

# Logica

Corso di Laurea in Informatica - Università degli studi di Verona

FEDERICO BRUTTI

Federico Brutti  
`federico.brutti@studenti.univr.it`

# Indice

## 4 | Introduzione

1.1	Che cos'è la logica matematica? .....	4
1.2	Connettivi e Quantificatori .....	4
1.3	Strumenti di lavoro .....	5

## 6 | Logica Proposizionale

2.1	Sintassi .....	6
2.2	Semantica .....	9
2.3	Sistemi Deduttivi .....	9
2.4	Correttezza e Completezza .....	9
2.5	Esercizi .....	9
2.6	Appunti .....	9
2.6.1	Semantica .....	9
2.6.2	Lavorare con la Logica Proposizionale .....	12
2.6.3	Induzione e Ricorsione .....	12
2.6.4	Tabelle di Verità .....	13
2.6.5	Deduzione Naturale .....	13

## 16 | Logica del Primo Ordine

3.1	Sintassi .....	16
3.2	Sistemi Deduttivi .....	16
3.3	Semantica .....	16
3.4	Esercizi .....	16
3.5	Appunti .....	16
3.5.1	Strutture e Tipi di Similarità .....	16
3.5.2	Derivazione Semantica .....	16
3.5.3	Deduzione Naturale .....	16
3.5.4	Teoremi di Correttezza e Completezza .....	16

*Non ho ancora conosciuto una singola persona a cui sia piaciuta questa  
materia.*

# Introduzione

## 1.1 Che cos'è la logica matematica?

La **Logica Matematica** ha lo scopo di formalizzare concetti matematici in una lingua artificiale, per dare una certezza di significato al linguaggio naturale, il quale risulterebbe ambiguo. Similmente a quest'ultimo, anche i linguaggi della logica si compongono di grammatica, significato e regole, identificati rispettivamente in:

- **Sintassi:** Definisce la corretta scrittura delle formule logiche.
- **Semantica:** Definisce il significato delle formule logiche.
- **Sistemi Deduttivi:** Strumenti sintattici utili a manipolare formule e costruire dimostrazioni, dette derivazioni.

Abbiamo quindi un insieme di parole da assemblare (sintassi) che con diverse combinazioni possono creare sentenze dai diversi significati (semantica) ed il tutto segue un determinato insieme di regole (sistemi deduttivi). Tutto ciò consente di creare teoremi, anch'essi composti da più parti:

- **Enunciato;** Ciò che il teorema vuole esprimere.
- **Dimostrazione;** La prova formalizzata di quanto espresso.

Sarà nostro compito dimostrare gli enunciati in base alle richieste degli esercizi. Servirà specificare una procedura che permetta di ottenere solo risultati veri o falsi in linguaggio logico.

## 1.2 Connettivi e Quantificatori

Qui sono elencati tutti i connettivi e i quantificatori utilizzati nel corso, con lo scopo di familiarizzare con loro a prescindere da quando verranno o meno utilizzati.

- **Connettivi**
  - **Congiunzione**  $\wedge$ :  
Ritorna vero solo se tutti gli elementi sono veri.
  - **Disgiunzione**  $\vee$ :  
Ritorna vero se almeno un elemento è vero.
  - **Negazione**  $\neg$ :  
Rende falso il vero e viceversa.
  - **Implicazione**  $\implies$  :  
Corrisponde a "Se, allora", ritorna vero nei casi  $0 \rightarrow 1$  oppure  $1 \rightarrow 1$ , mentre è falso se  $1 \rightarrow 0$  oppure  $0 \rightarrow 0$ .

- **Doppia Implicazione**  $\Longleftrightarrow$  :  
Corrisponde a "se e solo se, allora" e si rappresenta tramite due implicazioni:  $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ .
- **Bottom**  $\perp$ :  
Indica il valore di assurdo, 0.

- **Quantificatori**

- **Esiste**  $\exists$ :  
Indica l'esistenza di un elemento con una determinata proprietà.
- **Per ogni**  $\forall$ :  
indica che per ogni caso considerato, esiste un elemento con una data proprietà.

## 1.3 Strumenti di lavoro

Introduciamo in linguaggio naturale il funzionamento di tutti i nostri strumenti e strategie che utilizzeremo per la dimostrazione dei teoremi:

- **Sentenze:** Combinazioni dei singoli elementi dell'alfabeto. Equivalgono alle proposizioni in linguaggio naturale e possono assumere solamente due valori: vero o falso. Ne esistono di due tipi:
  - **Minimale;** Quando non si può scomporre ulteriormente.
  - **Composta;** Quando è composta da più sentenze minimali, ed è quindi scomponibile.
- **Induzione Strutturale:** Il principio di induzione è utilizzato per dimostrare la verità dell'enunciato a partire da un passo base semplice, per poi estendere l'ipotesi a ogni istanza successiva. Se l'ipotesi è provata, hai dimostrato l'enunciato.
- **Ricorsione Primitiva:** La ricorsione consiste nella formulazione di funzioni in termini di loro stesse all'interno della loro definizione. Le funzioni di una formula comprendono ogni caso possibile e riportano ad un passo base per terminare il processo.
- **Deduzione Naturale:** Processo di dimostrazione di una data formula a partire da delle ipotesi. Si tratta di un algoritmo basato puramente sulla correttezza sintattica.
- **Tabella di Verità:** Tabella che mostra ogni singolo caso presentabile per un determinato enunciato.
- **Valutazione di Verità:** Nella semantica, il processo di verifica della veridicità di una formula, esaminando il risultato di ogni singola formula presente nell'enunciato.
- **Sostituzione:** La sostituzione con una formula  $\psi$  tutte le occorrenze di un simbolo proposizionale o del primo ordine in un'altra formula  $\phi$ .
- **Modello:** Definizione in termini logico-matematici di una data formula o funzione.
- **Contromodello:** Scrittura in termini logico-matematici che prova la fallacia di una data ipotesi iniziale.

# Logica Proposizionale

Cominciamo lo studio effettivo della Logica Matematica attraverso la **Logica Proposizionale**. Fondamentalmente è una palestra da utilizzare come trampolino di lancio verso argomenti più complessi, come la Logica del Primo Ordine, che verrà vista in seguito.

Al termine del capitolo bisognerà saper valutare la veridicità di una formula e saper utilizzare la deduzione naturale per derivare sentenze.

## 2.1 Sintassi

In questo ambito il ragionamento si compone di entità linguistiche collegate dalla relazione "segue da". Queste si chiamano **Sentenze**. Come visto nell'introduzione possono essere minimali o composte, ma entrambe, se esprimono un pensiero completo si dicono **dichiarative**. Queste devono necessariamente avere un valore di vero o falso per essere considerate tali.

### Esempio. Tipi di sentenze

- Sentenza minimale:  $\phi$   
Entità irriducibile.
- Sentenza composta:  $\phi \vee \psi$   
Entità riducibile.

Ogni linguaggio logico è basato su un determinato **Alfabeto**, determinante i simboli utilizzabili in esso. Infatti, i linguaggi sono definiti come  $L \subseteq A^*$ , dove quest'ultimo insieme è quello delle **Stringhe**, ovvero tutte le sequenze di simboli generabili tramite l'alfabeto.

Le stringhe appartenenti ad  $L$  si dicono sintatticamente corrette. In particolare, definiamo ora l'alfabeto della logica proposizionale che consente la loro composizione:

### Definizione 2.1. Alfabeto $L^{\text{PROP}}$

L'alfabeto della Logica Proposizionale è definito tramite tre unità sintattiche distinte:

- Insieme di simboli proposizionali  $AT^{\text{PROP}}$ , i cui elementi saranno indicati con lettere greche.
- Connettivi  $\wedge, \vee, \rightarrow, \perp$ .
- Simboli ausiliari "(" e ")". Non variano il senso della frase, ma aiutano a renderla più leggibile.

Abbiamo constatato che il linguaggio proposizionale è costituito dall'insieme delle stringhe sintatticamente corrette. Si indica con **PROP** e si definisce induttivamente come segue:

**Definizione 2.2. Insieme delle proposizioni PROP**

- $\perp \in \text{PROP}$ ;
- Se  $p$  è un simbolo proposizionale, allora  $p \in \text{PROP}$ ;
- Se  $\phi, \psi \in \text{PROP}$ , allora valgono le seguenti relazioni:  
 $(\phi \wedge \psi) \in \text{PROP}$ ,  $(\phi \vee \psi) \in \text{PROP}$ ,  $(\phi \implies \psi) \in \text{PROP}$ ,  $(\neg \phi) \in \text{PROP}$ .
- Null'altro appartiene all'insieme PROP.

Per renderci la vita un pochino più semplice sono state ideate delle convenzioni sulla scrittura delle proposizioni; si possono infatti omettere alcune parentesi nelle formule e fissare un ordine di precedenza tra i connettivi:

- Si omettono le parentesi più esterne di una formula, così al posto di scrivere " $(\phi \wedge \psi)$ ", si scrive solo " $\phi \wedge \psi$ ".
- Il connettivo  $\neg$  è il più forte e associa il primo simbolo presente a destra, perciò " $\neg \phi \vee \psi$ " sarebbe uguale a " $(\neg \phi) \vee \psi$ ".
- Si assume che  $\wedge$  e  $\vee$  siano connettivi più forti rispetto a  $\implies$  e  $\iff$ , quindi scrivere " $\phi_1 \wedge \psi_1 \implies \phi_2 \vee \psi_2$ " è equivalente a intendere " $(\phi_1 \wedge \psi_1) \implies (\phi_2 \vee \psi_2)$ ".
- I connettivi  $\implies$ ,  $\wedge$ ,  $\vee$  associano a destra, perciò scrivere " $\phi \implies \psi \implies \sigma$ " equivale a dire: " $(\phi \implies (\psi \implies \sigma))$ ", mentre " $\phi \wedge \psi \vee \sigma$ " sta per " $(\phi \wedge (\psi \vee \sigma))$ ".

Anche la logica matematica si basa sulla teoria degli insiemi; questo significa che valgono i teoremi inerenti, ma anche il funzionamento delle proprietà di un insieme. Diciamo infatti

**Proposizione 2.3. Proprietà P su un insieme A**

Se abbiamo un insieme  $P$  ed è sottoinsieme proprio o improprio di un altro  $A$  abbiamo che:

$$P \subseteq A \implies \text{proprietà } P \text{ su } A.$$

Di conseguenza, se abbiamo un elemento  $a \in A$  possiamo dire che gode della proprietà se vale anche la relazione  $a \in P$ .

Finora abbiamo visto i soli elementi; ora di vedere cosa si può fare per dimostrare gli enunciati. Gli strumenti prendono la forma di **Induzione strutturale** e **Ricorsione primitiva**.

**Teorema 2.4. Induzione strutturale su PROP**

Considera una proprietà  $P \subseteq \text{PROP}$ . Supponendo veri i punti:

- $\forall \phi \in \text{AT}$ , vale  $P[\phi]$   
 Se la proprietà vale per ogni formula atomica...
- $\forall \phi, \psi \in \text{PROP}$ , valgono  $P[\phi]$ ,  $P[\psi]$   
 E che se la proprietà vale per ogni formula in PROP...

Allora varranno le relazioni  $P[\phi \wedge \psi], P[\phi \vee \psi], P[\phi \implies \psi]$ . Di conseguenza vale che  $\forall \phi \in \text{PROP}$ , vale  $P[\phi]$ .

**Esempio. Costruzione induttiva della formula  $P[\phi \implies (\phi \vee \psi)]$**

Fondamentalmente devi scomporre l'intera formula. Parti dalle formule atomiche

$$P[\phi], P[\psi]$$

Entrambe valgono, quindi puoi prendere una o l'altra. Puoi inoltre introdurre l'implicazione.

$$P[\phi] \implies P[\phi \vee \psi] := P[\phi \implies (\phi \vee \psi)]$$

E hai finito.

Parliamo ora di ricorsione. Premetto che tutte le volte in cui si usa un linguaggio definito induttivamente, è possibile descriverne gli elementi anche ricorsivamente. Sempre. Questo algoritmo genera funzioni in termini di sé stesse all'interno della propria definizione. Risulta più semplice con l'esempio del fattoriale:

**Esempio. Funzione fattoriale**

$$\text{fatt} = \begin{cases} 0! = 1 \\ n! = n \times (n-1)! \end{cases}$$

Come si può vedere, la funzione comprende ogni singolo caso che si può presentare. Se alla funzione è dato il valore 0 sei nel caso base, mentre in qualunque altra istanza sostituisci il numero dato alla  $n$  e poi svolgi i calcoli.

Terminiamo con due ultimi concetti sintattici definiti ricorsivamente:

**Definizione 2.5. Rango di una formula**

Definiamo  $r : \text{PROP} \rightarrow \mathbb{N}$  ricevente in input una formula, il valore del suo rango. Si tratta di una quantità statistica utile al fine di vedere una definizione per ricorsione.

- $r[\phi] = 0$ , se  $\phi \in \text{AT}$
- $r[(\neg\phi)] = 1 + r[\phi]$
- $r[(\phi \circ \psi)] = 1 + \max\{r[\phi], r[\psi]\}$ , con  $\circ \in \{\vee, \wedge, \implies\}$

**Definizione 2.6. Sottoformula**

Si definisce sottoformula  $\text{sub} : \text{PROP} \rightarrow 2^{\text{PROP}}$  la funzione che prende in input una formula proposizionale e ne restituisce l'insieme delle parti. Compone l'insieme di tutte le sottoformule presenti in una certa formula data.

- $\text{sub}[\phi] = \{\phi\}$ , se  $\phi \in AT$
- $\text{sub}[(\neg\phi)] = \{(\neg\phi)\} \cup \text{sub}[\phi]$
- $\text{sub}[(\phi \circ \psi)] = (\phi \circ \psi) \cup \text{sub}[\phi] \cup \text{sub}[\psi]$ , con  $\circ \in \{\wedge, \vee, \implies\}$

## 2.2 Semantica

## 2.3 Sistemi Deduttivi

## 2.4 Correttezza e Completezza

## 2.5 Esercizi

## 2.6 Appunti

### 2.6.1 Semantica

Finora abbiamo studiato il valore sintattico delle frasi, ma non dimentichiamo che, come nella lingua italiana, una composizione di frasi può avere un significato; qui nel linguaggio logico è necessario analizzare gli enunciati per determinarne eventualmente verità o falsità. Come fare?

#### Definizione 2.7. Valutazione di verità

Per comodità indichiamo il valore di verità con 1 e quello di falsità con 0; quindi sia la funzione  $V : \text{PROP} \rightarrow \{0, 1\}$ , tale che valgano le seguenti relazioni:

1.  $V[\perp] = 0$
2.  $V[\neg\psi] = 1 \iff V[\psi] = 0$
3.  $V[\psi \wedge \phi] = 1 \iff V[\psi] = 1 \wedge V[\phi] = 1$
4.  $V[\psi \vee \phi] = 1 \iff V[\psi] = 1 \vee V[\phi] = 1$
5.  $V[\psi \rightarrow \phi] = 1 \iff V[\psi] = 0 \vee V[\phi] = 1$
6.  $V[\psi \iff \phi] = 1 \iff V[\psi] = V[\phi]$

Queste relazioni sono riassumibili in tabelle di verità, molto utili in ogni caso ci si presenti davanti. Ora, considera le variabili che abbiamo usato finora; queste possono indicare composizioni di formule al loro interno, quindi non essere atomiche, si definiscono infatti **metavariabili**, e per conoscere il loro valore semantico è necessario valutare i loro simboli proposizionali. Ciò avviene grazie alla **Valutazione atomica**, la base per definire tutte le valutazioni di verità.

#### Definizione 2.8. Valutazione atomica

Sia la funzione  $V : AT \rightarrow \{1, 0\}$  tale che  $v[\perp] = 0$

Seguirà la valutazione come risultato:

$[\cdot]_v : \text{PROP} \rightarrow \{1, 0\}$  Tale che  $[\alpha]_v = v[\alpha]$  per  $\alpha \in \text{AT}$ .

Si possono presentare alcuni casi notevoli davanti alla valutazione del significato di un enunciato, come:

### Definizione 2.9. Concordanza

Dato un insieme finito di proposizioni  $P$ , se due valutazioni sono uguali, si dicono concordanti su  $P$ . Nel caso in cui una formula proposizionale è vera in ogni caso, si dice **Tautologia** e si indica con  $[\alpha]_v = 1$ .

Negli esercizi verranno utilizzate le seguenti scritture, con  $\alpha \in \text{PROP}$  generica:

- $\models \alpha$  : Tautologia.
- $\vdash \alpha$  : Teorema.

Per dimostrare una tautologia è necessario controllare che sia vera per ogni singolo caso; se in un'istanza si rivela falsa, non è tautologia.

### Teorema 2.10. Tautologie Notevoli - Inserire formule

- Legge di De Morgan
- Negazione Involutiva
- Legge Commutativa
- Legge Distributiva
- Legge Associativa

Questo concetto e ragionamento può essere esteso anche ad un insieme determinato di proposizioni, che chiameremo  $\Gamma$ . Definiamo quindi:

### Definizione 2.11. Soddisfacibilità di una formula e conseguenza logica

La valutazione di verità si estende ad un insieme di proposizioni arbitrario  $\Gamma$  tale che:  
 $[\Gamma]_v = 1 \iff \forall \alpha \in \Gamma \mid [\alpha]_v = 1^a$ .

Da questo insieme di proposizioni è possibile anche trarre delle conclusioni, dette *conseguenze logiche*. Si dice che una proposizione  $\psi$  segue logicamente da un insieme  $\Gamma$  se la verità delle proposizioni in esso contenute implica la verità di  $\psi$ ; formalmente:

Dato l'insieme  $\Gamma$  di formule, si dice  $\psi$  conseguenza logica delle formule di  $\Gamma$  e si scrive  $\Gamma \models \psi$ . Se  $[\Gamma]_v = 1 \rightarrow [\psi]_v = 1$ . Se l'insieme vuoto implica una valutazione positiva di  $\alpha$ , abbiamo una tautologia.

<sup>a</sup>Inoltre, se  $[\Gamma]_v \neq 1$ , significa che c'è almeno una valutazione dove risulta falsa, non che lo sono tutte. Se si afferma invece che  $[\Gamma]_v = 0$ , tutte le valutazioni saranno false.

Diremo infine che una formula  $\psi$  è detta *soddisfacibile* se esiste una valutazione  $v$  per cui risulti vera, altrimenti è *insoddisfacibile*. A sua volta, questo concetto si può applicare agli insiemi di proposizioni.  $\Gamma$  si dice soddisfacibile quando una sua formula  $\psi$  è sempre vera data una certa valutazione, altrimenti è insoddisfacibile.

### Esempio. Soddisfacibilità di un insieme $\Gamma$

- Se  $\{p\} \in \Gamma$ ,  $[p]_v = 1$  e quindi l'insieme è soddisfacibile.
- Se  $\{p, \neg p\} \in \Gamma$ , non tutte le valutazioni di  $p$  sono vere in quanto contraddittorie e quindi l'insieme non è soddisfacibile.

La **Sostituzione** è invece un'utile tecnica per sostituire variabili sotto un unico simbolo e rendere la scrittura più leggibile e semplice. Mettiamo di avere diverse variabili, tutte scritte in modo diverso, ma alcune di queste hanno lo stesso identico valore di altre; qui è possibile usufruire di questa tecnica.

### Definizione 2.12. Sostituzione Proposizionale

Sia la formula:  $\phi = ((p1 \rightarrow (p5 \vee p1)) \wedge p3)$ . Qui è possibile sostituire tutti i  $p1$  con  $\psi$ . Il passaggio intermedio si scrive  $\phi[\psi/p1]^a$  e risulta con:  
 $\phi = ((\psi \rightarrow (p5 \vee \psi)) \wedge p3)$ .

Se un termine nella formula presa in esame non compare affatto, è ovvio che la sostituzione non avrà alcuno scopo.

<sup>a</sup>Significa che  $\psi$  sostituirà il valore  $p1$ .

### Teorema 2.13. Proprietà della sostituzione

Siano  $\psi_1, \psi_2$  due formule e  $p \in AT$ . Se  $\models \psi_1 \iff \psi_2$ , prendendo una formula generica  $\phi$  otteniamo che:

$$\models \phi[\psi_1/p] \iff \phi[\psi_2/p].$$

Ci è quindi possibile tirar fuori dal cappello magico nuove formule, perché questa scrittura è equivalente alla prima.

Formalizziamo infine il concetto di **Relazione di equivalenza**. Molto semplicemente, i simboli  $\leftrightarrow, \iff$  Rappresentano rispettivamente equivalenza semantica e sintattica; sono una scrittura compatta del seguente concetto:  $\phi \iff \psi = (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ . Restituisce il valore di vero solamente quando entrambe le formule sono vere.<sup>1</sup>

Una relazione di equivalenza  $R$  si scrive  $\approx$  ed è tale se vengono rispettati questi tre criteri:

1. *Riflessività*;  $\forall a \in A \mid aRa$ .
2. *Transitività*;  $\forall a, b, c \in A \mid (aRb \wedge bRc) \rightarrow aRc$ .

<sup>1</sup>Ciò implica anche che se le formule sono equivalenti e la valutazione di una delle due è vera, lo sarà per forza anche l'altra.

3. *Simmetria*;  $\forall a, b \in A \mid aRb \rightarrow bRa$ .

Diremo infine che date due formule  $\phi, \psi \in \text{PROP}$ , se  $\models \phi \leftrightarrow \psi$ , si scriverà  $\phi \approx \psi$ .

## 2.6.2 Lavorare con la Logica Proposizionale

Prima di iniziare a lavorare, è necessario introdurre il concetto di **Sistema Deduttivo**. Si tratta di un costrutto che fornisce una struttura formale di regole inferenziali che permettono di derivare una conclusione a partire da alcune premesse.

Che cosa consente tutto ciò? Ora possiamo effettuare lavori di deduzione per verificare la veridicità di una formula data.

## 2.6.3 Induzione e Ricorsione

Le dimostrazioni per **induzione strutturale** e **ricorsione primitiva** sono concetti fondamentali che ti verranno sicuramente richiesti all'esame; quindi vediamoli nello specifico.

### **Teorema 2.14. Principio di Induzione strutturale sull'insieme PROP**

Considera una proprietà  $P$  da provare su  $\text{PROP}$ , quindi  $P \subseteq \text{PROP}$  supponendo veri ambo gli enunciati:

- $\forall \psi \in \text{AT}$ , vale  $P[\psi]$ .
- $\forall \psi, \phi \in \text{PROP}$ , valgono  $P[\psi], P[\phi]$

Varranno quindi le relazioni  $(\psi \wedge \phi), (\psi \vee \phi), (\psi \rightarrow \phi)$ .

Allora, noi dobbiamo provare la formula  $P$  su  $\text{PROP}$ . Supponiamo che per ogni formula atomica e per ogni elemento di  $\text{PROP}$  valga la proprietà. Ne consegue che sarà valida anche su  $(\psi \wedge \phi), (\psi \vee \phi), (\psi \rightarrow \phi)$ , comportando al fatto che  $\forall \psi \in \text{PROP}$  valga  $P[\psi]$ .

**Dimostrazione.** Basta dimostrare che  $\text{PROP} \subseteq P$ . Nota che  $P$  gode delle tre proprietà di  $\text{PROP}$  e di conseguenza, per minimalità di  $\text{PROP}$ , si ha la dimostrazione.  $\square$

Questo era un esempio teorico, vediamo un esercizio:

**Esempio.** Si dimostri per induzione strutturale la formula  $p_2 \rightarrow (p_3 \vee p_2)$  che gode della proprietà  $P$

1.  $P[p_2], P[p_3 \vee p_2]$
2.  $P[p_2 \rightarrow (p_3 \vee p_2)]$  - Finito.

Se si ha un linguaggio definito tramite induzione è sicuramente possibile utilizzare anche la ricorsione primitiva, che vedremo adesso. Se hai prestato attenzione alle lezioni di programmazione, ti sarà tutto molto più comprensibile.

La ricorsione si occupa di partire da una determinata formula per arrivare ad un caso base; vediamo un esempio messo a paragone con il linguaggio C:

**Esempio.** Cifre pari nel numero  $\alpha = 25781$

Definiamo la proprietà P, la quale ricevuto un numero  $\alpha$ , riporta quante sono le sue cifre pari.

1.  $P[\alpha] = n = P[\alpha]$ , se  $\alpha \% 2 \neq 0$ .
2.  $P[\alpha] = n = P[\alpha] + 1$ , se  $\alpha \% 2 = 0$ .
3.  $P[\alpha] = \alpha / 10$ , se  $\alpha > 10$
4.  $P[\alpha] = n$ , se  $\alpha < 10$

In C, questa funzione ricorsiva riceverebbe in input il numero 25781 e mediante l'operazione "modulo 2" è possibile capire se un numero è pari o dispari dipendentemente dal resto della divisione per 2. Se il numero è maggiore di 10, significa che ci sono altre cifre da controllare, quindi ritorna il valore del numero diviso 10 per passare alla successiva a sinistra. Se il numero è minore di 10, dopo l'ultimo controllo ritornerà il valore di n, ovvero le cifre pari di cui abbiamo bisogno.

Descrivere il caso base appropriatamente è di fondamentale importanza; poiché senza di esso, la ricorsione andrebbe avanti ad infinitum.

**Dimostrazione.** Si trovi quante cifre pari sono contenute nel numero  $\alpha = 25781$

1.  $25781 \% 2 \neq 0 \rightarrow n = 0, \alpha = \alpha / 10$
2.  $2578 \% 2 = 0 \rightarrow n = 0 + 1, \alpha = \alpha / 10$
3.  $257 \% 2 \neq 0 \rightarrow n = 1 + 0, \alpha = \alpha / 10$
4.  $25 \% 2 \neq 0 \rightarrow n = 1 + 0, \alpha = \alpha / 10$
5.  $2 \% 2 = 0 \rightarrow n = 1 + 1, \alpha = \alpha / 10$ , arrivati al caso base.

Numeri pari  $n \in \alpha = 2$

□

## 2.6.4 Tabelle di Verità

## 2.6.5 Deduzione Naturale

La **Deduzione naturale** è un metodo di ragionamento puramente sintattico. Si effettua stabilendo:

- *Assunzioni*; Ragionamenti temporanei.
- *Derivazioni*; Dalle formule è possibile effettuare derivazioni attraverso determinate leggi. Ogni passaggio si dice **Passo di derivazione**.
- *Scaricamento*; La chiusura delle assunzioni fatte all'inizio.

Tutte le derivazioni sono rappresentate graficamente tramite **alberi**, dove in cima stanno le *foglie*, ovvero le assunzioni e alla base la *radice*, che è la formula da derivare. Una derivazione è conclusa solamente quando ogni foglia è stata *chiusa*, quindi scaricata.

**Definizione 2.15. Teorema**

Una determinata formula  $\psi$  si dice teorema se esiste una derivazione per cui l'insieme delle ipotesi  $\text{Hp}[D] = \emptyset$ , ovvero quando è vuoto poiché tutte le foglie sono state chiuse. Possono inoltre esistere più derivazioni per un singolo teorema.

Passiamo ora alle **Regole deduttive**. Si tratta di algoritmi utili per la derivazione delle formule nella deduzione naturale. Si dividono in due tipi:

- *Introduzione*; Premesse collegate introducendo un connettivo logico.
- *Eliminazione*; Dalle premesse è possibile rimuovere un connettivo logico.

Negli esercizi, le regole deduttive consentono di effettuare i passi di derivazione e sono rappresentati in forma di frazione. Sopra stanno le premesse, sotto la conclusione e a fianco la regola utilizzata. Le formule all'interno di parentesi quadre sono poi ipotesi assunte/scaricate da una determinata regola di derivazione, la quale dovrà essere appropriatamente numerata. Vedremo come fare insieme alla rappresentazione delle regole a nostra disposizione:

- **Regole dell'implicazione ( $\rightarrow$ )**

- **Introduzione dell'implicazione**

Se dall'ipotesi  $\alpha$  segue  $\beta$ , con una certa derivazione  $D$ , allora si deriva  $\alpha \rightarrow \beta$  tramite l'introduzione dell'implicazione.

Questa regola assume/scarica come ipotesi l'implicante dell'implicazione, indicando quella formula tra le parentesi quadre e un indice.

$$\frac{[\beta]^1}{\alpha \rightarrow \beta} \text{I}^1$$

- **Eliminazione dell'implicazione**

- **Indebolimento**

- **Regole della congiunzione ( $\wedge$ )**

- **Introduzione dell'AND**

- **Eliminazione dell'AND**

- **Regole del bottom ( $\perp$ )**

- **Riduzione ad assurdo (RAA)**

- **Eliminazione del bottom**

- **Regole della disgiunzione ( $\vee$ )**

- **Introduzione dell'OR**

- **Eliminazione dell'OR**

- **Regole della doppia implicazione ( $\leftrightarrow$ )**

- Introduzione della doppia implicazione
- Eliminazione della doppia implicazione
- Assiomi

assiomi, prove indirette, derivabilità.

# Logica del Primo Ordine

La **Logica del Primo Ordine**, o dei Predicati è un'espansione della Logica Proposizionale. La novità principale è la presenza di due nuovi simboli detti **Quantificatori**: *Per ogni*  $\forall$  ed *Esiste*  $\exists$ , con i quali sarà possibile descrivere strutture matematiche. La loro particolarità è che legano più di ogni altro connettivo.

## 3.1 Sintassi

## 3.2 Sistemi Deduttivi

formalizzazione, contromodello, struttura di Peano

## 3.3 Semantica

## 3.4 Esercizi

## 3.5 Appunti

### 3.5.1 Strutture e Tipi di Similarità

### 3.5.2 Derivazione Semantica

### 3.5.3 Deduzione Naturale

### 3.5.4 Teoremi di Correttezza e Completezza