto che Γ è massimalmente consistente, possiamo applicare il Lemma 5.4 e dedurre $\psi \in \Gamma$, come volevasi dimostrare. Per dimostrare (ii) \Rightarrow (i), supponiamo che se $\varphi \in \Gamma$ allora $\psi \in \Gamma$. Distinguiamo due casi: se $\varphi \in \Gamma$, allora per l'ipotesi vale che $\psi \in \Gamma$, da cui segue $(\varphi \Rightarrow \psi) \in \Gamma$. Se invece $\varphi \notin \Gamma$, allora $\neg\varphi \in \Gamma$. Da questo segue $\Gamma \vdash \neg\varphi$. Ma allora vale $\Gamma \vdash \varphi \Rightarrow \psi$ e quindi vale $(\varphi \Rightarrow \psi) \in \Gamma$ per il Lemma 5.4. \square

Lemma 5.7. Per ogni insieme consistente Γ esiste un insieme massimalmente consistente Γ^* tale che $\Gamma \subseteq \Gamma^*$.

Dimostrazione. Visto che l'insieme delle formule atomiche è numerabile, anche l'insieme di tutte le formule lo è. Fissiamo quindi un'enumerazione delle formule

$$\varphi_0, \varphi_1, \dots, \varphi_n, \varphi_{n+1}, \dots$$

Definiamo una famiglia di insiemi di formule $(\Gamma_n \mid n \in \mathbb{N})$ nel modo seguente:

$$\begin{aligned} \Gamma_0 &=_{\text{def}} \Gamma \\ \Gamma_{n+1} &=_{\text{def}} \begin{cases} \Gamma_n \cup \{\varphi_{n+1}\} & \text{se } \Gamma_n \cup \{\varphi_{n+1}\} \text{ è consistente,} \\ \Gamma_n & \text{altrimenti.} \end{cases} \end{aligned}$$

Si noti che Γ_n è consistente per ogni $n \in \mathbb{N}$. Ora definiamo

$$\Gamma^* =_{\text{def}} \bigcup_{n \in \mathbb{N}} \Gamma_n.$$

Chiaramente, $\Gamma \subseteq \Gamma^*$. Per dimostrare che Γ^* è consistente, procediamo per assurdo. Supponiamo che $\Gamma^* \vdash \perp$. Allora, esiste un albero di derivazione con ipotesi non scaricate $\psi_1, \dots, \psi_m \in \Gamma^*$ e conclusione \perp . Per $1 \leq i \leq m$, esiste $n_i \in \mathbb{N}$ tale che $\psi_i \in \Gamma_{n_i}$. Definiamo $n =_{\text{def}} \max(n_i \mid 1 \leq i \leq m)$. Da questo segue che $\psi_1, \dots, \psi_m \in \Gamma_n$ e quindi che $\Gamma_n \vdash \perp$, una contraddizione. Infine, dimostriamo che Γ^* è massimalmente consistente. Sia Δ consistente e tale

che $\Gamma^* \subseteq \Delta$. Dimostriamo che $\Delta \subseteq \Gamma^*$. Sia $\varphi \in \Delta$. Dall'esistenza di una enumerazione di tutte le formule, segue che esiste $n \in \mathbb{N}$ tale che $\varphi = \varphi_n$. Visto che $\Gamma_n \subseteq \Gamma^* \subseteq \Delta$ e che Δ è consistente, abbiamo che $\Gamma_n \cup \{\varphi_n\}$ è consistente. Quindi $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$, da cui segue che $\varphi_n \in \Gamma^*$. \square

Lemma 5.8 (Lemma di Esistenza di Valutazioni). Sia Γ un insieme di formule. Se Γ è consistente allora esiste una valutazione che soddisfa ogni formula di Γ .

Dimostrazione. Supponiamo che Γ sia consistente. Per il Lemma 5.7, esiste un insieme massimamente consistente Γ^* tale che $\Gamma \subseteq \Gamma^*$. La dimostrazione si conclude in tre passi. Per il primo passo, definiamo una valutazione v fissando

$$v(p) =_{\text{def}} \begin{cases} 1 & \text{se } p \in \Gamma^*, \\ 0 & \text{altrimenti.} \end{cases}$$

Per il secondo passo, dimostriamo per induzione che, per ogni formula φ , vale $v(\varphi) = 1$ se e solo se $\varphi \in \Gamma^*$.

Caso base. Per le formula atomiche $p \in P$, vale che $v(p) = 1$ se e solo se $p \in \Gamma^*$ per definizione di v .

Primo caso induttivo. Date formule φ e ψ , dobbiamo dimostrare che $v(\varphi \wedge \psi) = 1$ se e solo se $v(\varphi) = 1$ e $v(\psi) = 1$. Per ipotesi induttiva, questo vale se e solo se $\varphi \in \Gamma^*$ e $\psi \in \Gamma^*$. Per il Lemma 5.4, questo vale se e solo se $\varphi \wedge \psi \in \Gamma^*$.

Secondo caso induttivo. Date formule φ e ψ , dobbiamo dimostrare che $v(\varphi \vee \psi) = 1$ se e solo se $\varphi \vee \psi \in \Gamma^*$. Abbiamo che $v(\varphi \vee \psi) = 1$ se e solo se $v(\varphi) = 1$ o $v(\psi) = 1$, il che vale se e solo se $\varphi \in \Gamma^*$ o $\psi \in \Gamma^*$, per ipotesi induttiva. Rimane quindi da dimostrare che $\varphi \vee \psi \in \Gamma^*$ se e solo se $\varphi \in \Gamma^*$ o $\psi \in \Gamma^*$. Ma questo segue dal primo caso induttivo, utilizzando il fatto che $\varphi \vee \psi \Leftrightarrow \neg(\neg\varphi \wedge \neg\psi)$ è un teorema.

Terzo caso induttivo. Date formule φ e ψ , dobbiamo dimostrare che $v(\varphi \Rightarrow \psi) = 0$ se e solo se $(\varphi \Rightarrow \psi) \notin \Gamma^*$. Si ha che $v(\varphi \Rightarrow \psi) = 0$ se e solo se $v(\varphi) = 1$ e $v(\psi) = 0$. Per ipotesi induttiva, questo vale se e solo se $\varphi \in \Gamma^*$ e $\psi \notin \Gamma^*$. Per il Lemma 5.6, questo vale se e solo se $(\varphi \Rightarrow \psi) \notin \Gamma^*$.

Per il terzo e ultimo passo, dimostriamo che v soddisfa tutte le formule in Γ . Avendo dimostrato che $v(\varphi) = 1$ se e solo se $\varphi \in \Gamma^*$, dal fatto che $\Gamma \subseteq \Gamma^*$ segue che $v(\varphi) = 1$ per ogni $\varphi \in \Gamma$, come volevasi dimostrare. \square

Teorema 5.9 (Teorema di Completezza). Se $\Gamma \models \varphi$ allora $\Gamma \vdash \varphi$.

Dimostrazione. Dimostriamo che se $\Gamma \not\models \varphi$ allora $\Gamma \not\vdash \varphi$. Supponiamo che $\Gamma \not\models \varphi$. Da questo segue che $\Gamma \cup \{\neg\varphi\}$ è consistente. Per il Lemma di Esistenza di Valutazioni, esiste una valutazione v che soddisfa tutte le formule in Γ ma che non soddisfa φ . Questo implica che $\Gamma \not\vdash \varphi$. \square

6 Esercizi

Esercizio 6.1. Costruendo appropriati alberi di deduzione, si dimostri che le seguenti formule sono teoremi della logica proposizionale.

- (i) $((\varphi \Rightarrow \psi_1) \vee (\varphi \Rightarrow \psi_2)) \Rightarrow (\varphi \Rightarrow \psi_1 \vee \psi_2)$
- (ii) $(\varphi \wedge \psi \Rightarrow \chi) \Leftrightarrow (\varphi \Rightarrow (\psi \Rightarrow \chi))$
- (iii) $(\varphi \Rightarrow \psi) \vee (\psi \Rightarrow \varphi)$
- (iv) $(\varphi \Rightarrow (\psi \wedge \chi)) \Rightarrow (\varphi \Rightarrow (\psi \vee \chi))$
- (v) $(\varphi \Rightarrow (\psi \vee \chi)) \Rightarrow ((\varphi \wedge \neg\psi) \Rightarrow \chi)$
- (vi) $((\varphi \vee \psi) \Rightarrow \neg\chi) \Rightarrow (\chi \Rightarrow \neg\varphi \wedge \neg\psi)$
- (vii) $\neg(\varphi \Rightarrow \psi) \Rightarrow (\varphi \wedge \neg\psi)$
- (viii) $(\varphi \Rightarrow \neg(\psi \wedge \chi)) \Rightarrow ((\varphi \wedge \chi) \Rightarrow \neg\psi)$
- (ix) $(\varphi \Rightarrow \neg\psi) \Rightarrow (\psi \wedge \neg\varphi).$

Esercizio 6.2. Costruendo opportune tavole di verità, si stabilisca se le formule seguenti sono tautologie della logica proposizionale.

- (i) $(p \vee q) \Rightarrow (p \wedge q)$
- (ii) $(p \Rightarrow (q \vee \neg r)) \Rightarrow (p \Rightarrow (q \wedge r))$
- (iii) $(p \Rightarrow (q \wedge \neg r)) \Rightarrow (p \Rightarrow (q \vee r))$

Esercizio 6.3. Si stabilisca quali dei seguenti insiemi di formule sono consistenti.

- (i) $\{\neg p_1 \wedge p_2 \Rightarrow p_0, p_1 \Rightarrow (\neg p_0 \Rightarrow p_2), p_0 \Leftrightarrow \neg p_2\}.$
- (ii) $\{p_0 \Rightarrow p_1, p_1 \Rightarrow p_2, p_2 \Rightarrow p_3, p_3 \Rightarrow \neg p_0\}.$
- (iii) $\{p \Rightarrow q, q \vee r \Rightarrow \neg p, p \wedge r\}.$

Esercizio 6.4.

- (i) Si dimostri che per ogni insieme consistente Γ esiste un insieme inconsistente Γ' tale che $\Gamma \subseteq \Gamma'$.
- (ii) Si dimostri che se Γ è consistente, allora ogni sottoinsieme $\Gamma' \subseteq \Gamma$ è consistente.
- (iii) Si dia un esempio di insiemi consistenti Γ_1, Γ_2 tali che $\Gamma_1 \cup \Gamma_2$ non sia consistente.

Esercizio 6.5.

- (i) Si enunci la definizione di teorema.

- (ii) Si enunci la definizione di tautologia.
- (iii) È possibile che una formula sia una tautologia, ma non un teorema? Si giustifichi la risposta.

Esercizio 6.6.

- (i) Si dia un esempio di insiemi Γ_1 e Γ_2 e di una formula φ diversa da \perp tali che $\Gamma_1 \not\vdash \varphi$, $\Gamma_2 \not\vdash \varphi$, ma $\Gamma_1 \cup \Gamma_2 \vdash \varphi$.
- (ii) Si dimostri che se $\Gamma_1 \vdash \varphi_1$ e $\Gamma_2 \vdash \varphi_2$ allora $\Gamma_1 \cup \Gamma_2 \vdash \varphi_1 \wedge \varphi_2$.

Esercizio 6.7. Si dimostri che le seguenti affermazioni sono equivalenti.

- (i) $\{\varphi, \psi\}$ è consistente.
- (ii) $\not\vdash \neg(\varphi \wedge \psi)$.
- (iii) $\not\vdash \varphi \Rightarrow \neg\psi$.

Capitolo 2

Logica del primo ordine

1 Linguaggi del primo ordine

Definizione 1.1. Un linguaggio del primo ordine L consiste dei seguenti dati:

- (i) un insieme di variabili (x, y, z, \dots) , Var_L ,
- (ii) un insieme di costanti $(a, b, c \dots)$
- (iii) per ogni $n \in \mathbb{N}$, un insieme di simboli di predicato n -ario (P, Q, R, \dots)
- (iv) per ogni $n \in \mathbb{N}$, un insieme di simboli di funzione n -aria (f, g, h, \dots) .

D'ora in poi considereremo un linguaggio del primo ordine L fissato e faremo l'assunzione che L sia numerabile, ovvero che ciascuno degli insiemi di variabili, costanti, simboli di funzioni e simboli di predicato sia numerabile. L'insieme dei *termini* di L è definito come il piú piccolo insieme X che soddisfa le proprietà seguenti:

- (i) se x è una variabile di L , allora $x \in X$,
- (ii) se a è una costante di L , allora $a \in X$,
- (iii) se $t_1, \dots, t_n \in X$ e f è un simbolo di funzione n -aria, allora $f(t_1, \dots, t_n) \in X$.

L'insieme delle *formule* associato ad L è il piú piccolo insieme X che soddisfa le proprietà seguenti:

- (i) $\perp, \top \in X$,
- (ii) se t_1, \dots, t_n sono termini e P è un simbolo per predicato n -ario, allora $P(t_1, \dots, t_n) \in X$,
- (iii) se t_1 e t_2 sono termini, allora $(t_1 = t_2) \in X$
- (iv) se $\varphi, \psi \in X$, allora $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \Rightarrow \psi) \in X$,
- (v) se $\varphi \in X$ e x è una variabile, allora $(\forall x)\varphi \in X$ e $(\exists x)\varphi \in X$.

Non enunceremo i principi di induzione e di ricorsione associate all'insieme dei termini e dell'insieme delle formule, ma li utilizzeremo in modo analogo a quanto fatto nel caso della logica proposizionale.

Sia t un termine di L . L'insieme $FV(t)$ delle *variabili libere* di t è definito ricorsivamente tramite le clausole seguenti.

- $FV(x) = \{x\}$, se x è una variabile.
- $FV(a) = \emptyset$, se a è una costante.
- $FV(f(t_1, \dots, t_n)) = FV(t_1) \cup \dots \cup FV(t_n)$, se f è un simbolo per funzione n -aria e t_1, \dots, t_n sono termini.

Sia φ una formula di L . L'insieme $FV(\varphi)$ delle variabili libere di φ è definito ricorsivamente tramite le clausole seguenti.

- $FV(\perp) = FV(\top) = \emptyset$.
- $FV(P(t_1, \dots, t_n)) = FV(t_1) \cup \dots \cup FV(t_n)$.
- $FV(t_1 = t_2) = FV(t_1) \cup FV(t_2)$.
- $FV(\varphi * \psi) = FV(\varphi) \cup FV(\psi)$, ove $*$ $\in \{\wedge, \vee, \Rightarrow\}$.
- $FV((\forall x)\varphi) = FV(\varphi) \setminus \{x\}$ e $FV((\exists x)\varphi) = FV(\varphi) \setminus \{x\}$.

Una formula φ è detta *chiusa* se $FV(\varphi) = \emptyset$.

Definizione 1.2. Una *teoria del primo ordine* consiste in un linguaggio del primo ordine L ed un insieme (non necessariamente finito) di formule chiuse in L , dette gli *assiomi* della teoria.

Esempio 1.3. Diamo quattro esempi di teorie del primo ordine: l'aritmetica di Peano, la teoria degli insiemi di Zermelo-Fraenkel, la teoria dei gruppi e la teoria degli ordini parziali.

- (i) Il linguaggio dell'aritmetica di Peano contiene, oltre a variabili, una costante 0 (zero), due simboli per funzioni binarie $+$ e \cdot (addizione e moltiplicazione), e un simbolo di funzione unaria S (successore). Per esempio, abbiamo termini

$$0 + S(0), \quad S(S(0)), \quad S(0) \cdot S(S(0)), \quad x + y,$$

e formule

$$x = S(0), \quad S(S(0)) + 0 = 0, \quad (\exists x)(x + 0 = y).$$

Gli assiomi di PA sono i seguenti:

$$\begin{aligned} & \forall x(Sx \neq 0), \\ & \forall x \forall y(Sx = Sy \Rightarrow x = y), \\ & \forall x(x + 0 = x), \\ & \forall x \forall y(x + Sy = S(x + y)), \\ & \forall x(x \cdot 0 = 0), \\ & \forall x \forall y(x \cdot Sy = x \cdot y + x), \\ & \varphi(0) \wedge \forall x(\varphi(x) \Rightarrow \varphi(Sx)) \Rightarrow \forall x \varphi(x). \end{aligned}$$

L'ultimo assioma è in realtà uno *schema*, ovvero una famiglia infinita di assiomi, uno per ogni formula $\varphi(x)$ del linguaggio di PA.

- (ii) Il linguaggio della teoria degli insiemi di Zermelo-Fraenkel contiene solo variabili e un simbolo per un predicato binario \in (appartenza). Le espressioni

$$(\exists x)(\forall y)(y \in x \Leftrightarrow \perp), \quad x \in y \wedge x \in z \Rightarrow x \in u$$

sono formule. Gli assiomi della teoria degli insiemi di Zermelo-Fraenkel saranno presentati nel Capitolo 4. Diamo solo un esempio, l'assioma della coppia:

$$(\forall x_1)(\forall x_2)(\exists u)(\forall x)(x \in u \Leftrightarrow x = x_1 \vee x = x_2).$$

- (iii) Il linguaggio della teoria dei gruppi contiene, oltre a variabili, una costante 1 (elemento neutro), un simbolo per funzione binaria \cdot (moltiplicazione) e un simbolo di funzione unaria $()^{-1}$ (inverso). Le espressioni

$$1 \cdot (x \cdot y), \quad x^{-1} \cdot x, \quad (x \cdot y) \cdot x^{-1}$$

sono termini, mentre le espressioni

$$x \cdot 1 = y, \quad (\forall x)(\forall y)(x \cdot y = y \cdot x)$$

sono formule. Gli assiomi della teoria dei gruppi sono i seguenti:

$$\begin{aligned} & (\forall x)(\forall y)(\forall z)(x \cdot (y \cdot z) = (x \cdot y) \cdot z), \\ & (\forall x)((x \cdot 1 = x) \wedge (1 \cdot x = x)), \\ & (\forall x)((x \cdot x^{-1} = 1) \wedge (x^{-1} \cdot x = 1)). \end{aligned}$$

- (iv) Il linguaggio degli ordini parziali contiene oltre a variabili, solo un simbolo di predicato binario, \leq . In questo linguaggio, possiamo definire le seguenti formule:

$$\begin{aligned} x < y & \stackrel{\text{def}}{=} x \leq y \wedge x \neq y, \\ x > y & \stackrel{\text{def}}{=} y < x, \\ x \geq y & \stackrel{\text{def}}{=} y \leq x, \\ x \leq y \leq z & \stackrel{\text{def}}{=} x \leq y \wedge y \leq z. \end{aligned}$$

La teoria degli ordini parziali è data dai seguenti assiomi:

$$\begin{aligned} \forall x \forall y \forall z (x \leq y \leq z \Rightarrow x \leq z), \\ \forall x \forall y (x \leq y \leq x \Leftrightarrow x = y). \end{aligned}$$

2 Il calcolo della deduzione naturale

Introduciamo adesso il calcolo di deduzione naturale per la logica del primo ordine. Avremo regole per l'uguaglianza, regole per i connettivi proposizionali, e regole per i quantificatori. Le regole per i connettivi proposizionali sono le stesse della regole introdotte per la logica proposizionale e quindi non le ripeteremo. Per le regole per i quantificatori, come nel caso della logica proposizionale daremo, per ciascun quantificatore, regole di introduzione e regole di eliminazione. Come vedremo, le regole di deduzione naturale per i quantificatori coinvolgono anche condizioni che ne restringono l'applicazione.

Regole per l'uguaglianza. Ci sono cinque regole. Le prime tre regole esprimono che l'uguaglianza è una relazione riflessiva, simmetrica, e transitiva:

$$\frac{}{x = x} \qquad \frac{x = y}{y = x} \qquad \frac{x = y \quad y = z}{x = z}$$

Le ultime due regole determinano il comportamento dell'uguaglianza rispetto alla sostituzione:

$$\frac{x_1 = y_1 \quad \cdots \quad x_n = y_n}{t(x_1, \dots, x_n) = t(y_1, \dots, y_n)}$$

$$\frac{x_1 = y_1 \quad \cdots \quad x_n = y_n \quad \varphi(x_1, \dots, x_n)}{\varphi(y_1, \dots, y_n)}$$

Regole per il quantificatore universale. La regola di introduzione è

$$\frac{\varphi(x)}{\forall x \varphi(x)} \forall_I$$

Questa regola è soggetta alla condizione che la variabile x non sia una delle variabili libere delle ipotesi non scaricate nel sottoalbero con conclusione $\varphi(x)$. La regola di eliminazione per il quantificatore universale è

$$\frac{\forall x \varphi(x)}{\varphi(t)} \forall_E$$

Questa regola è soggetta alla condizione che il termine t sia *sostituibile* per x in $\varphi(x)$. Definiremo precisamente che cosa vuol dire che un termine t è sostituibile per una variabile x in una formula $\varphi(x)$ più avanti. Informalmente, questa condizione garantisce che nessuna delle variabili libere che compaiono in t venga legata (ovvero resa non libera) da uno dei quantificatori che compaiono

in $\varphi(x)$. Per il momento, ci basterà sapere che x è sempre sostituibile per x in $\varphi(x)$, cosicchè la seguente istanza della regola \forall_E è sempre corretta:

$$\frac{\forall x\varphi(x)}{\varphi(x)} \forall_E \quad (*)$$

Esempio 2.1. Diamo due esempi di applicazioni corrette delle regole.

- (i) Costruiamo un albero di derivazione la cui unica ipotesi non scaricata è $\forall x\forall y\varphi(x, y)$ e la cui conclusione è $\forall y\forall x\varphi(x, y)$. L'albero è costruito come segue:

$$\frac{\frac{\frac{\frac{\forall x\forall y\varphi(x, y)}{\forall y\varphi(x, y)} \forall_E}{\varphi(x, y)} \forall_E}{\forall x\varphi(x, y)} \forall_I}{\forall y\forall x\varphi(x, y)} \forall_I$$

Si noti come le applicazioni della regola \forall_I soddisfano la condizione necessaria per la loro applicazione, visto che le variabili x e y non compaiono come variabili libere nell'ipotesi non scaricata dell'albero. La regola \forall_E è stata poi applicata solo nel caso speciale indicato in (*).

- (ii) Costruiamo un albero con conclusione la formula $\forall x\varphi(x) \wedge \forall x\psi(x)$ e la cui unica ipotesi non scaricata è la formula $\forall x(\varphi(x) \wedge \psi(x))$. L'albero è costruito come segue:

$$\frac{\frac{\frac{\frac{\forall x(\varphi(x) \wedge \psi(x))}{\varphi(x) \wedge \psi(x)} \forall_E}{\varphi(x)} \forall_I}{\forall x\varphi(x)} \forall_I \quad \frac{\frac{\frac{\forall x(\varphi(x) \wedge \psi(x))}{\varphi(x) \wedge \psi(x)} \forall_E}{\psi(x)} \forall_I}{\forall x\psi(x)} \forall_I}{\forall x\varphi(x) \wedge \forall x\psi(x)} \wedge_I$$

Esempio 2.2. Diamo due esempi di applicazioni *scorrette* delle regole.

- (i) Nel linguaggio dell'aritmetica di Peano, l'applicazione

$$\frac{x = 0}{\forall x(x = 0)}$$

della regola \forall_I non è corretta, in quanto la variabile x compare come variabile libera nella formula $x = 0$, che non è scaricata.

- (ii) L'applicazione

$$\frac{\forall x\neg\forall y(x = y)}{\neg\forall y(y = y)}$$

della regola \forall_E non è corretta. Abbiamo applicato la regola \forall_E con la formula $\neg\forall y(x = y)$ al posto della formula $\varphi(x)$ e y al posto del termine t .

Secondo la definizione che daremo piú avanti, tuttavia, y non è sostituibile per x in $\neg\forall y(x = y)$. Informalmente, questo avviene perchè la variabile y appare tra le variabili quantificate di $\neg\forall y(x = y)$ e quindi se sostituiamo y per x in questa formula otteniamo che y viene ‘legata’ dal quantificatore $\forall y$. Anche se non abbiamo introdotto ancora la nozione di validità di una formula, anticipiamo che l’ipotesi non scaricata dell’albero è valida in ogni struttura con almeno due elementi distinti, mentre la conclusione non è valida in una tale struttura.

Regole per il quantificatore esistenziale. La regola di introduzione è

$$\frac{\varphi(t)}{\exists x\varphi(x)} \exists_I$$

Questa regola è soggetta alla condizione che il termine t sia sostituibile per x in $\varphi(x)$. La regola di eliminazione è

$$\frac{\begin{array}{c} [\varphi(x)]_* \\ \vdots \\ \psi \end{array}}{\exists x\varphi(x)} \frac{\psi}{\psi} \exists_{E,*}$$

Questa regola è soggetta alla condizione che la variabile x non sia una delle variabili libere nè di ψ nè delle ipotesi non scaricate nel sottoalbero con conclusione ψ tranne $\varphi(x)$.

Esempio 2.3. Costruiamo un albero con conclusione $\exists x\varphi(x) \vee \exists x\psi(x)$ e con unica ipotesi non scaricata $\exists x(\varphi(x) \vee \psi(x))$. L’albero è il seguente:

$$\frac{\frac{\frac{[\varphi(x)]_1}{\exists x\varphi(x)} \quad \frac{[\psi(x)]_2}{\exists x\psi(x)}}{[\varphi(x) \vee \psi(x)]_3} \quad \frac{\exists x\varphi(x) \vee \exists x\psi(x)}{\exists x\varphi(x) \vee \exists x\psi(x)} \vee_{E,1,2}}{\exists x(\varphi(x) \vee \psi(x))} \exists_{E,3}$$

Si noti che l’applicazione della regola \exists_E è corretta in quanto le ipotesi $\varphi(x)$ e $\psi(x)$ che potrebbero contenere la x tra le loro variabili libere sono state scaricate nell’applicazione della regola \vee_E , e quindi non sono piú tra le ipotesi non scaricate del sottoalbero con conclusione $\exists x\varphi(x) \vee \exists x\psi(x)$.

Le nozioni di albero di derivazione, di derivabilità di una formula φ da un insieme di formule Γ e di teorema sono definite come per la logica proposizionale. Useremo notazione identica a quella introdotta precedentemente. In particolare, scriveremo $\Gamma \vdash \varphi$ per indicare che esiste un albero di derivazione con conclusione φ e le cui ipotesi non scaricate sono contenute in Γ .

Concludiamo questa sezione definendo che cosa vuol dire che un termine t è *sostituibile* per una variabile x in una formula φ . Per ricorsione su φ , definiamo:

- t è sempre sostituibile per x in $P(t_1, \dots, t_n)$.
- t è sostituibile per x in $\varphi * \psi$, ove $*$ $\in \{\wedge, \vee, \Rightarrow\}$, se t è sostituibile per x in φ e ψ .
- t è sostituibile per x in $\forall y\varphi$ e $\exists y\varphi$ se vale almeno una delle seguenti due possibilità:
 - x non compare come variabile libera in $(\forall y)\varphi$ e $\exists y\varphi$.
 - y non compare come variabile libera in t e t è sostituibile per x in φ .

3 Strutture

Definizione 3.1. Una *struttura* per L consiste dei seguenti dati.

- Un insieme non vuoto D , detto il *dominio* della struttura.
- Una funzione che assegna ad ogni costante a di L un elemento $\llbracket a \rrbracket \in D$.
- Una funzione che assegna ad ogni simbolo di funzione n -ario f di L una funzione

$$\llbracket f \rrbracket : \underbrace{D \times \dots \times D}_{n \text{ volte}} \rightarrow D$$

- Una funzione che assegna ad ogni simbolo di predicato n -ario P di L un sottoinsieme

$$\llbracket P \rrbracket \subseteq \underbrace{D \times \dots \times D}_{n \text{ volte}} .$$

Definizione 3.2. Sia D una struttura per L . Una *assegnazione* per le variabili di L in D è una funzione $s : \text{Var}_L \rightarrow D$.

È utile introdurre la seguente notazione per modificare il valore di una valutazione su di una variabile. Data una valutazione $s : \text{Var}_L \rightarrow D$, $x \in \text{Var}_L$, e $d \in D$, definiamo una nuova valutazione

$$s[x \mapsto d] : \text{Var}_L \rightarrow D$$

nel modo seguente. Il valore di questa valutazione su x è d , ovvero:

$$s[x \mapsto d](x) =_{\text{def}} d ,$$

mentre il valore su di ogni altra variabile y è lo stesso di s , ovvero:

$$s[x \mapsto d](y) =_{\text{def}} s(y) .$$

Fissiamo ora una struttura D per L , e un'assegnazione $s : \text{Var}_L \rightarrow D$. Dato un termine t di L , definiamo l'elemento $\llbracket t \rrbracket_s \in D$, detto l'*interpretazione* di t in D relativa ad s , tramite la seguente definizione ricorsiva:

$$\begin{aligned} \llbracket x \rrbracket_s &=_{\text{def}} s(x) \\ \llbracket a \rrbracket_s &=_{\text{def}} \llbracket a \rrbracket \\ \llbracket f(t_1, \dots, t_n) \rrbracket_s &=_{\text{def}} \llbracket f \rrbracket(\llbracket t_1 \rrbracket_s, \dots, \llbracket t_n \rrbracket_s) \end{aligned}$$

Usando questa definizione e la notazione introdotta precedentemente, data una formula φ di L , definiamo l'elemento $\llbracket \varphi \rrbracket_s \in \{0, 1\}$, detto il *valore di verità* di φ in D relativo ad s , tramite la seguente definizione ricorsiva:

$$\llbracket P(t_1, \dots, t_n) \rrbracket_s =_{\text{def}} \begin{cases} 1 & \text{se } (\llbracket t_1 \rrbracket_s, \dots, \llbracket t_n \rrbracket_s) \in \llbracket P \rrbracket_s \\ 0 & \text{altrimenti} \end{cases}$$

$$\llbracket t_1 = t_2 \rrbracket_s =_{\text{def}} \begin{cases} 1 & \text{se } \llbracket t_1 \rrbracket_s = \llbracket t_2 \rrbracket_s \\ 0 & \text{altrimenti} \end{cases}$$

$$\llbracket \varphi \wedge \psi \rrbracket_s =_{\text{def}} \llbracket \varphi \rrbracket_s \wedge \llbracket \psi \rrbracket_s$$

$$\llbracket \varphi \vee \psi \rrbracket_s =_{\text{def}} \llbracket \varphi \rrbracket_s \vee \llbracket \psi \rrbracket_s$$

$$\llbracket \varphi \Rightarrow \psi \rrbracket_s =_{\text{def}} \llbracket \varphi \rrbracket_s \Rightarrow \llbracket \psi \rrbracket_s$$

$$\llbracket \forall x \varphi(x) \rrbracket_s =_{\text{def}} \begin{cases} 1 & \text{se per ogni } d \in D \text{ vale che } \llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1, \\ 0 & \text{altrimenti.} \end{cases}$$

$$\llbracket \exists x \varphi(x) \rrbracket_s =_{\text{def}} \begin{cases} 1 & \text{se esiste } d \in D \text{ tale che } \llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1, \\ 0 & \text{altrimenti.} \end{cases}$$

È utile notare che se φ è una formula chiusa, ovvero senza variabili libere, il suo valore di verità non dipende dall'assegnazione s . In questo caso, scriveremo quindi semplicemente $\llbracket \varphi \rrbracket$ anziché $\llbracket \varphi \rrbracket_s$.

Definizione 3.3.

- Sia φ una formula, D una struttura per L e s un'assegnazione. Diremo che la coppia (D, s) *soddisfa* φ se vale che $\llbracket \varphi \rrbracket_s = 1$.
- Sia Γ un insieme di formule e sia φ una formula. Diremo che Γ *implica semanticamente* φ se ogni coppia (D, s) che soddisfa tutte le formule in Γ soddisfa anche φ . Scriveremo $\Gamma \models \varphi$ per indicare che Γ implica semanticamente φ .
- Sia φ una formula. Diremo che φ è una *tautologia* se ogni coppia (D, s) soddisfa φ . Scriveremo $\models \varphi$ per indicare che φ è una tautologia.

Le nozioni introdotte nella Definizione 3.3 possono essere semplificate nel caso si tratti di formule chiuse. Se φ è una formula chiusa, diremo che una struttura D soddisfa φ se vale che $\llbracket \varphi \rrbracket = 1$.

Osservazione 3.4. Sia Γ un insieme di formule chiuse, φ è una formula chiusa.

- Γ implica semanticamente φ se e solo se ogni struttura che soddisfa tutte le formule di Γ soddisfa anche φ .
- φ è una tautologia se e solo se ogni struttura D soddisfa φ .

4 Il teorema di validità

Lemma 4.1 (Lemma di Sostituzione). Sia $\varphi(x)$ una formula e t un termine. Per ogni struttura D e ogni $s : \text{Var}_L \rightarrow D$, vale che

$$\llbracket \varphi(x) \rrbracket_{s[x \mapsto \llbracket t \rrbracket_s]} = \llbracket \varphi(t) \rrbracket_s.$$

Traccia della dimostrazione. Consideriamo solo un caso particolare. Per esempio, si ha

$$\llbracket x = y \rrbracket_{s[x \mapsto \llbracket t \rrbracket_s]} = \llbracket t = y \rrbracket_s.$$

Infatti il valore di verità sulla sinistra è 1 se e solo se vale che

$$\llbracket x \rrbracket_{s[x \mapsto \llbracket t \rrbracket_s]} = \llbracket y \rrbracket_{s[x \mapsto \llbracket t \rrbracket_s]}.$$

Questo vale se e solo se $\llbracket t \rrbracket_s = \llbracket y \rrbracket_s$, e quest'ultima asserzione vale se e solo se $\llbracket t = y \rrbracket_s = 1$. \square

Osservazione 4.2. Nel seguito, faremo l'assunzione che se $\Gamma \vdash \forall x \varphi(x)$ allora la variabile x non compare tra le variabili libere di Γ . Infatti, anche quando questa assunzione non è verificata è sempre possibile cambiare nome alle variabili. Per esempio, si consideri

$$\{\forall x(P(x) \Rightarrow Q(x)), \forall x P(x), R(x)\} \vdash \forall x Q(x). \quad (*)$$

In questo caso, $x \in \text{FV}(\Gamma)$ in quanto $x \in \text{FV}(R(x))$. Tuttavia, l'asserzione in (*) è chiaramente equivalente a

$$\{\forall y(P(y) \Rightarrow Q(y)), \forall y P(y), R(x)\} \vdash \forall y Q(y).$$

E adesso abbiamo che $y \notin \text{FV}(\Gamma)$, come volevasi.

Teorema 4.3 (Teorema di Validità). Se $\Gamma \vdash \varphi$ allora $\Gamma \models \varphi$.

Dimostrazione. Si ricordi che $\Gamma \vdash \varphi$ significa che esiste un albero di derivazione il cui insieme di ipotesi non scaricate è contenuto in Γ e con conclusione φ . Per dimostrare il teorema di validità procederemo, come nel caso della logica proposizionale, per induzione sull'altezza di quest'albero, distinguendo diversi casi a seconda di quale sia l'ultima regola applicata. Tutti i casi riguardanti i connettivi della logica proposizionale sono trattati in maniera completamente analoga a quanto fatto nella dimostrazione del teorema di validità per la logica proposizionale. Si tratta quindi di considerare solo i casi in cui l'ultima regola applicata sia una delle regole per l'uguaglianza oppure una delle regole per i quantificatori. Il caso delle regole per l'uguaglianza è omesso, in quanto la verifica è immediata.

Caso dell'introduzione del \forall . Se l'ultima regola applicata è l'introduzione di un quantificatore universale, la derivazione ha la forma:

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \varphi(x) \end{array}}{\forall x \varphi(x)}$$

Ipotesi induttiva. Se Γ contiene tutte le ipotesi del sottoalbero con conclusione $\varphi(x)$, allora $\Gamma \models \varphi(x)$.

Tesi. Se Γ contiene tutte le ipotesi dell'albero con conclusione $\forall x\varphi(x)$, allora $\Gamma \models \forall x\varphi(x)$.

Dimostrazione della tesi. Sia Γ come nella tesi. Salvo cambiare variabili, possiamo supporre che x non compaia tra le variabili libere di Γ . Dall'ipotesi induttiva (che possiamo applicare visto che l'albero e il sottoalbero hanno lo stesso insieme di ipotesi non scaricate), segue che $\Gamma \models \varphi(x)$. Dobbiamo dimostrare che $\Gamma \models \forall x\varphi(x)$. Sia quindi (D, s) una coppia che soddisfa tutte le formule in Γ . Dobbiamo dimostrare che

$$\llbracket \forall x\varphi(x) \rrbracket_s = 1.$$

Questo vale se e solo se per ogni $d \in D$ si ha che

$$\llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1$$

Visto che x non compare in Γ , si ha che la coppia $(D, s[x \mapsto d])$ soddisfa tutte le formule in Γ . Dal fatto che $\Gamma \models \varphi(x)$, segue che

$$\llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1$$

come volevasi dimostrare.

Caso dell'eliminazione del \forall . Se l'ultima regola applicata è un'eliminazione del quantificatore universale, la derivazione ha la forma:

$$\frac{\begin{array}{c} \vdots \\ \forall x\varphi(x) \end{array}}{\varphi(t)}$$

Ipotesi induttiva. Se Γ contiene tutte le ipotesi non scaricate del sottoalbero con conclusione $\forall x\varphi(x)$, allora $\Gamma \models \forall x\varphi(x)$.

Tesi. Se Γ contiene tutte le ipotesi dell'albero con conclusione $\varphi(t)$ allora $\Gamma \models \varphi(t)$.

Dimostrazione della tesi. Sia Γ come nella tesi. Applicando l'ipotesi induttiva, si ha che $\Gamma \models \forall x\varphi(x)$. Dobbiamo dimostrare che $\Gamma \models \varphi(t)$. Sia quindi (D, s) una coppia che soddisfa tutte le formule in Γ . Dobbiamo dimostrare che

$$\llbracket \varphi(t) \rrbracket_s = 1$$

Da $\Gamma \models \forall x\varphi(x)$ segue che per ogni $d \in D$ si ha

$$\llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1$$

In particolare, considerando $d = \llbracket t \rrbracket_s$, si ha che

$$\llbracket \varphi(x) \rrbracket_{s[x \mapsto \llbracket t \rrbracket_s]} = 1$$

Applicando il Lemma di Sostituzione, abbiamo che

$$\llbracket \varphi(t) \rrbracket_s = \llbracket \varphi(x) \rrbracket_{s[x \mapsto \llbracket t \rrbracket_s]} = 1$$

come volevasi dimostrare.

Caso dell'introduzione del \exists . Se l'ultima regola applicata è l'introduzione del quantificatore esistenziale, la derivazione ha la forma:

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \varphi(t) \end{array}}{\exists x \varphi(x)}$$

Ipotesi induttiva. Se Γ contiene tutte le ipotesi del sottoalbero con conclusione $\varphi(t)$, allora $\Gamma \models \varphi(t)$.

Tesi. Se Γ contiene tutte le ipotesi del sottoalbero con conclusione $\exists x \varphi(x)$, allora $\Gamma \models \exists x \varphi(x)$.

Dimostrazione della tesi. Sia Γ come nella tesi. Applicando l'ipotesi induttiva, si ha che $\Gamma \models \varphi(t)$. Dobbiamo dimostrare che $\Gamma \models \exists x \varphi(x)$. Sia (D, s) che soddisfa tutte le formule in Γ . Quindi vale che

$$\llbracket \varphi(t) \rrbracket_s = 1.$$

Dobbiamo dimostrare che

$$\llbracket \exists x \varphi(x) \rrbracket_s = 1$$

Questo vale se e solo se esiste $d \in D$ tale che

$$\llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1$$

Consideriamo $d = \llbracket t \rrbracket_s$. Applicando il Lemma di Sostituzione, abbiamo che

$$\llbracket \varphi(x) \rrbracket_{s[x \mapsto \llbracket t \rrbracket_s]} = \llbracket \varphi(t) \rrbracket_s = 1$$

come volevasi dimostrare.

Caso dell'eliminazione del \exists . Se l'ultima regola applicata è l'eliminazione del quantificatore esistenziale, la derivazione ha la forma:

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \exists x \varphi(x) \end{array} \quad \begin{array}{c} [\varphi(x)]_* \\ \vdots \\ \chi \end{array}}{\chi} \exists_{E,*}$$

Ipotesi induttive. Abbiamo due ipotesi induttive, corrispondenti ai due sottoalberi nelle premesse della regola.

- Se Γ contiene tutte le ipotesi del sottoalbero con conclusione $\exists x\varphi(x)$, allora $\Gamma \models \exists x\varphi(x)$.
- Se Γ' contiene tutte le ipotesi del sottoalbero

$$\begin{array}{c} \varphi(x) \\ \vdots \\ \chi \end{array}$$

allora $\Gamma' \models \chi$.

Tesi. Se Γ contiene tutte le ipotesi del sottoalbero con conclusione χ , allora $\Gamma \models \chi$.

Dimostrazione della tesi. Sia Γ come nella tesi. Salvo cambiare variabili, possiamo supporre che x non compaia in Γ . Dobbiamo dimostrare che $\Gamma \models \chi$. Sia (D, s) che soddisfa tutte le formule in Γ . Per la prima ipotesi induttiva, vale che $\Gamma \models \exists x\varphi(x)$. Quindi, si ha che

$$\llbracket \exists x\varphi(x) \rrbracket_s = 1$$

Questo vuol dire che esiste $d \in D$ tale che

$$\llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1$$

Sia $\Gamma' =_{\text{def}} \Gamma \cup \{\varphi(x)\}$. Chiaramente, vale che $(D, s[x \mapsto d])$ soddisfa tutte le formule in Γ' , visto che x non compare in Γ e $\llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1$, come abbiamo appena visto. Da questo e dal fatto che $\Gamma' \models \chi$, che vale per la seconda ipotesi induttiva, segue che

$$\llbracket \chi \rrbracket_{s[x \mapsto d]} = 1$$

Ma x non compare tra le variabili libere di χ e quindi

$$\llbracket \chi \rrbracket_s = \llbracket \chi \rrbracket_{s[x \mapsto d]} = 1$$

come volevasi dimostrare.

La dimostrazione del teorema di validit a   completa. □

5 Il teorema di completezza e alcune sue applicazioni

Teorema 5.1 (Teorema di Completezza). Sia Γ un insieme di formule, φ una formula. Se $\Gamma \models \varphi$ allora $\Gamma \vdash \varphi$.

La dimostrazione del teorema di completezza per la logica del primo ordine non   in programma, ma ne discutiamo qualche aspetto. Il passaggio fondamentale consiste nel dimostrare il Lemma di Esistenza dei Modelli, che   l'analogo per la logica del primo ordine del Lemma di Esistenza di Valutazioni per la logica proposizionale.

Lemma 5.2 (Lemma di Esistenza dei Modelli). Sia Γ un insieme di formule. Se Γ è consistente, allora Γ ha un modello, ovvero esiste una coppia (D, s) che soddisfa tutte le formule in Γ .

L'idea alla base della dimostrazione del Lemma di Esistenza dei Modelli è di costruire il modello richiesto a partire dal linguaggio L . A tal fine, si potrebbe fissare

$$D =_{\text{def}} \{t \mid t \text{ termine di } L\}$$

e definire $s : \text{Var}_L \rightarrow D$ ponendo $s(x) =_{\text{def}} x$. Ci sono due problemi da risolvere. Come primo problema, dobbiamo decidere come assegnare valori di verità alle formule di L , in modo da rendere valida ogni formula di Γ . Come secondo problema, dobbiamo essere sicuri che se $\Gamma \vdash \exists x\varphi(x)$ allora la struttura D contiene un elemento $d \in D$, ovvero un termine del linguaggio, tale che $\llbracket \varphi(x) \rrbracket_{s[x \mapsto d]} = 1$. Per risolvere il primo problema, è sufficiente considerare, come nel caso della logica proposizionale, insiemi massimalmente consistenti che contengano Γ . Per risolvere il secondo problema, invece, è necessario estendere il linguaggio con nuove costanti, in modo da garantire l'esistenza di termini che siano dei testimoni per le asserzioni esistenziali derivabili da Γ . Questo secondo problema viene risolto tramite il metodo delle cosiddette *teorie di Henkin*, che non tratteremo. Discutiamo invece qualche applicazione del Teorema di Completezza.

Teorema 5.3 (Teorema di Compattatezza). Un insieme di formule Γ ha un modello se e solo se ogni sottoinsieme finito $\Gamma' \subseteq \Gamma$ ha un modello.

Dimostrazione. Sia Γ come nell'enunciato. Dimosteremo che Γ non ha nessun modello se e solo se esiste un sottoinsieme finito $\Gamma' \subseteq \Gamma$ che non ha nessun modello. Supponiamo che Γ non abbia nessun modello. Allora, Γ non può essere consistente. Infatti, se lo fosse, avrebbe un modello per il Lemma di Esistenza di Modelli. Quindi $\Gamma \vdash \perp$. Ma allora esiste un insieme finito $\Gamma' \subseteq \Gamma$ tale che $\Gamma' \vdash \perp$. Questo perchè ogni derivazione ha un numero finito di premesse non scaricate. Dal fatto che $\Gamma' \vdash \perp$ segue che Γ' non ha nessun modello, come volevasi dimostrare. Supponiamo adesso che esista un sottoinsieme $\Gamma' \subseteq \Gamma$ che non ha nessun modello. Allora è impossibile che esista un modello di Γ . Infatti, se esistesse, questo sarebbe anche un modello di Γ' , essendo $\Gamma' \subseteq \Gamma$. \square

Teorema 5.4 (Teorema di Löwenheim-Skolem Downward). Sia Γ un insieme di formule. Se Γ ha un modello, allora Γ ha un modello di cardinalità numerabile.

Traccia della dimostrazione. Il modello di Γ è costruito a partire dall'insieme dei termini del linguaggio L , che è numerabile, tramite operazioni che preservano la numerabilità dell'insieme. Per esempio, se aggiungiamo al linguaggio costanti c_φ per ogni formula chiusa della forma $\exists x\varphi(x)$, il linguaggio rimane numerabile, in quanto le formule della forma $\exists x\varphi(x)$ sono anch'esse numerabili. \square

Teorema 5.5 (Teorema di Löwenheim-Skolem Upward). Sia Γ un insieme di formule. Se Γ ha un modello infinito, allora Γ ha modelli di cardinalità arbitrariamente grande.

Dimostrazione. Sia I un insieme infinito. Vogliamo costruire un modello di Γ di cardinalità almeno $|I|$. A questo fine, definiamo L' come l'estensione del linguaggio L con una nuova costante c_i per ogni $i \in I$. Sia adesso

$$\Gamma' = \Gamma \cup \{c_i \neq c_j \mid i, j \in I, i \neq j\}$$

Adesso dimostriamo che Γ' ha un modello di cardinalità almeno $|I|$. Si osservi che ogni sottoinsieme finito di Γ' ha un modello, visto che Γ ha un modello infinito, e quindi è sempre possibile trovare elementi distinti del modello per una quantità finita delle costanti c_i , in modo quindi da rendere valide le formule $c_i \neq c_j$. Un'applicazione del teorema di compattezza ci permette di concludere che Γ' ha un modello. Questo modello deve avere almeno cardinalità $|I|$, visto che deve contenere elementi distinti $\llbracket c_i \rrbracket$ per ogni $i \in I$. \square

Esempio 5.6. Gli assiomi della teoria dell'aritmetica del primo ordine hanno un modello infinito, dato dall'insieme \mathbb{N} con la ovvia struttura. Per il teorema di Löwenheim-Skolem Upward, si sa che esistono modelli dell'aritmetica arbitrariamente grandi, i cosiddetti *modelli non-standard*.

6 Definibilità

Definizione 6.1. Sia D una struttura per L . Sia $n \in \mathbb{N}$. Un sottoinsieme $P \subseteq D^n$ è detto essere *definibile* in L se esiste una formula $\varphi(x_1 \dots x_n)$ di L tale che per ogni $a_1, \dots, a_n \in D$

$$(a_1, \dots, a_n) \in P \quad \text{se e solo se} \quad \llbracket \varphi(x_1 \dots x_n) \rrbracket_{x_1 \mapsto a_1, \dots, x_n \mapsto a_n} = 1.$$

Definizione 6.2. Sia T una teoria in L .

- (i) Una formula $\varphi(x_1, \dots, x_n, y)$ è detta essere *dimostrabilmente funzionale* in T se vale che

$$\begin{aligned} T \vdash \forall x_1 \dots \forall x_n \exists y \varphi(x_1, \dots, x_n, y) \\ T \vdash \forall x_1 \dots \forall x_n \forall y \forall y' (\varphi(x_1, \dots, x_n, y) \wedge \varphi(x_1, \dots, x_n, y') \Rightarrow y = y') \end{aligned}$$

- (ii) Sia D un modello di T , ovvero una struttura per L in cui tutti gli assiomi di T sono validi. Diremo che una funzione $f : D^n \rightarrow D$ è detta essere *definibile* in T se esiste una formula dimostrabilmente funzionale $\varphi(x_1, \dots, x_n, y)$ tale che

$$f(a_1, \dots, a_n) = b \quad \text{sse} \quad \llbracket \varphi(x_1, \dots, x_n, y) \rrbracket_{x_1 \mapsto a_1, \dots, x_n \mapsto a_n, y \mapsto b} = 1.$$

Si noti che, affinché una funzione sia definibile è necessario che il suo grafico,

$$\{(a_1, \dots, a_n, b) \in D^{n+1} \mid f(a_1, \dots, a_n) = b\}$$

sia un sottoinsieme definibile di D^{n+1} .

Proposizione 6.3.

- (i) La funzione identità $1_D : D \rightarrow D$ è definibile.
- (ii) Se $f : D \rightarrow D$ e $g : D \rightarrow D$ sono definibili, allora $g \circ f : D \rightarrow D$ è definibile.

Dimostrazione.

- (i) La funzione identità è definita dalla formula $y = x$.
- (ii) Supponiamo che $f : D \rightarrow D$ sia definita dalla formula $\varphi(x, y)$ e che $g : D \rightarrow D$ sia definita dalla formula $\psi(y, z)$. Da questo segue che la funzione composta $g \circ f : D \rightarrow D$ è definita dalla formula $\exists y(\varphi(x, y) \wedge \psi(y, z))$.

In entrambi i casi, la verifica dei dettagli è lasciata come esercizio. \square

7 Esercizi

Esercizio 7.1. Si dimostri che le seguenti formule sono teoremi della logica del primo ordine.

- (i) $\neg P(t) \Rightarrow \neg \forall x P(x)$
- (ii) $\forall x \exists y (\varphi(x) \Rightarrow \psi(x, y)) \Rightarrow \forall x (\varphi(x) \Rightarrow \exists y \psi(x, y))$
- (iii) $\neg \exists x (\varphi(x) \wedge \psi(x)) \Rightarrow \forall x (\neg \varphi(x) \vee \neg \psi(x))$
- (iv) $\forall x (\varphi(x) \Rightarrow \exists y \psi(x, y)) \Rightarrow \forall x (\neg \exists y \psi(x, y) \Rightarrow \neg \varphi(x))$
- (v) $\forall x P(x) \vee \forall x Q(x) \Rightarrow \forall x (P(x) \vee Q(x))$
- (vi) $(\exists x P(x) \Rightarrow \forall x Q(x)) \Rightarrow \forall x (P(x) \Rightarrow Q(x))$

Esercizio 7.2. Si dimostri che le seguenti formule non sono teoremi della logica del primo ordine.

- (i) $\exists x P(x) \Rightarrow \forall x P(x)$
- (ii) $\exists x P(x) \wedge \exists x Q(x) \Rightarrow \exists x (P(x) \wedge Q(x))$
- (iii) $\exists x (P(x, x) \Rightarrow \forall y P(x, y))$

Esercizio 7.3.

- (i) Si verifichi che la formula

$$\forall x_1 \forall x_2 \forall x_3 (x_1 < x_2 < x_3 \Rightarrow \exists y_1 \exists y_2 (x_1 < y_1 < x_2 < y_2 < x_3))$$

non è valida in $(\mathbb{N}, <)$.

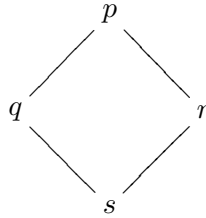
- (ii) Si dimostri che la formula

$$\forall x \exists y_1 \exists y_2 (y_1 < x < y_2)$$

è valida in $(\mathbb{Z}, <)$.

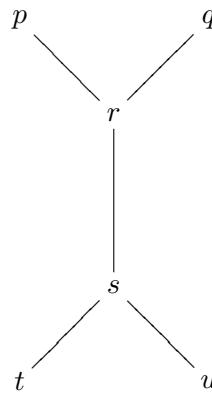
Esercizio 7.4.

(i) Sia $P = \{p, q, r, s\}$ e si consideri l'ordine parziale descritto dal diagramma



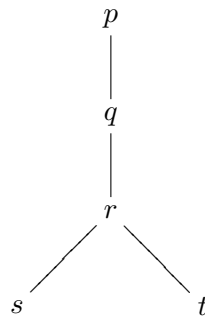
Si verifichi che $\exists x \forall y (x \leq y)$ è valida in (P, \leq) .

(ii) Sia $P = \{p, q, r, s, t, u\}$ e si consideri l'ordine parziale (P, \leq) descritto dal diagramma

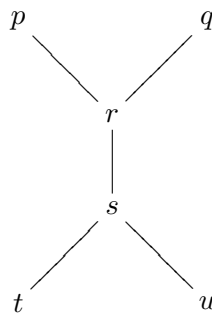


Si verifichi che $\forall x_1 \forall x_2 \exists y (x_1 \leq y \wedge x_2 \leq y)$ non è valida in (P, \leq) .

Esercizio 7.5. Si definisca una formula del linguaggio degli ordini parziali che sia valida nell'ordine parziale descritto dal diagramma



ma non sia valida nell'ordine parziale descritto dal diagramma



Esercizio 7.6. Si considerino gli ordini parziali (\mathbb{N}, \leq) , (\mathbb{Z}, \leq) , (\mathbb{Q}, \leq) .

- (i) Si definisca una formula valida in (\mathbb{N}, \leq) ma non in (\mathbb{Z}, \leq)
- (ii) Si definisca una formula ψ valida in (\mathbb{Z}, \leq) ma non in (\mathbb{Q}, \leq) .

Esercizio 7.7. Si dimostri che i seguenti insiemi sono definibili in PA.

- (i) $\{n \in \mathbb{N} \mid n \text{ è divisibile per } 6\}$.
- (ii) $\{n \in \mathbb{N} \mid n \text{ è dispari}\}$.
- (iii) $\{n \in \mathbb{N} \mid n \text{ è primo}\}$.

Capitolo 3

Teoria della calcolabilità

1 Funzioni ricorsive

L'insieme delle funzioni ricorsive è il più piccolo insieme X di funzioni da potenze di \mathbb{N} (ovvero insiemi della forma \mathbb{N}^n , per qualche $n \in \mathbb{N}$) a \mathbb{N} che soddisfa le proprietà seguenti.

- (i) Per ogni $1 \leq i \leq n$, la proiezione i -esima

$$\begin{aligned} \pi_i : \quad \mathbb{N}^n &\rightarrow \mathbb{N} \\ (x_1, \dots, x_n) &\mapsto x_i \end{aligned}$$

è in X .

- (ii) La funzione costante 0 e la funzione successore

$$\begin{array}{ll} c_0 : \mathbb{N} \rightarrow \mathbb{N} & S : \mathbb{N} \rightarrow \mathbb{N} \\ x \mapsto 0 & x \mapsto x + 1 \end{array}$$

sono in X .

- (iii) Se $f : \mathbb{N}^n \rightarrow \mathbb{N}$ è in X e, per ogni $1 \leq i \leq n$, $g_i : \mathbb{N}^m \rightarrow \mathbb{N}$ è in X , allora la funzione composta

$$\begin{aligned} f(g_1, \dots, g_n) : \quad \mathbb{N}^m &\rightarrow \mathbb{N} \\ x_1, \dots, x_m &\mapsto f(g_1(x_1, \dots, x_m), \dots, g_n(x_1, \dots, x_m)) \end{aligned}$$

è in X .

- (iv) Se $f : \mathbb{N}^n \rightarrow \mathbb{N}$ e $g : \mathbb{N}^{n+2} \rightarrow \mathbb{N}$ sono in X , allora la funzione

$$\begin{aligned} h : \quad \mathbb{N}^{n+1} &\rightarrow \mathbb{N} \\ (x_1, \dots, x_n, 0) &\mapsto f(x_1, \dots, x_n) \\ (x_1, \dots, x_n, y + 1) &\mapsto g(x_1, \dots, x_n, y, h(x_1, \dots, x_n, y)) \end{aligned}$$

è in X .

(v) Se $f : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ è in X , allora la funzione

$$\begin{aligned} \text{Min}(f) : \quad \mathbb{N}^n &\rightarrow \mathbb{N} \\ (x_1, \dots, x_n) &\rightarrow \min\{x \in \mathbb{N} \mid f(x_1, \dots, x_n, x) = 0\} \end{aligned}$$

è in X .

Scriveremo Rec per indicare l'insieme delle funzioni ricorsive. Le funzioni in (i) e (ii) sono dette le *funzioni ricorsive di base*. Le funzioni ottenute tramite l'applicazione delle regole in (i)-(iv) sono dette le *funzioni primitive ricorsive*. Si noti che l'insieme delle funzioni ricorsive include non solo funzioni totali, ma anche funzioni parziali $f : \mathbb{N}^n \rightarrow \mathbb{N}$, che non necessariamente associano un valore ad ogni elemento $(x_1, \dots, x_n) \in \mathbb{N}^n$. Per esempio, anche se $f : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ è una funzione totale, $\text{Min}(f) : \mathbb{N}^n \rightarrow \mathbb{N}$ è una funzione parziale qualora ci sia una sequenza $(x_1, \dots, x_n) \in \mathbb{N}^n$ per cui non esiste $x \in \mathbb{N}$ tale che $f(x_1, \dots, x_n, x) = 0$.

Nel caso $n = 0$, la definizione per ricorsione in (iv) assume la seguente forma. Dato un elemento $a \in \mathbb{N}$ e una funzione $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, abbiamo una funzione

$$\begin{aligned} h : \quad \mathbb{N} &\rightarrow \mathbb{N} \\ 0 &\mapsto a \\ y + 1 &\mapsto g(y, h(y)) \end{aligned}$$

Nel caso $n = 1$, la definizione per ricorsione in (iv) assume la seguente forma. Date funzioni $f : \mathbb{N} \rightarrow \mathbb{N}$ e $g : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, abbiamo una funzione

$$\begin{aligned} h : \quad \mathbb{N} \times \mathbb{N} &\rightarrow \mathbb{N} \\ (x, 0) &\mapsto f(x) \\ (x, y + 1) &\mapsto g(x, y, h(x, y)) \end{aligned}$$

Osservazione 1.1. La funzione $f : \mathbb{N} \rightarrow \mathbb{N}$ che calcola i numeri di Fibonacci soddisfa le equazioni

$$\begin{aligned} f(0) &= 1 \\ f(1) &= 1 \\ f(x + 2) &= f(x) + f(x + 1) \end{aligned}$$

Nonostante la definizione di questa funzione non rientri nello schema delle funzioni ricorsive, è possibile tuttavia dimostrare che è ricorsiva. In generale possiamo definire funzioni ricorsive $f : \mathbb{N} \rightarrow \mathbb{N}$ anche quando $f(x + 1)$ dipende da tutti i valori $f(0), \dots, f(x)$, e non solo da $f(x)$. Per dimostrare questo, è necessario codificare i valori $f(0), \dots, f(x)$ in un singolo numero. L'idea che ci permette di fare questo è di associare ad una sequenza

$$x_0, x_1, \dots, x_n$$

di numeri (che, per semplicità, supponiamo diversi da zero), il numero

$$2^{x_0} \cdot 3^{x_1} \cdot \dots \cdot p_n^{x_n}$$

ove, per $0 \leq i \leq n$, p_i denota il numero primo $(i+1)$ -esimo. È chiaro che questa definizione determina una funzione ricorsiva $f : \mathbb{N}^n \rightarrow \mathbb{N}$. È meno evidente, ma comunque vero, che per ogni $0 \leq i \leq n$, possiamo definire una funzione da $\mathbb{N} \times \mathbb{N}$ a \mathbb{N} che assegna ad un numero x il numero $(x)_i$, definito come l'esponente della più alta potenza di p_i che divide x . In con questa definizione, è chiaro che vale

$$(2^{x_0} \cdot 3^{x_1} \cdot \dots \cdot p_n^{x_n})_i = x_i$$

per ogni $0 \leq i \leq n$. Grazie ha questa idea, data una funzione ricorsiva $f : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ è possibile definire una nuova funzione ricorsiva $\bar{f} : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ tale che $\bar{f}(x_1, \dots, x_n, y+1)$ codifica tutti i valori $f(x_0, \dots, x_n, 0), \dots, f(x_0, \dots, x_n, y)$.

2 Funzioni calcolabili

Introdurremo la nozione di funzione calcolabile. A questo fine, introduciamo le macchine a registri, modelli idealizzati di calcolatori, molto simili alle macchine di Turing. Una *macchina a registri* consiste di un nastro infinito

R_1	R_2	R_3	R_4	\dots
-------	-------	-------	-------	---------

ove $R_1, R_2, \dots \in \mathbb{N}$, sono detti i *registri* della macchina. Un programma per una macchina a registri è una sequenza di istruzioni, o *stati*. Abbiamo due stati speciali:

- S_0 , lo stato terminale, in cui l'esecuzione del programma termina.
- S_1 , lo stato iniziale, da cui comincia sempre l'esecuzione del programma.

Per ogni stato S_i abbiamo due istruzioni possibili:

- Aggiungi 1 al registro R_n ed esegui S_j ,
- Verifica se $R_n = 0$. Se sí, esegui S_j ; altrimenti, sottrai 1 al registro R_n ed esegui S_k .

Per convenzione, se un programma termina, considereremo R_1 come il risultato del programma. Utilizzeremo la seguente notazione per abbreviare questi comandi:

- $R_n := R_n + 1$ ed esegui S_j ,
- Se $R_n = 0$ allora esegui S_j altrimenti $R_n := R_n - 1$ ed esegui S_k .

Esempio 2.1. Il seguente programma aggiunge il valore di R_2 a R_1 .

S_0	Termina
S_1	Se $R_2 = 0$ allora esegui S_0 altrimenti $R_2 := R_2 - 1$ ed esegui S_2
S_2	$R_1 := R_1 + 1$ ed esegui S_1 ,

Come esercizio, si verifichi che il programma, cominciando con

5	2	0	0	...
---	---	---	---	-----

termina con

7	0	0	0	...
---	---	---	---	-----

Si verifichi che è possibile scrivere programmi che non terminano, ovvero che non raggiungono mai ad eseguire S_0 .

Definizione 2.2. Una funzione $f : \mathbb{N}^n \rightarrow \mathbb{N}$ è *calcolabile* se esiste un programma P tale che, per ogni $(x_1, \dots, x_n) \in \mathbb{N}^n$, si ha che:

- se $f(x_1, \dots, x_n)$ esiste, allora il programma, cominciando con

x_1	x_2	...	x_n	0	0	...
-------	-------	-----	-------	---	---	-----

termina arrivando a

$f(x_1, \dots, x_n)$...
----------------------	-----

- se $f(x_1, \dots, x_n)$ non esiste, allora P non termina.

3 La tesi di Church

Teorema 3.1. Ogni funzione ricorsiva è calcolabile.

Dimostrazione. Definiamo

$$X = \{f : \mathbb{N}^n \rightarrow \mathbb{N} \mid f \text{ è calcolabile} \}$$

Dimostreremo che X soddisfa le proprietà (i)-(v) con cui abbiamo definito l'insieme Rec delle funzioni ricorsive. Dalla minimalità di Rec, seguirà che $\text{Rec} \subseteq X$, da cui si ottiene l'asserzione desiderata. Dimostreremo ciascuna delle proprietà separatamente.

Per (i), dobbiamo dimostrare che ciascuna delle proiezioni $\pi_i : \mathbb{N}^n \rightarrow \mathbb{N}$, per $1 \leq i \leq n$, è calcolabile. Per $i = 1$, è sufficiente scrivere un programma che non modifica R_1 e che poi termina. Per esempio:

S_0	Termina
S_1	$R_2 := R_2 + 1$ ed esegui S_0

Per $i > 1$, scriviamo un programma che azzerava R_1 , trasferisce il valore di R_i in R_1 e termina:

S_0	Termina
S_1	Se $R_1 = 0$ allora esegui S_2 altrimenti $R_1 := R_1 - 1$ ed esegui S_1
S_2	Se $R_i = 0$ allora esegui S_0 altrimenti $R_i := R_i - 1$ ed esegui S_3
S_3	$R_1 := R_1 + 1$ ed esegui S_2

Per (ii), la funzione costante $c_0 : \mathbb{N} \rightarrow \mathbb{N}$ è calcolata dal programma:

S_0 Termina
 S_1 Se $R_1 = 0$ allora esegui S_0 altrimenti $R_1 := R_1 - 1$ ed esegui S_1

La funzione successore $S : \mathbb{N} \rightarrow \mathbb{N}$ è invece calcolata dal programma:

S_0 Termina
 S_1 $R_1 := R_1 + 1$ ed esegui S_0

Per (iii), supponiamo di avere una funzione calcolabile $f : \mathbb{N}^n \rightarrow \mathbb{N}$ e, per ogni $1 \leq i \leq n$, una funzione calcolabile $g_i : \mathbb{N}^m \rightarrow \mathbb{N}$. Dobbiamo dimostrare che la funzione $f(g_1, \dots, g_n) : \mathbb{N}^m \rightarrow \mathbb{N}$ è calcolabile. Per capire come scrivere il programma, si ricordi che il valore di questa funzione su (x_1, \dots, x_m) è dato da:

$$f(g_1(x_1, \dots, x_m), \dots, g_n(x_1, \dots, x_m))$$

Calcoleremo separatamente i valori di $g_1(x_1, \dots, x_m), \dots, g_n(x_1, \dots, x_m)$. A questo fine, è necessario salvare i valori di x_1, \dots, x_m in modo da poterli utilizzare più di una volta. Possiamo poi applicare il programma che calcola f sui valori di $g_1(x_1, \dots, x_m), \dots, g_n(x_1, \dots, x_m)$ per ottenere il risultato desiderato. Il programma desiderato è quindi il seguente.

Primo passo. Copiamo i valori di R_1, \dots, R_m in R_{k+1}, \dots, R_{k+m} ,

R_1	R_2	\dots	R_m	\dots	\dots	\dots	\dots	\dots	\dots	R_{k+1}	\dots	R_{k+m}
-------	-------	---------	-------	---------	---------	---------	---------	---------	---------	-----------	---------	-----------

ove $k > m$ è sufficientemente grande da fare in modo che questi valori non vengano modificati dai calcoli che seguono. Questo ci permette di utilizzare ripetutamente i valori iniziali di R_1, \dots, R_m , come faremo nel secondo passo.

Secondo passo. Per ogni $1 \leq i \leq n$, copiamo i valori di R_{k+1}, \dots, R_{k+m} in R_{n+1}, \dots, R_{n+m} :

\dots	\dots	\dots	\dots	R_{n+1}	\dots	\dots	R_{n+m}	\dots	R_{k+1}	\dots	R_{k+m}
---------	---------	---------	---------	-----------	---------	---------	-----------	---------	-----------	---------	-----------

A questo punto, applichiamo il programma che calcola $g_i : \mathbb{N}^m \rightarrow \mathbb{N}$, traslato di n , ai valori in R_{n+1}, \dots, R_{n+m} . Il valore ottenuto è contenuto in R_{n+1} . Copiamo questo valore in R_i .

Terzo passo. A questo punto, per $1 \leq i \leq n$, il registro R_i contiene il valore ottenuto dall'applicazione del programma che calcola g_i sui valori iniziali di R_1, \dots, R_m . Possiamo quindi applicare il programma che calcola f su

R_1	R_2	\dots	R_n	\dots
-------	-------	---------	-------	---------

e terminare il programma.

Per (iv), supponiamo di avere due funzioni computabili $f : \mathbb{N}^n \rightarrow \mathbb{N}$ e $g : \mathbb{N}^{n+2} \rightarrow \mathbb{N}$. Dobbiamo dimostrare che la funzione $h : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$, definita da

$$\begin{aligned} h(x_1, \dots, x_n, 0) &= f(x_1, \dots, x_n), \\ h(x_1, \dots, x_n, y + 1) &= g(x_1, \dots, x_n, y, h(x_1, \dots, x_n, y)). \end{aligned}$$

è calcolabile. Per capire come funziona il programma che calcola h , consideriamo qualche esempio:

$$\begin{aligned} h(x_1, \dots, x_n, 0) &= f(x_1, \dots, x_n) \\ h(x_1, \dots, x_n, 1) &= g(x_1, \dots, x_n, 0, h(x_1, \dots, x_n, 0)) \\ h(x_1, \dots, x_n, 2) &= g(x_1, \dots, x_n, 1, h(x_1, \dots, x_n, 1)) \\ h(x_1, \dots, x_n, 3) &= g(x_1, \dots, x_n, 2, h(x_1, \dots, x_n, 2)) \\ &\vdots \qquad \qquad \qquad \vdots \end{aligned}$$

Consideriamo i seguenti registri:

R_1	\dots	R_n	R_{n+1}	R_{n+2}	\dots	\dots	R_{k+1}	\dots	R_{k+n}	R_{k+n+1}	R_{k+n+2}
-------	---------	-------	-----------	-----------	---------	---------	-----------	---------	-----------	-------------	-------------

I registri saranno utilizzati come segue.

- I registri R_1, \dots, R_n vengono utilizzati per calcolare f o g e poi ristabiliti al loro valore iniziale.
- Dopo averne salvato il valore iniziale, il registro R_{n+1} viene fissato a 0 e poi incrementato progressivamente, assumendo i valori $0, 1, 2, \dots$, analogamente all'incremento dell'argomento $(n+1)$ -esimo di g nel lato destro delle equazioni precedenti.
- Il registro R_{n+2} contiene i valori temporanei di h .
- I registri R_{k+1}, \dots, R_{k+n} conservano i valori iniziali di R_1, \dots, R_n .
- Il registro R_{k+n+1} viene inizialmente fissato con il valore di R_{n+1} e poi progressivamente decrementato, in modo da segnalare quando dobbiamo terminare il programma.
- Il registro R_{k+n+2} viene inizialmente fissato a 0 e poi progressivamente incrementato, permettendoci di incrementare R_{n+1} .

Inizio. Assumiamo che il registro abbia la forma

R_1	R_2	\dots	R_n	R_{n+1}	0	0	\dots
-------	-------	---------	-------	-----------	---	---	---------

Primo passo. Copiamo R_1, \dots, R_n, R_{n+1} in $R_{k+1}, \dots, R_{k+n}, R_{k+n+1}$

R_1	R_2	\dots	R_n	R_{n+1}	\dots	R_{k+1}	\dots	R_{k+n}	R_{k+n+1}	\dots
-------	-------	---------	-------	-----------	---------	-----------	---------	-----------	-------------	---------

ove $k > n + 1$ è sufficientemente grande.

Secondo passo. Applichiamo il programma che calcola f ai valori in R_1, \dots, R_n .

Terzo passo. Se $R_{k+n+1} = 0$ allora esegui S_0 altrimenti $R_{k+n+1} := R_{k+n+1} - 1$ ed esegui il seguente sottoprogramma.

- Copia R_1 in R_{n+2} .
- Copia R_{k+1}, \dots, R_{k+n} in R_1, \dots, R_n .

- Copia R_{k+n+2} in R_{n+1}
- Esegui il programma che calcola g su $R_1, \dots, R_n, R_{n+1}, R_{n+2}$.
- Incrementiamo di 1 il valore di R_{k+n+2} .
- Ritorna ad eseguire i comandi del terzo passo.

Per (v), supponiamo che $f : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ sia calcolabile. Dobbiamo dimostrare che $\text{Min}(f) : \mathbb{N}^n \rightarrow \mathbb{N}$ è calcolabile. Il programma che calcola $\text{Min}(f)$ consiste, essenzialmente, nel calcolare

$$\begin{array}{c} f(x_1, \dots, x_n, 0) \\ f(x_1, \dots, x_n, 1) \\ \vdots \\ f(x_1, \dots, x_n, x) \\ \vdots \end{array}$$

Verificando, ad ogni passaggio, se abbiamo ottenuto 0. Una volta ottenuto 0, diamo il valore di x . Ancora una volta, per fare questo, abbiamo bisogno di salvare i valori iniziali di x_1, \dots, x_n in una parte remota del nastro. Il programma è esplicitamente dato come segue.

Primo passo. Copiamo i valori di R_1, \dots, R_n in R_{k+1}, \dots, R_{k+n} , ove k è sufficientemente grande.

Secondo passo. Eseguiamo il seguente sottoprogramma.

- Copiamo $R_{k+1}, \dots, R_{k+n+1}$ in R_1, \dots, R_{n+1} ;
- Appliciamo il programma che calcola f su R_1, \dots, R_{n+1} .

Terzo passo. Se $R_1 = 0$ allora esegui S_i altrimenti $R_1 := R_1 - 1$ ed esegui S_j , ove S_i copia il contenuto di R_{k+n+1} in R_1 e termina il programma, mentre S_j incrementa R_{k+n+1} di 1 e ritorna all'inizio del secondo passo.

La dimostrazione è completata. □

Teorema 3.2. Ogni funzione calcolabile è ricorsiva.

Traccia della dimostrazione. Sia $f : \mathbb{N}^n \rightarrow \mathbb{N}$ una funzione calcolabile. Sia P un programma che calcola f . Definiamo una funzione ausiliaria $g : \mathbb{N}^{n+2} \rightarrow \mathbb{N}$ come segue:

- $g(x_1, \dots, x_n, 0, t)$ è definito come il numero dello stato raggiunto dal P dopo aver eseguito t passi, assumendo di iniziare con

x_1	\dots	x_n	0	0	\dots
-------	---------	-------	---	---	---------

Conveniamo che se P termina in meno di t passi, siamo nello stato S_0 .

- Per $i > 0$, $g(x_1, \dots, x_n, i, t)$ è definito come il valore del registro R_i dopo che P ha eseguito t passi.

Dimostreremo che

$$f(x_1, \dots, x_n) = g(x_1, \dots, x_n, 1, h(x_1, \dots, x_n)) \quad (*)$$

ove $h(x_1, \dots, x_n)$ è definito come il minimo t tale che $g(x_1, \dots, x_n, 0, t) = 0$. Per dimostrare (*), è sufficiente osservare che $g(x_1, \dots, x_n, 1, h(x_1, \dots, x_n))$ è il valore di R_1 dopo che P ha eseguito $h(x_1, \dots, x_n)$ passi. Per definizione di h , quindi, $g(x_1, \dots, x_n, 1, h(x_1, \dots, x_n))$ è il valore di R_1 dopo che P ha eseguito il minimo numero di passi t tale che $g(x_1, \dots, x_n, 0, t) = 0$. Per definizione di g , quindi, questo valore è il contenuto del registro R_1 dopo che P ha eseguito il minimo numero di passi che permettono di raggiungere S_0 , ovvero di terminare. Visto che P calcola f , il valore in R_1 quando P termina è esattamente $f(x_1, \dots, x_n)$. Rimane da dimostrare che g , e quindi h sono ricorsive. Da questo, tramite l'equazione in (*), seguirà che f è ricorsiva, come vogliamo dimostrare. Quest'ultima verifica non è parte del programma. \square

4 Esempio di una funzione non ricorsiva

Vogliamo dare un esempio di una funzione $f : \mathbb{N}^n \rightarrow \mathbb{N}$ che non sia ricorsiva. È chiaro che una tale funzione deve esistere, visto che l'insieme delle funzioni ricorsive $f : \mathbb{N}^n \rightarrow \mathbb{N}$ è numerabile, mentre l'insieme di tutte le funzioni $f : \mathbb{N}^n \rightarrow \mathbb{N}$ non lo è. Per definire una funzione non ricorsiva, procederemo in tre passi. La dimostrazione che la funzione definita non è ricorsiva utilizzerà il metodo di diagonalizzazione di Cantor.

Primo passo. Osserviamo che un'istruzione del tipo

$$R_n := R_n + 1 \text{ ed esegui } S_j$$

è completamente determinata dai numeri n e j . Per convenzione, le assegneremo il codice $2^n \cdot 5^j$. Invece, un'istruzione del tipo:

$$\text{Se } R_n = 0 \text{ allora esegui } S_j \text{ altrimenti } R_n := R_n - 1 \text{ ed esegui } S_k$$

è completamente determinata dai numeri n , j , e k . Quindi, per convenzione, le assegnamo il codice $2^n \cdot 3 \cdot 5^j \cdot 7^k$. Il fattore 3 in questo codice serve per aiutarci a distinguere i numeri che codificano istruzioni del primo tipo rispetto ad istruzioni del secondo tipo.

Secondo passo. Si noti che un programma per macchine a registri non è nient'altro che una sequenza di istruzioni. Quindi possiamo assegnare a ciascun programma un codice nel modo seguente. Dato un programma che consiste di istruzioni S_0, \dots, S_n , innanzitutto assegnamo un codice a ciascuna istruzione, secondo quanto spiegato nel primo passo, e otteniamo quindi una sequenza di numeri i_0, \dots, i_n . Dopodichè, associamo a questa sequenza di numeri il singolo numero

$$2^{i_0} \cdot 3^{i_1} \cdot \dots \cdot p_n^{i_n}$$

Tra i numeri naturali, esisteranno quindi numeri n ottenuti tramite questa procedura. Possiamo pensare a questi numeri come codici di programmi per macchine a registri. Dato $n \in \mathbb{N}$, scriveremo P_n per il programma codificato da n , se questo esiste.

Terzo passo. dati $n \in \mathbb{N}$ e $k \in \mathbb{N}$ definiamo $f_{n,k}$ come la funzione ricorsiva $f_{n,k} : \mathbb{N}^k \rightarrow \mathbb{N}$ calcolata dal programma P_n , se questo esiste.

Teorema 4.1. La funzione $g : \mathbb{N} \rightarrow \mathbb{N}$ definita da

$$g(n) = \begin{cases} f_{n,1}(n) + 1, & \text{se } f_{n,1} \text{ esiste e } f_{n,1}(n) \text{ è definito.} \\ 0, & \text{altrimenti.} \end{cases}$$

non è ricorsiva.

Dimostrazione. Innanzitutto, non esiste $n \in \mathbb{N}$ tale che $g = f_{n,1}$. Infatti se esistesse, avremmo sia che $g(n) = f_{n,1}(n)$ (per ipotesi) che $g(n) = f_{n,1}(n) + 1$ (per definizione di g). Da questo segue che g non è ricorsiva. Infatti, se lo fosse, esisterebbe un programma P_n per qualche $n \in \mathbb{N}$, che calcola g . Ma allora avremmo $g = f_{n,1}$, in contraddizione con quanto appena stabilito. \square

5 Insiemi ricorsivi e ricorsivamente enumerabili

Definizione 5.1. Un sottoinsieme $S \subseteq \mathbb{N}$ è detto essere *ricorsivo* se la sua funzione caratteristica $\chi_S : \mathbb{N} \rightarrow \mathbb{N}$, definita da

$$\chi_S(x) = \begin{cases} 1 & \text{se } x \in S, \\ 0 & \text{altrimenti,} \end{cases}$$

è una funzione ricorsiva.

Definizione 5.2. Un sottoinsieme $S \subseteq \mathbb{N}$ è detto essere *ricorsivamente enumerabile* se la funzione $\varphi_S : \mathbb{N} \rightarrow \mathbb{N}$ definita da

$$\varphi_S(x) = \begin{cases} 1 & \text{se } x \in S \\ \text{non definito} & \text{altrimenti} \end{cases}$$

è ricorsiva.

Lemma 5.3. Sia $S \subseteq \mathbb{N}$. Le seguenti condizioni sono equivalenti.

- (i) S è ricorsivamente enumerabile.
- (ii) Esiste una funzione ricorsiva $f : \mathbb{N}^n \rightarrow \mathbb{N}$ tale che

$$S = \{f(x_1, \dots, x_n) \in \mathbb{N} \mid (x_1, \dots, x_n) \in \mathbb{N}^n\}.$$

- (iii) Esiste una funzione ricorsiva $f : \mathbb{N} \rightarrow \mathbb{N}$ tale che

$$S = \{x \in \mathbb{N} \mid f(x) \text{ è definito}\}.$$

(iv) La funzione $\psi_S : \mathbb{N} \rightarrow \mathbb{N}$ definita da

$$\varphi_S(x) = \begin{cases} x & \text{se } x \in S \\ \text{non definito} & \text{altrimenti} \end{cases}$$

è ricorsiva.

Dimostrazione. Dimostreremo (iii) \Rightarrow (i) \Rightarrow (iv) \Rightarrow (ii) \Rightarrow (iii).

(iii) \Rightarrow (i). Sia f come in (iii). Dato un programma P che calcola f , possiamo ottenere un programma che calcola φ_S aggiungendo a P un'istruzione che, dopo aver calcolato $f(x)$, assegna il valore 1 al registro R_1 .

(i) \Rightarrow (iv). Dato un programma che calcola φ_S è immediato scrivere un programma che calcola ψ_S . Per esempio, basta copiare x in una parte remota della macchina, eseguire il programma per φ_S , e poi copiare x su R_1 .

(iv) \Rightarrow (ii). È sufficiente considerare f data da ψ_S .

(ii) \Rightarrow (iii). Sia $f : \mathbb{N}^n \rightarrow \mathbb{N}$ come in (ii). Dato un programma che calcola f , descriviamo un programma Q che, dato un input $x \in S$, termina se e solo se $x \in S$. Dato un input x , il programma Q calcola i valori di $f(x_1, \dots, x_n)$ e confronta il risultato ottenuto con x . Se il risultato ottenuto è uguale ad x , allora il programma termina. Altrimenti, Q procede a calcolare un altro valore di $f(x_1, \dots, x_n)$.

□

Definizione 5.4. Un sottoinsieme $S \subseteq \mathbb{N}$ che soddisfa le condizioni equivalenti del Lemma 5.3 è detto essere *ricorsivamente enumerabile*.

Proposizione 5.5. Un sottoinsieme $S \subseteq \mathbb{N}$ è ricorsivo se e solo se sia S che $\mathbb{N} \setminus S$ sono ricorsivamente enumerabili.

Dimostrazione. Sia S un insieme ricorsivo. Per definizione, χ_S è ricorsiva. Da questo segue che sia φ_S che $\varphi_{\mathbb{N} \setminus S}$ sono ricorsive, e quindi che sia S che $\mathbb{N} \setminus S$ sono ricorsivamente enumerabili. Viceversa, supponiamo che S e $\mathbb{N} \setminus S$ siano ricorsivamente enumerabili. Allora φ_S e $\varphi_{\mathbb{N} \setminus S}$ sono ricorsive. Utilizzando i programmi che calcolano φ_S e $\varphi_{\mathbb{N} \setminus S}$, è possibile scrivere un programma che calcola χ_S nel modo seguente: dato un input x , il programma calcola simultaneamente $\varphi_S(x)$ e $\varphi_{\mathbb{N} \setminus S}(x)$ e produce 1 o 0 a seconda di quale programma termina prima. Si noti che almeno uno dei due programmi deve terminare. □

Teorema 5.6. Esiste una funzione ricorsiva $u : \mathbb{N}^3 \rightarrow \mathbb{N}$ tale che, per $(n, k, m) \in \mathbb{N}^3$, si ha che se n è il codice di un programma, m è il codice di una sequenza $((m)_1, \dots, (m)_k)$, e $f_{n,k}((m)_1, \dots, (m)_k) = r$, allora $u(n, k, m) = r$; altrimenti, $u(n, k, m)$ non è definito.

Un programma che calcola la funzione u del Teorema 5.6 è detto un programma *universale*, in quanto è capace di simulare l'azione di un qualsiasi programma per macchine a registri. Applichiamo questo teorema al seguente corollario.

Corollario 5.7. Le inclusioni

$$\{S \mid S \text{ è ricorsivo} \} \subseteq \{S \mid S \text{ è ricorsivamente enumerabile} \} \subseteq \mathcal{P}(\mathbb{N})$$

sono inclusioni strette, ovvero

$$\{S \mid S \text{ è ricorsivo} \} \subsetneq \{S \mid S \text{ è ricorsivamente enumerabile} \} \subsetneq \mathcal{P}(\mathbb{N}).$$

Dimostrazione. Sia u una funzione come nel Teorema 5.6. Definiamo

$$S =_{\text{def}} \{n \in \mathbb{N} \mid u(n, 1, 3^n) \text{ è definito} \}.$$

L'insieme S è ricorsivamente enumerabile perché soddisfa la proprietà (iii) del Lemma 5.3. Verifichiamo che S non è ricorsivo. Abbiamo che

$$\begin{aligned} u(n, 1, 3^n) = r \quad \text{sse} \quad & n \text{ è il codice di un programma, } 3^n \text{ è il codice di} \\ & \text{una sequenza } (3^n)_1 \text{ e } f_{n,1}((3^n)_1) = r \\ & \text{sse} \quad n \text{ è il codice di un programma e } f_{n,1}(n) = r \end{aligned}$$

Quindi,

$$u(u, 1, 3^n) = \begin{cases} f_{n,1}(n) & \text{se } n \text{ è il codice di un programma} \\ \text{non è definito} & \text{altrimenti} \end{cases}$$

Da questo segue che la funzione non ricorsiva $g : \mathbb{N} \rightarrow \mathbb{N}$ definita nella Sezione 4, può essere descritta come

$$g(n) = \begin{cases} u(n, 1, 3^n) + 1 & \text{se } n \text{ è il codice di un programma} \\ 0 & \text{altrimenti} \end{cases}$$

Se S fosse ricorsivo, allora anche g sarebbe ricorsiva. Quindi, S non è ricorsivo, come volevasi dimostrare.

Diamo adesso un esempio di un insieme non ricorsivamente enumerabile. Sia

$$S =_{\text{def}} \{n \in \mathbb{N} \mid f_{n,1} \text{ è totale} \}.$$

Se S fosse ricorsivamente enumerabile, esisterebbe una funzione ricorsiva totale $h : \mathbb{N} \rightarrow \mathbb{N}$ tale che $S = \{h(n) \mid n \in \mathbb{N}\}$ (questo è un fatto che non dimostreremo). Allora, possiamo definire

$$h'(n) = f_{h(n),1} + 1$$

Si noti che

$$h'(n) = f_{h(n),1} + 1 = u(h(n), 1, 3^n) + 1$$

A questo punto avremmo una contraddizione tra il fatto che h' è ricorsiva e totale e il fatto che $h' \neq f_{n,1}$ per ogni $n \in S$. Quindi, S non può essere ricorsivo, come volevasi dimostrare. \square

6 Definibilità in PA

Proposizione 6.1. Ogni funzione polinomiale è definibile in PA.

Dimostrazione. Considereremo, per semplicità, solo il caso di funzioni polinomiali $f : \mathbb{N} \rightarrow \mathbb{N}$. In questo caso, abbiamo

$$f(a) = c_n \cdot a^n + \dots c_1 \cdot a + c_0$$

ove $c_0, \dots, c_n \in \mathbb{N}$. Chiaramente, la somma e il prodotto sono definibili in PA. Visto che le funzioni definibili sono chiuse rispetto all'operazione di composizione, rimane da dimostrare che, per ogni $n \in \mathbb{N}$, la funzione

$$\begin{array}{ccc} \mathbb{N} & \rightarrow & \mathbb{N} \\ a & \mapsto & a^n \end{array}$$

Lo dimostriamo per induzione su n .

Caso base. È sufficiente dimostrare che la funzione costantemente 1 è definibile. A tal fine, si osservi che tale funzione è definita dalla formula $\varphi_0(x, y)$ data da $y = S(0)$.

Passo induttivo. Supponiamo che la funzione $a \mapsto a^n$ sia definita dalla formula $\varphi_n(x, y)$. La funzione $a \mapsto a^{n+1}$ è definita dalla formula $\varphi_{n+1}(x, z)$ data da

$$\exists y(\varphi_n(x, y) \wedge z = x \cdot y).$$

Per ipotesi induttiva, se a, b sono tali che $\llbracket \varphi_n(x, y) \rrbracket_{[x \mapsto a, y \mapsto b]} = 1$, allora $b = a^n$. Quindi, se abbiamo anche c tale che $\llbracket z = x \cdot y \rrbracket_{[x \mapsto a, y \mapsto b, z \mapsto c]} = 1$, allora deve valere che $c = a \cdot b$ e quindi che $c = a \cdot a^n = a^{n+1}$, come volevasi dimostrare. \square

Teorema 6.2. Ogni funzione ricorsiva è definibile in PA.

Traccia della dimostrazione. Si dimostra che l'insieme delle funzioni definibili soddisfa le proprietà (i)–(v) con cui abbiamo definito l'insieme delle funzioni ricorsive. Discutiamo i diversi casi separatamente.

- (i) Dobbiamo dimostrare che, per $1 \leq i \leq k$, la proiezione $\pi_i : \mathbb{N}^k \rightarrow \mathbb{N}$ è definibile. La formula che la definisce è data da $y = x_i$.
- (ii) La formula $c_0 : \mathbb{N} \rightarrow \mathbb{N}$ costantemente 0 è definita da $y = 0$. La funzione successore $S : \mathbb{N} \rightarrow \mathbb{N}$ è definita da $y = S(x)$.
- (iii) Le funzioni definibili sono sempre chiuse rispetto all'operazione di composizione. Un caso particolare è trattato negli esercizi.
- (iv) Non è parte del programma.

- (v) Dobbiamo dimostrare che se $f : \mathbb{N}^{k+1} \rightarrow \mathbb{N}$ è definibile, lo è anche la funzione $g : \mathbb{N}^k \rightarrow \mathbb{N}$ tale che $g(a_1, \dots, a_k) = b$ se e solo se $f(a_1, \dots, a_k, b) = 0$ e, per ogni $b' < b$, si ha $f(a_1, \dots, a_k, b') \neq 0$. Quindi, se $\varphi(x_1, \dots, x_k, y, z)$ è la formula che definisce f , la funzione g è definita dalla formula

$$\varphi(x_1, \dots, x_k, y, 0) \wedge \forall y' (y' < y \Rightarrow \exists z (\neg z = 0 \wedge \varphi(x_1, \dots, x_k, y', z)))$$

Questo conclude la dimostrazione del teorema. \square

7 Esercizi

Esercizio 7.1.

- (i) Si definiscano un numero $a \in \mathbb{N}$ ed una funzione ricorsiva $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ tali che la funzione ricorsiva $h : \mathbb{N} \rightarrow \mathbb{N}$ definita da

$$\begin{cases} h(0) &= a, \\ h(n+1) &= g(n, h(n)), \end{cases}$$

soddisfi $h(n) = n^2 + 1$ per ogni $n \in \mathbb{N}$.

- (ii) Si definiscano funzioni ricorsive $f : \mathbb{N} \rightarrow \mathbb{N}$ e $g : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ tali che la funzione ricorsiva $h : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ definita da

$$\begin{cases} h(m, 0) &= f(m), \\ h(m, n+1) &= g(m, n, h(m, n)), \end{cases}$$

soddisfi $h(m, n) = m^2 + n^2$, per ogni $m, n \in \mathbb{N}$.

- (iii) Si definiscano due funzioni ricorsive $f : \mathbb{N} \rightarrow \mathbb{N}$ e $g : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ tali che la funzione ricorsiva $h : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ definita da

$$\begin{cases} h(m, 0) &= f(m), \\ h(m, n+1) &= g(m, n, h(m, n)), \end{cases}$$

soddisfi $h(m, n) = m + n^3$, per ogni $m, n \in \mathbb{N}$.

Esercizio 7.2.

- (i) Si definisca un programma per macchine a registri che calcoli la funzione $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ definita da

$$f(m, n) = \begin{cases} m - n, & \text{se } m \geq n, \\ 0, & \text{altrimenti.} \end{cases}$$

- (ii) Si descriva un programma per macchine a registri che calcoli la funzione parziale $f : \mathbb{N}^3 \rightarrow \mathbb{N}$ definita da

$$f(m, n, p) = \begin{cases} m - n - p, & \text{se } m \geq n + p, \\ \text{non definita,} & \text{altrimenti.} \end{cases}$$

- (iii) Si descriva un programma per macchine a registri che calcoli la funzione $f : \mathbb{N}^3 \rightarrow \mathbb{N}$ definita da

$$f(m, n, p) = \begin{cases} m - n - p & \text{se } m \geq n + p, \\ 0 & \text{altrimenti.} \end{cases}$$

- (iv) Si illustri la differenza tra i programmi in (ii) e (iii) con opportuni esempi.

Esercizio 7.3.

- (i) Si dimostri che se S_1 e S_2 sono insiemi ricorsivi, allora $S_1 \cap S_2$ è ricorsivo.
(ii) Si dimostri che se $S \subseteq \mathbb{N}$ è ricorsivo, allora $\mathbb{N} \setminus S$ è ricorsivo.

Esercizio 7.4.

- (i) Si dimostri che $S = \{n \in \mathbb{N} \mid n \text{ è pari}\}$ è ricorsivamente enumerabile.
(ii) Si dimostri che $S = \{n \in \mathbb{N} \mid n \text{ è pari}\}$ è ricorsivo.

Esercizio 7.5.

- (i) Si dimostri che l'insieme $\{n \in \mathbb{N} \mid n \geq 4\}$ è ricorsivamente enumerabile.
(ii) Si dimostri che l'insieme $S = \{n \in \mathbb{N} \mid n \text{ pari}\}$ è ricorsivo.

Capitolo 4

Teoria degli insiemi

1 La teoria degli insiemi di Zermelo-Fraenkel

La teoria degli insiemi di Zermelo-Fraenkel (ZF) è una teoria del primo ordine nel linguaggio che contiene, oltre a variabili, il solo simbolo di predicato binario \in , detto *appartenenza*. Il linguaggio non contiene né costanti né simboli di funzione. Per convenienza, utilizzeremo sia x, y, z, \dots che a, b, c, \dots come variabili. Come al solito, definiamo

$$a \subseteq b =_{\text{def}} \forall x (x \in a \Rightarrow x \in b).$$

Elenchiamo gli assiomi di ZF, dando una breve spiegazione informale per ciascuno di essi. L'assiomatizzazione di ZF che presentiamo non è minimale, nel senso che è possibile derivare alcuni assiomi da altri, ma facilita l'esposizione.

(A1) **Assioma di estensionalità.**

$$\forall a \forall b [\forall x (x \in a \Leftrightarrow x \in b) \Leftrightarrow a = b].$$

Questo assioma dice che due insiemi sono uguali se e solo se hanno gli stessi elementi.

(A2) **Assioma dell'insieme vuoto.**

$$\exists a \forall x (x \in a \Leftrightarrow \perp)$$

Questo assioma dice che esiste un insieme a tale che

$$\forall x (x \in a \Leftrightarrow \perp).$$

Per l'assioma di estensionalità, un tale insieme a è unico. Possiamo quindi introdurre un simbolo speciale per indicarlo. Come al solito, utilizzeremo il simbolo \emptyset .

(A3) **Assioma di separazione.** Questo assioma è uno schema, ovvero una famiglia infinita di assiomi, uno per ogni formula $\varphi(x)$ del linguaggio. Data una formula $\varphi(x)$, abbiamo l'assioma

$$\forall a \exists b \left[\forall x (x \in b \Leftrightarrow x \in a \wedge \varphi(x)) \right].$$

Questo assioma dice che per ogni insieme a esiste un insieme b tale che

$$\forall x(x \in b \Leftrightarrow x \in a \wedge \varphi(x)).$$

Come prima, un tale insieme b è necessariamente unico. Nel seguito, sarà indicato con la notazione $\{x \in a \mid \varphi(x)\}$.

(A4) **Assioma della coppia.**

$$\forall a \forall b \exists c [\forall x (x \in c \Leftrightarrow x = a \vee x = b)].$$

Questo assioma dice che per ogni due insiemi a e b esiste un insieme c tale che

$$\forall x(x \in c \Leftrightarrow x = a \vee x = b).$$

Un tale insieme c è necessariamente unico, sempre per l'assioma di estensionalità. Lo indicheremo con la notazione $\{a, b\}$. Nel caso $a = b$, scriveremo semplicemente $\{a\}$ anziché $\{a, a\}$.

(A5) **Assioma dell'unione.**

$$\forall a \exists b [\forall x (x \in b \Leftrightarrow \exists y \in a (x \in y))].$$

Questo assioma dice che per ogni insieme a esiste un insieme b tale che

$$\forall x(x \in b \Leftrightarrow \exists y \in a (x \in y)).$$

Ancora una volta, un tale insieme b è necessariamente unico e noi lo indicheremo con $\bigcup a$. Per ogni coppia di insiemi a e b , utilizzando l'assioma della coppia e l'assioma dell'unione, possiamo definire l'unione di a e b fissando $a \cup b =_{\text{def}} \bigcup \{a, b\}$.

(A6) **Assioma delle parti.**

$$\forall a \exists b [\forall x (x \in b \Leftrightarrow x \subseteq a)].$$

Questo assioma dice che per ogni insieme a esiste un insieme b tale che

$$\forall x(x \in b \Leftrightarrow x \subseteq a).$$

Anche in questo caso, un tale b è unico e noi lo indicheremo con $\mathcal{P}(a)$.

(A7) **Assioma dell'infinito.**

$$\exists a [\emptyset \in a \wedge \forall x (x \in a \Rightarrow x \cup \{x\} \in a)].$$

Come vedremo, questo assioma implica l'esistenza dell'insieme dei numeri naturali.

(A8) **Assioma di fondazione.**

$$\forall a [a \neq \emptyset \Rightarrow \exists x \in a (\forall y (y \in x \Rightarrow y \notin a))].$$

Per spiegare questo assioma introduciamo della terminologia ausiliaria. Diremo che una relazione binaria, data da una formula $R(x, y)$ con variabili libere x e y , è *ben fondata* se ogni insieme non vuoto ha un elemento R -minimale, ovvero vale che

$$\forall a [a \neq \emptyset \Rightarrow \exists x \in a \forall y (R(x, y) \Rightarrow y \notin a)].$$

L'assioma di fondazione dice che la relazione di appartenenza è ben fondata, ovvero che non esistono catene discendenti infinite del tipo

$$\dots \in x_{n+1} \in x_n \in \dots \in x_1 \in a.$$

ove $x_1, \dots, x_n, \dots \in a$.

(A9) **Assioma di Rimpiazzamento.**

$$\forall a \left[\forall x \in a \exists ! y \varphi(x, y) \Rightarrow \exists b \left[\forall x \in a \exists ! y \in b \varphi(x, y) \right] \right].$$

Questo assioma è uno schema come l'assioma di separazione. Consideriamo una formula $\varphi(x, y)$ fissata. L'assioma implica che, per ogni insieme a , se

$$\forall x \in a \exists ! y \varphi(x, y), \quad (*)$$

allora esiste un insieme b tale che

$$\forall y (\exists x \in a \varphi(x, y) \Rightarrow y \in b).$$

Per l'assioma di separazione, è possibile definire un insieme c tale che

$$\forall y (y \in c \Leftrightarrow \exists x \in a \varphi(x, y)). \quad (**)$$

L'ipotesi in (*) esprime che la formula $\varphi(x, y)$ descrive un'operazione con dominio l'insieme a , mentre l'insieme (**) rappresenta l'immagine di questa operazione. L'assioma di rimpiazzamento, quindi, ci permette di concludere che l'immagine di un'operazione con dominio un insieme è un insieme.

Riassumiamo la definizione degli insiemi definiti finora elencando le loro *proprietà caratteristiche*, ovvero formule che ne determinano completamente gli elementi.

$$\begin{aligned} \forall x (x \in \emptyset &\Leftrightarrow \perp), \\ \forall x (x \in \{a, b\} &\Leftrightarrow x = a \vee x = b), \\ \forall x (x \in \bigcup a &\Leftrightarrow \exists y \in a (x \in y)), \\ \forall x (x \in a \cup b &\Leftrightarrow x \in a \vee x \in b), \\ \forall x (x \in \{x \in a \mid &\varphi(x)\} \Leftrightarrow x \in a \wedge \varphi(x)), \\ \forall x (x \in \mathcal{P}(a) &\Leftrightarrow x \subseteq a). \end{aligned}$$

Gli assiomi di ZF ci permettono di definire anche altri insiemi comunemente utilizzati nella pratica matematica. Per esempio, dati due insiemi a e b , l'assioma di separazione ci permette di definire nuovi insiemi $a \cap b$, l'*intersezione* di a e b , e $a \setminus b$, la *differenza* tra a e b , le cui proprietà caratteristiche sono

$$\begin{aligned}\forall x (x \in a \cap b &\Leftrightarrow x \in a \wedge x \in b), \\ \forall x (x \in a \setminus b &\Leftrightarrow x \in a \wedge x \notin b).\end{aligned}$$

Dato un insieme non vuoto a , è possibile definire un insieme $\bigcap a$ tale che

$$\forall x (x \in \bigcap a \Leftrightarrow (\forall y \in a) x \in y).$$

Per convenzione, se $a = \emptyset$ si definisce $\bigcap a =_{\text{def}} \emptyset$.

Proposizione 1.1. Le seguenti uguaglianze e inclusioni sono teoremi di ZF.

$$\begin{aligned}a \cup a &= a, \\ a \cap a &= a, \\ (a \cap b) \cap c &= a \cap (b \cap c), \\ (a \cup b) \cup c &= a \cup (b \cup c), \\ a \cup (b \cap c) &= (a \cup b) \cap (a \cup c), \\ a \cap (b \cap c) &= (a \cap b) \cap (a \cap c), \\ a \setminus (b \cup c) &= (a \setminus b) \cap (a \setminus c), \\ a \setminus (b \cap c) &= (a \setminus b) \cup (a \setminus c), \\ \left(\bigcup a\right) \cup \left(\bigcup b\right) &= \bigcup (a \cup b), \\ \left(\bigcap a\right) \cap \left(\bigcap b\right) &\subseteq \bigcap (a \cap b).\end{aligned}$$

Dimostrazione. Esercizio. □

Nel seguito, utilizzeremo spesso la notazione delle classi. Formalmente, una *classe* è una formula del linguaggio di ZF con una variabile libera. Rappresenteremo la classe data da una formula $\varphi(x)$ con la notazione $\{x \mid \varphi(x)\}$. Un *elemento* di una classe $A = \{x \mid \varphi(x)\}$ è un insieme x per cui vale $\varphi(x)$. Nel seguito, scriveremo $x \in A$ per indicare che x è un elemento di A . Due classi sono uguali se e solo se hanno gli stessi elementi. Un insieme a può essere sempre visto come una classe, ovvero la classe dei suoi elementi $\{x \mid x \in a\}$, ma il viceversa non vale sempre. Una classe $A = \{x \mid \varphi(x)\}$ è un insieme se e solo se esiste un insieme a che ha gli stessi elementi di A , ovvero

$$\forall x (x \in a \Leftrightarrow \varphi(x)).$$

Esistono classi che non sono insiemi. Per esempio, la classe di tutti gli insiemi

$$V =_{\text{def}} \{x \mid x = x\}$$

non è un insieme (a meno che ZF non sia inconsistente). Infatti, supponiamo che esista un insieme a tale che

$$\forall x (x \in a \Leftrightarrow x = x).$$

Per l'assioma di separazione, possiamo definire $b =_{\text{def}} \{x \in a \mid x \notin x\}$. Dalla definizione di a , segue che

$$b = \{x \mid x \notin x\}.$$

Ma adesso abbiamo $b \in b \Leftrightarrow b \notin b$, una contraddizione. Una classe che non è un insieme è detta una classe *propria*. Oltre a V , incontreremo un altro esempio di classe propria, la classe degli ordinali.

2 Esempi della codifica della matematica in ZF

Diamo tre esempi di come sia possibile codificare concetti familiari della pratica matematica all'interno di ZF: i numeri naturali, il prodotto cartesiano di due insiemi e l'insieme di funzioni da un insieme ad un altro.

Per cominciare, dimostriamo l'esistenza dell'insieme \mathbb{N} dei numeri naturali. Definiamo $0 =_{\text{def}} \emptyset$. Dato un insieme x , definiamo il *successore* di x come l'insieme $x + 1 =_{\text{def}} x \cup \{x\}$. Dato un insieme a , definiamo la formula

$$\text{Ind}(a) =_{\text{def}} 0 \in a \wedge \forall x (x \in a \Rightarrow x + 1 \in a).$$

Un insieme a per cui vale $\text{Ind}(a)$ verrà detto *induttivo*. Vogliamo definire l'insieme dei numeri naturali come il più piccolo insieme induttivo. A questo fine, definiamo la formula

$$\text{Nat}(x) =_{\text{def}} \forall a [\text{Ind}(a) \Rightarrow x \in a].$$

Proposizione 2.1.

- (i) La classe $\mathbb{N} =_{\text{def}} \{x \mid \text{Nat}(x)\}$ è un insieme.
- (ii) Il principio di induzione per i numeri naturali,

$$\left[\varphi(0) \wedge (\forall x \in \mathbb{N})(\varphi(x) \Rightarrow \varphi(x + 1)) \right] \Rightarrow (\forall x \in \mathbb{N}) \varphi(x),$$

è un teorema di ZF.

Dimostrazione. Per (i), l'assioma dell'infinito ci dice che esiste un insieme induttivo a . Vogliamo dimostrare che $\mathbb{N} = \{x \in a \mid \text{Nat}(x)\}$, da cui segue quanto richiesto. L'unica cosa da dimostrare è che se vale $\text{Nat}(x)$ allora $x \in a$. Visto che a è induttivo, vale $\forall x [\text{Nat}(x) \Rightarrow x \in a]$, che implica quanto desiderato. Per (ii), ragioneremo in maniera informale. Supponiamo che valgano $\varphi(0)$ e $(\forall x \in \mathbb{N})(\varphi(x) \Rightarrow \varphi(x + 1))$. Definiamo

$$a =_{\text{def}} \{x \in \mathbb{N} \mid \varphi(x)\}$$

Le ipotesi implicano che a è induttivo. Quindi vale che $\forall x [\text{Nat}(x) \Rightarrow x \in a]$. Da questo segue che $(\forall x \in \mathbb{N}) \varphi(x)$, come volevasi dimostrare. \square

Per definire il prodotto cartesiano di due insiemi, cominciamo col definire la nozione di *coppia ordinata* utilizzando la cosiddetta codifica di Kuratowski. Dati x e y , la coppia ordinata (x, y) è definita da:

$$(x, y) =_{\text{def}} \{\{x\}, \{x, y\}\}.$$

Si noti che se $x \in a$ e $y \in b$, allora $(x, y) \in \mathcal{P}(\mathcal{P}(a \cup b))$. Vogliamo adesso definire le operazioni di proiezione π_1 e π_2 , in modo tale che $\pi_1(x, y) = x$ e $\pi_2(x, y) = y$. A tal fine, definiamo

$$\begin{aligned} \pi_1(z) &=_{\text{def}} \begin{cases} \bigcup \bigcap z & \text{se } z \neq \emptyset, \\ \emptyset & \text{altrimenti.} \end{cases} \\ \pi_2(z) &=_{\text{def}} \begin{cases} \bigcup (\bigcup z \setminus \bigcap z) & \text{se } \bigcup z \setminus \bigcap z \neq \emptyset, \\ \pi_1(z) & \text{altrimenti.} \end{cases} \end{aligned}$$

Lemma 2.2.

- (i) Se $(x, y) = (u, v)$ allora $x = u$, $y = v$.
- (ii) $\pi_1(x, y) = x$, $\pi_2(x, y) = y$.

Dimostrazione. La dimostrazione di (i) è lasciata come esercizio. Per quanto riguarda (ii), dimostriamo le due uguaglianze separatamente. Per la prima, abbiamo

$$\begin{aligned} \pi_1(x, y) &= \pi_1(\{\{x\}, \{x, y\}\}) \\ &= \bigcup \bigcap \{\{x\}, \{x, y\}\} \\ &= \bigcup (\{x\} \cap \{x, y\}) \\ &= \bigcup \{x\} \\ &= x. \end{aligned}$$

Per la seconda, osserviamo che

$$\begin{aligned} \bigcup (\bigcup (x, y) \setminus \bigcap (x, y)) &= \bigcup \{\{x\}, \{x, y\}\} \setminus \bigcap \{\{x\}, \{x, y\}\} \\ &= \{x, y\} \setminus \{x\} \\ &= \{y\} \\ &\neq \emptyset \end{aligned}$$

Quindi,

$$\begin{aligned} \pi_2(x, y) &= \pi_2 \bigcup \{\{x\}, \{x, y\}\} \setminus \bigcap \{\{x\}, \{x, y\}\} \\ &= \bigcup \{y\} \\ &= y \end{aligned}$$

□

Proposizione 2.3. Per ogni coppia di insiemi a e b , la classe

$$a \times b =_{\text{def}} \{z \mid (\exists x \in a) (\exists y \in b) z = (x, y)\}$$

è un insieme.

Dimostrazione. Si osservi che se $x \in a$ e $y \in b$, allora $(x, y) \in \mathcal{P}(\mathcal{P}(a \cup b))$. Da questo segue che

$$a \times b = \{z \in \mathcal{P}(\mathcal{P}(a \cup b)) \mid z = (\pi_1(z), \pi_2(z)) \wedge \pi_1(z) \in a \wedge \pi_2(z) \in b\}$$

La conclusione segue dall'assioma di separazione. \square

Concludiamo questa sezione mostrando come sia possibile definire insiemi di funzioni in ZF. Siano a e b due insiemi. Una *relazione* tra a e b è un sottoinsieme $r \subseteq a \times b$. Diremo che una relazione $r \subseteq a \times b$ è una *funzione* da a a b se vale che

- (i) r è *totale*, $(\forall x \in a) (\exists y \in b) (x, y) \in r$.
- (ii) r è *univoca*, $(\forall x \in a) (\forall y_1, y_2 \in b) [(x, y_1) \in r \wedge (x, y_2) \in r \Rightarrow y_1 = y_2]$.

Chiaramente, esiste una formula $\text{Fun}(r, a, b)$ del linguaggio di ZF che esprime che r è una funzione da a a b .

Proposizione 2.4. Per ogni due insiemi a e b , la classe

$$b^a =_{\text{def}} \{r \mid \text{Fun}(r, a, b)\}$$

è un insieme.

Dimostrazione. Visto che una relazione tra a e b è un sottoinsieme di $a \times b$, abbiamo che $b^a = \{r \in \mathcal{P}(a \times b) \mid \text{Fun}(r, a, b)\}$. Quindi, b^a è un insieme per l'assioma delle parti e l'assioma di separazione. \square

Nel seguito, continueremo ad utilizzare la notazione abituale per le funzioni. In particolare, scriveremo $f : a \rightarrow b$ per indicare che f è una funzione da a a b . Data $f : a \rightarrow b$ e $x \in a$, scriveremo $f(x)$ per indicare l'unico elemento $y \in b$ tale che $(x, y) \in f$.

3 Induzione insiemistica

Proposizione 3.1. I seguenti principi sono equivalenti.

- (i) Principio di induzione insiemistica,

$$\forall a [\forall x \in a \varphi(x) \Rightarrow \varphi(a)] \Rightarrow \forall a \varphi(a).$$

- (ii) V è la piú piccola classe X tale che se $a \subseteq X$ allora $a \in X$.

Dimostrazione. Assumiamo che valga il principio di induzione insiemistica. Sia $X = \{x \mid \varphi(x)\}$ una classe. Supponiamo che valga che, per ogni a , se $a \subseteq X$ allora $a \in X$. Questo significa che vale

$$\forall x \in a \varphi(x) \Rightarrow \varphi(a)$$

Per il principio di induzione insiemistica, vale allora che $\forall a \varphi(a)$. Ma questo implica $V \subseteq X$, come volevasi dimostrare. Viceversa, supponiamo (ii) e assumiamo la premessa del principio di induzione insiemistica, ovvero

$$\forall a [\forall x \in a \varphi(x) \Rightarrow \varphi(a)]$$

Definiamo $X = \{x \mid \varphi(x)\}$. L'assunzione implica che, per ogni a , se $a \subseteq X$ allora $a \in X$. Ma V , per (ii), è la piú piccola classe con questa proprietá e quindi $V \subseteq X$. Da questo segue $\forall a \varphi(a)$, come volevasi dimostrare. \square

Vogliamo dimostrare che gli assiomi di ZF dimostrano il principio di induzione insiemistico. A questo fine, introduciamo la nozione di insieme transitivo.

Definizione 3.2. Un insieme a è detto essere *transitivo* se vale che

$$\forall x (x \in a \Rightarrow x \subseteq a).$$

Dato un insieme a , la *chiusura transitiva* di a è il piú piccolo insieme transitivo che contiene a . Questo insieme, che indicheremo con $\text{TC}(a)$, può essere definito esplicitamente come

$$\text{TC}(a) = \{a, \bigcup a, \bigcup \bigcup a, \dots\}.$$

Proposizione 3.3. $\text{ZF} \vdash$ Principio di Induzione Insiemistica.

Dimostrazione. Supponiamo che valga

$$\forall a [\forall x \in a \varphi(x) \Rightarrow \varphi(a)]. \quad (*)$$

Dobbiamo dimostrare che vale $\forall a \varphi(a)$. Per assurdo, si supponga che non valga, e che quindi esista a tale che $\neg \varphi(a)$. Consideriamo adesso l'insieme

$$b = \{x \in \text{TC}(\{a\}) \mid \neg \varphi(x)\}.$$

Vale che $a \in b$ e quindi b non è vuoto. Per l'assioma di fondazione, deve avere un elemento \in -minimale, diciamo x . Visto che $x \in b$, deve valere $\neg \varphi(x)$. Ma vale anche che

$$\forall y \in x \varphi(y),$$

visto che gli elementi di x sono in $\text{TC}(\{a\}) \setminus b$. Ma allora, dall'ipotesi in (*), segue $\varphi(x)$, una contraddizione. \square

La dimostrazione della Proposizione 3.3 utilizza in maniera essenziale l'assioma di fondazione. Sulla base degli altri assiomi di ZF, infatti, l'assioma di fondazione è equivalente al principio di induzione insiemistico.

4 Ordinali

Definizione 4.1. Sia $<$ una relazione binaria su di un insieme a . Diremo che $<$ è

- *irriflessiva*, se vale che $\forall x \in a (x \not< x)$,
- *antisimmetrica*, se vale che $\forall x, y \in a (x < y \Rightarrow y \not< x)$
- *transitiva*, se vale che $\forall x, y, z \in a (x < y < z \Rightarrow x < z)$.
- *tricotoma*, se vale che $\forall x, y \in a ((x < y) \vee (y < x) \vee (x = y))$.

Si osservi che se una relazione è irriflessiva e transitiva, allora è antisimmetrica.

Definizione 4.2.

- (i) Una relazione binaria $<$ è detta essere un *ordine totale* (oppure un *ordine lineare*) se è irriflessiva, antisimmetrica, transitiva, e tricotoma.
- (ii) Una relazione binaria $<$ su di un insieme a è detta essere un *buon ordinamento* se è un ordine totale che ha un elemento minimo, ovvero esiste un elemento di $x \in a$ tale che $\forall y \in a (x \leq y)$

Proposizione 4.3.

- (i) Se una relazione è ben fondata, allora è irriflessiva.
- (ii) Se una relazione è ben fondata e tricotoma, allora è transitiva.
- (iii) Se una relazione è ben fondata, allora è un buon ordinamento se e solo se è tricotoma.

Dimostrazione. Per dimostrare (i), assumiamo che $<$ sia ben fondata. Se non fosse irriflessiva, esisterebbe x tale che $x < x$. Ma allora potremmo costruire la catena infinita:

$$\dots x < x < \dots x < x$$

contraddicendo la ben fondatezza della relazione. Per dimostrare (ii), supponiamo che $<$ sia ben fondata e tricotoma. Supponiamo che valgano $x < y$ e $y < z$. Se non valesse che $x < z$, allora l'insieme $\{x, y, z\}$ non avrebbe un elemento $<$ -minimale, in contraddizione con la ben fondatezza della relazione. Per dimostrare (iii), supponiamo che $<$ sia ben fondata e tricotoma. Da (i) segue che $<$ è irriflessiva; da (ii) segue che $<$ è transitiva, e quindi $<$ è antisimmetrica. Rimane da dimostrare che ha un elemento minimo. Ma se non lo avesse, non sarebbe ben fondata. L'implicazione inversa è parte della definizione di buon ordinamento. \square

Definizione 4.4. Un *ordinale* è un insieme transitivo per cui la relazione di appartenenza è un buon ordinamento.

Ci sono essenzialmente tre tipi di ordinali: lo zero, 0 , gli ordinali successivi, $\alpha + 1$, e gli ordinali limite, $\bigcup \alpha$. Gli ordinali finiti sono dati dai numeri naturali $0, 1, 2, \dots$. Il primo ordinale limite, tradizionalmente indicato con ω , è dato dall'insieme dei numeri naturali (si verifichi per esercizio che è un ordinale). Si può dimostrare che la classe \mathcal{O} degli ordinali è la più piccola classe X che soddisfa le seguenti proprietà:

- (i) $\forall x (x \in X \Rightarrow x + 1 \in X)$,
- (ii) $\forall x (x \subseteq X \Rightarrow \bigcup x \in X)$.

Da questa caratterizzazione induttiva della classe degli ordinali, segue che possiamo dimostrare proprietà degli ordinali per induzione e definire funzioni dalla classe degli ordinali per ricorsione. Un esempio di funzione definita ricorsiva dalla classe degli ordinali permette di definire la *gerarchia di Von Neumann*, data da una funzione

$$\begin{aligned} \mathcal{O} &\rightarrow V \\ \alpha &\mapsto V_\alpha \end{aligned}$$

definita da

$$\begin{aligned} V_0 &= \emptyset \\ V_{\alpha+1} &= \mathcal{P}(V_\alpha) \\ V_{\bigcup \alpha} &= \bigcup \{V_\beta \mid \beta \in \alpha\} \end{aligned}$$

Per ogni insieme a è possibile definire un ordinale $\text{rk}(a)$, detto il *rango* di a , tale che $x \in V_\alpha$ se e solo se $\text{rk}(x) < \alpha$. La totalità degli insiemi può essere quindi vista come organizzata in livelli, dati dagli ordinali. Il rango di un insieme a è esattamente il livello di a e l'insieme V_α ha come elementi tutti gli insiemi con livello inferiore ad α .

5 L'assioma della scelta

Nonostante ZF sia sufficiente per codificare e dimostrare gran parte della matematica, ci sono alcune asserzioni che possono essere dimostrate solo in ZFC, l'estensione di ZF con l'Assioma della Scelta (AC), come per esempio il teorema di Tychonoff (il prodotto cartesiano di una famiglia di spazi topologici compatti è compatto), il teorema dell'ideale massimale (ogni ideale di un anello è contenuto in un ideale massimale), e il teorema di Hamel (ogni spazio vettoriale ha una base). Cominciamo con l'enunciare l'assioma della scelta.

Assioma della scelta. Se $(a_i \mid i \in I)$ è una famiglia di insiemi non vuoti, ovvero $a_i \neq \emptyset$ per ogni $i \in I$, allora esiste una funzione

$$f : I \rightarrow \bigcup \{a_i \mid i \in I\}$$

tale che $f(i) \in a_i$ per ogni $i \in I$.

Proposizione 5.1. Le seguenti asserzioni sono equivalenti.

- (i) Assioma della scelta.
(ii) Ogni insieme a ha una *funzione di scelta*, ovvero una funzione

$$f : \mathcal{P}(a) \setminus \{\emptyset\} \rightarrow a$$

tale che $f(p) \in p$ per ogni $p \in \mathcal{P}(a) \setminus \{\emptyset\}$.

Dimostrazione.

(i) \Rightarrow (ii). Si consideri $I = \mathcal{P}(a) \setminus \{\emptyset\}$ e la famiglia $(p \mid p \in \mathcal{P}(a) \setminus \{\emptyset\})$. Ogni elemento della famiglia è non vuoto, e quindi possiamo applicare l'assioma della scelta e derivare l'esistenza di una funzione

$$f : \mathcal{P}(a) \setminus \{\emptyset\} \rightarrow \bigcup \{p \mid p \in \mathcal{P}(a) \setminus \{\emptyset\}\}$$

tale che $f(p) \in p$ per ogni $p \in \mathcal{P}(a) \setminus \{\emptyset\}$. La conclusione segue osservando che

$$\bigcup \{p \mid p \in \mathcal{P}(a) \setminus \{\emptyset\}\} = a.$$

(ii) \Rightarrow (i). Data una famiglia $(a_i \mid i \in I)$, definiamo

$$a = \bigcup \{a_i \mid i \in I\}.$$

Sia $f : \mathcal{P}(a) \setminus \{\emptyset\} \rightarrow a$ una funzione di scelta per a . La funzione desiderata è data dalla funzione $f' : I \rightarrow \bigcup \{a_i \mid i \in I\}$ è definita da $f'(i) =_{\text{def}} f(a_i)$, per $i \in I$. \square

Proposizione 5.2. Sia a un insieme. Le seguenti affermazioni sono equivalenti.

- (i) Esiste un buon ordinamento di a ,
(ii) Esiste una funzione di scelta su a .

Traccia della dimostrazione. Per l'implicazione (i) \Rightarrow (ii), sia $<$ un buon ordinamento su a . La funzione di scelta $f : \mathcal{P}(a) \setminus \{\emptyset\} \rightarrow a$ è data definendo $f(p)$ come l'elemento $<$ -minimale di p , ove $p \in \mathcal{P}(a) \setminus \{\emptyset\}$. L'implicazione (ii) \Rightarrow (i) non è in programma. \square

Corollario 5.3. I seguenti principi sono equivalenti.

- (i) L'assioma della scelta.
(ii) Ogni insieme ha un buon ordinamento.

Dimostrazione. Conseguenza immediata della proposizione precedente. \square

Vogliamo adesso introdurre il Lemma di Zorn, un principio equivalente all'Assioma della Scelta che viene utilizzato spesso nella pratica matematica. Per enunciarlo, dobbiamo fissare un po' di terminologia. Sia (a, \leq) un ordine parziale (ove $x \leq y \Leftrightarrow x < y \vee x = y$). Una *catena* in a è un sottoinsieme $p \subseteq a$ che è ordinato totalmente dalla restrizione di $<$ a p . In altre parole, vale che $(p, <)$ è un ordine totale. Diremo che (a, \leq) è *induttivo* se per ogni catena $p \subseteq a$ ha

un limite superiore, ovvero esiste $x \in a$ tale che $y \leq x$ per ogni $y \in p$. Infine, diciamo che $m \in a$ è un elemento *massimale* di (a, \leq) se vale che

$$\forall x \in a (m \leq x \Rightarrow m = x).$$

Lemma di Zorn. Se un insieme parzialmente ordinato è induttivo, allora ha un elemento massimale.

Proposizione 5.4. I seguenti principi sono equivalenti.

- (i) L'assioma della scelta.
- (ii) Il lemma di Zorn.

Dimostrazione. L'implicazione (i) \Rightarrow (ii) non è in programma. L'implicazione (ii) \Rightarrow (i) è un tipico esempio di applicazione del lemma di Zorn. Sia a un insieme. Vogliamo definire una funzione di scelta su a . La strategia della dimostrazione è di definire un insieme parzialmente induttivo il cui elemento massimale è la funzione richiesta. A questo fine, consideriamo l'insieme delle funzioni f con dominio un sottoinsieme di $\mathcal{P}(a) \setminus \{\emptyset\}$ tali che $f(p) \in p$, per ogni $p \in \text{dom}(f)$. Questo insieme ha un ordine parziale definito dichiarando $f_1 \leq f_2$ se f_2 estende f_1 , ovvero $\text{dom}(f_1) \subseteq \text{dom}(f_2)$ e $f_1(p) = f_2(p)$ per ogni $p \in \text{dom}(f_1)$. Questo ordine parziale è induttivo e quindi, per il lemma di Zorn, ha un elemento massimale, diciamo f . Vogliamo dimostrare che $\text{dom}(f) = \mathcal{P}(a) \setminus \{\emptyset\}$. Se non valesse, esisterebbe $x \in (\mathcal{P}(a) \setminus \{\emptyset\}) \setminus \text{dom}(f)$, ovvero un elemento di $\mathcal{P}(a) \setminus \{\emptyset\}$ che non appartiene al dominio di f . Visto che x non è vuoto, esiste $y \in x$ e quindi possiamo definire una funzione $f' : \text{dom}(f) \cup \{x\} \rightarrow a$ che coincide con f su $\text{dom}(f)$ e definita da $f'(x) = y$ su $\{x\}$. L'esistenza di questa funzione contraddice la massimalità di f . Quindi, deve valere che $\text{dom}(f) = \mathcal{P}(a) \setminus \{\emptyset\}$, come volevasi dimostrare. \square

6 Cardinali

La nozione di cardinale è quanto si ottiene astraendo dalla nozione di insieme il fatto che gli elementi di un insieme sono essi stessi degli insiemi. Più precisamente, vorremmo definire un cardinale come una classe di equivalenza rispetto alla relazione di equivalenza definita da

$$a \sim b \text{ se e solo se esiste una funzione biettiva } f : a \rightarrow b.$$

A priori, le classi di equivalenza di questa relazione di equivalenza sono classi proprie, non insiemi. Tuttavia, è possibile definire una funzione che sceglie un rappresentante per ciascuna di queste classi di equivalenza, ovvero una funzione

$$\text{card} : V \rightarrow V$$

tale che

$$\forall a \forall b (\text{card}(a) = \text{card}(b) \Leftrightarrow a \sim b). \quad (*)$$

Noi non definiremo la funzione card , ma ci limiteremo ad utilizzarla tramite la proprietà in (*). Possiamo adesso definire un cardinale come un insieme della

forma $\text{card}(a)$, ove a è un insieme qualsiasi. Come abituale, utilizzeremo le lettere $\kappa, \lambda, \mu, \dots$ per indicare cardinali. Introduciamo una relazione d'ordine tra i cardinali dichiarando che vale $\kappa \leq \lambda$ se e solo se esiste una funzione iniettiva $i : a \rightarrow b$, ove $\text{card}(a) = \kappa$ e $\text{card}(b) = \lambda$.

Teorema 6.1 (Teorema di Cantor-Schroeder-Bernstein). Se $\kappa \leq \lambda$ e $\lambda \leq \kappa$, allora $\kappa = \lambda$.

Dimostrazione. Non è in programma. Si osservi solo che il risultato segue una volta che si dimostra che, date funzioni iniettive $i : a \rightarrow b$ e $j : b \rightarrow c$, è possibile definire una biiezione $f : a \rightarrow b$. \square

Le operazioni fondamentali dell'aritmetica dei cardinali sono le seguenti:

Somma. $\kappa + \lambda = \text{card}(a + b)$, ove $\text{card}(a) = \kappa$, $\text{card}(b) = \lambda$, e $a + b$ denota l'unione disgiunta di a e b .

Prodotto. $\kappa \cdot \lambda = \text{card}(a \times b)$, ove $\text{card}(a) = \kappa$, $\text{card}(b) = \lambda$, e $a \times b$ denota il prodotto cartesiano di a e b .

Esponenziazione. $\lambda^\kappa = \text{card}(b^a)$, ove $\text{card}(a) = \kappa$, $\text{card}(b) = \lambda$, e b^a denota l'insieme delle funzioni da a a b .

Si osservi che, per ogni insieme a , esiste una biiezione $\chi : \mathcal{P}(a) \rightarrow 2^a$, ove $2 = \{0, 1\}$, definita mappando $p \in \mathcal{P}(a)$ nella funzione caratteristica di p , $\chi_p : a \rightarrow 2$. Da questo segue

$$\text{card}(\mathcal{P}(a)) = \text{card}(2^a) = 2^{\text{card}(a)}.$$

Definiamo $\aleph_0 = \text{card}(\mathbb{N})$ e \aleph_1 come il piú piccolo cardinale maggiore di \aleph_0 . Essendo l'insieme dei numeri reali \mathbb{R} in biiezione con $\mathcal{P}(\mathbb{N})$, vale che

$$\text{card}(\mathbb{R}) = \text{card}(\mathcal{P}(\mathbb{N})) = 2^{\text{card}(\mathbb{N})} = 2^{\aleph_0}$$

L'ipotesi del continuo (CH) è l'asserzione che $2^{\aleph_0} = \aleph_1$. Il teorema seguente contiene due dei risultati piú importanti della logica matematica del XX secolo.

Teorema 6.2.

(i) Se $\text{ZFC} \not\vdash \perp$, allora $\text{ZFC} \not\vdash \neg\text{CH}$.

(ii) Se $\text{ZFC} \not\vdash \perp$, allora $\text{ZFC} \not\vdash \text{CH}$.

L'enunciato (i), dimostrato da Gödel, esprime che gli assiomi di ZFC non possono refutare l'ipotesi del continuo. L'enunciato in (ii), dimostrato da Cohen, esprime che gli assiomi di ZFC non sono sufficienti a dimostrare l'ipotesi del continuo.

7 Esercizi

Esercizio 7.1. Siano a, b, c insiemi. Si dimostrino le seguenti implicazioni.

- (i) Se $b \subseteq a$, allora $a = (a \setminus b) \cup b$.
- (ii) Se $b \subseteq a$, allora $b = a \setminus (a \setminus b)$.
- (iii) Se $b \subseteq a$ e $c \subseteq a$ allora $(a \setminus b) \setminus c = a \setminus (b \cup c)$.

Esercizio 7.2. Siano a, b insiemi. Si dimostrino le seguenti asserzioni.

- (i) Se $a \subseteq b$ allora $\mathcal{P}(a) \subseteq \mathcal{P}(b)$
- (ii) $\mathcal{P}(a \cap b) \subseteq \mathcal{P}(a) \cap \mathcal{P}(b)$.
- (iii) $\mathcal{P}(a) \cup \mathcal{P}(b) \subseteq \mathcal{P}(a \cup b)$.

Esercizio 7.3. Sia a un insieme. Si dimostri che le seguenti asserzioni sono equivalenti.

- (i) a è transitivo.
- (ii) $\bigcup a \subseteq a$.
- (iii) $a \subseteq \mathcal{P}(a)$.
- (iv) $\mathcal{P}(a)$ è transitivo.

Esercizio 7.4. Per ciascuno dei seguenti ordini parziali, si stabilisca se la relazione d'ordine è irreflessiva, antisimmetrica, transitiva, o tricotoma.

- (i) (\mathbb{N}, \prec) , ove $x \prec y$ se e solo se $x \neq y$ e x divide y .
- (ii) $(\mathbb{N} \times \mathbb{N}, \prec)$ ove $(x_1, x_2) \prec (y_1, y_2)$ se e solo se $x_1 < y_1$ e $x_2 < y_2$.
- (iii) $(\mathcal{P}(\mathbb{N}), \subset)$.

Esercizio 7.5. Siano κ, λ, μ cardinali.

- (i) Si dimostri che $(\kappa \cdot \lambda) \cdot \mu = \kappa \cdot (\lambda \cdot \mu)$.
- (ii) Si dimostri che se $\kappa \leq \lambda$ e $\lambda \leq \mu$, allora $\kappa \leq \mu$.
- (iii) Si dimostri che $\kappa \leq \kappa \cdot \kappa$.
- (iv) Si dimostri che se $\kappa \leq \lambda$ allora $\kappa \cdot \mu \leq \lambda \cdot \mu$.

Esercizio 7.6.

- (i) Sia κ un cardinale e sia $2 = \text{card}(\{0, 1\})$. Si dimostri che $\kappa^2 = \kappa \cdot \kappa$.
- (ii) Si dimostri che $\text{card}\{n \in \mathbb{N} \mid n \text{ pari}\} = \aleph_0$.
- (iii) Si dimostri che $\text{card}(\mathbb{Z}) = \aleph_0$.

Bibliografia

- [BBJ07] G. S. Boolos, J. P. Burgess, and R. C. Jeffrey. *Computability and Logic*. Cambridge University Press, 5th edition, 2007.
- [DS96] F. R. Drake and D. Singh. *Intermediate Set Theory*. Wiley, 1996.
- [End01] H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 2nd edition, 2001.
- [Joh87] P. T. Johnstone. *Notes on Logic and Set Theory*. Cambridge University Press, 1987.
- [MvO] I. Moerdijk and J. van Oosten. *Sets, Models and Proofs*. Available from J. van Oosten's web page, 2009.
- [vD08] D. van Dalen. *Logic and Structure*. Springer, 4th edition, 2008.