Logica per l'informatica - Lezione 1.5.
Teorie degli insiemi (cenni)

Andrea Malvezzi

20 Settembre, 2024

1 Introduzione

I matematici ragionano su vari elementi, alcuni concreti ed altri astratti.
Per fare ciò occorre assumere che questi elementi soddisfino certe proprietà, ragionando al contempo stesso sulle conseguenze di queste.

1.1 Esempio di proposizione non sempre valida

Un esempio significativo lo si ha con la geometria euclidea.
Prendendo uno dei suoi principali assiomi, ovvero "due rette parallele non si incontrano mai", si va ad esaminare una proposizione che non sempre risulta valida:

- Prendendo ad esempio come concretizzazione delle due rette due segni a matita paralleli disegnati su un foglio, ovviamente i due non si incontreranno mai;

- Prendendo invece come concretizzazione due persone che camminano parallelarmente sulla Terra, le due potrebbero eventualmente incontrarsi, a causa della forma del pianeta.

Per rendere tale proposizione consistente occorre quindi effettuare assunzioni a loro volta consistenti, tentando di ridurre al minimo le incongruenze nel nostro pensiero.

2 Dare fondamento a un sistema logico

Dare un fondamento a un sistema logico significa quindi, all'occorrenza, implementare un nuovo set di **enti** (concetto di cui si vuole parlare) e ipotesi partendo da un subset più piccolo che sappiamo essere consistente/coerente. Partendo dal piccolo siamo quindi in grado di implementare qualcosa di nuovo.
Un esempio di ciò è la **teoria degli insiemi**, adottata verso il 1900 come **fondamento** per la matematica.
Per poter ridurre gli enti matematici a insiemi, gli enti di partenza della teoria degli insiemi devono essere il più *"bassi"* ed *"astratti"* possibile, escludendo quindi numeri, funzioni etc... che verranno poi implementate in seguito se necessario.

3 La teoria degli insiemi e l'informatica

Si può affermare che la teoria degli insiemi sia il "*linguaggio macchina della matematica*".
Questo costituisce difatti uno strumento utile per sviluppare nuovi teoremi, implementando nuovi insiemi e fornendo così una capacità di astrazione infinita (grazie a tale fondamento si può effettivamente parlare di "infiniti numerici").
Inoltre un insieme è un ente molto flessibile, in quanto ogni cosa può essere considerata un insieme, ed un insieme può essere un recipiente per altri insiemi, rimanendo comunque **unitario** e non un sottoinsieme a sua volta.
Basti pensare a un sacchetto contenente delle biglie di 3 colori diversi: il sacchetto costituisce un insieme senza **soprainsieme**, contenente tanti altri insiemi minori (le palline di un certo colore, quelle con un certo dettaglio, etc...)
Tuttavia, così facendo si perde la capacità computazionale in cui l'informatica invece eccelle: basti pensare all'inefficienza nel rappresentare i dati.
Il numero 10, banalmente, richiede un centinaio di insiemi per essere definito e rappresentato correttamente, mentre per un calcolatore si tratta solamente di 4 bit disposti in una certa maniera.

4 La teoria naive degli insiemi

Fu il primo tentativo di descrivere la teoria degli insiemi da parte di Cantor. Essenzialmente secondo questa teoria:

- Posso formare insiemi in qualunque modo mi venga in mente.
  - es. elencando gli elementi: $A = \{1, 2, 3, ...\}$;
  - es. tramite **l'assioma di comprensione**: $B = \{X — P(X)\}$;
- Gli insiemi non devono essere **omogenei** (elementi anche non della stessa specie).
- Le ripetizioni e l'ordine non contano.
- I predicati sono l'uguaglianza e l'appartenenza.

Gli insiemi si possono quindi vedere (spesso) come delle scatole, ed è perciò corretto parlare di **appartenenza** e di **inclusione**.

- Appartenenza ($\in$): A è un elemento dell'insieme B se "aprendo" B vedo direttamente A.
  - es. di **non** appartenenza: $\{1, 2\} \notin \{1, 2, 3\}$ in quanto $\{1, 2\}$ sono contenuti in una scatola a sé (in un sottoinsieme quindi);
  - es. di appartenenza: $1 \in \{1, 2, 3\}$ in quanto 1 non è contenuto in un'altra scatola.
- inclusione ($\subset$): A è sottoinsieme di B sse aprendo le scatole vedo tutti gli elementi di A anche in B.
  - es. di **non** inclusione: $\{1, 3\} \not\subset \{\{1, 2\}, 3\}$, in quanto aprendo sia l'insieme A ($\{1, 2\}$) che quello B ($\{\{1, 2\}, 3\}$) non si vedono direttamente tutti gli elementi di A in B;
  - es. di inclusione: $\{1, 3\} \subset \{1, 2, 3\}$, in quanto "semplificando" le parentesi (non è il termine corretto, è solamente per rendere l'idea di "aprire" la scatola) si ritrova il lato *sx* in quello *dx*, quindi A in B.

4.1 Il paradosso di Russell

Il sistema Naive - ed in particolare il suo **assioma di comprensione** - vennero messi in crisi da Russell e dalla seguente affermazione:

$$\boldsymbol{X} = \{ Y — Y \notin Y \}$$

Prende in esame tutti gli insiemi X che non contengono se stessi, e si chiede se X contenga se stesso:

- se si: X contiene un insieme che contiene se stesso $\rightarrow$ la premessa viene violata;
- se no: X **non contiene un insieme che non contiene se stesso** (premessa violata) $\rightarrow$ non contiene nessuno dei suoi sottoinsiemi, i quali non contengono sè stessi;

Quindi, $X \in X$ sse $X \notin X$, da cui il titolo di "paradosso".

5 Teoria assiomatica degli insiemi

Per risolvere il Paradosso di Russell (a cui da ora ci riferiremo come *PdR*), occorre quindi eliminare l'assioma di comprensione e controllare l'uso *meta-linguistico.*
Occorre quindi sviluppare una teoria basata su:

- non definizione dei concetti di insieme, uguaglianza e appartenenza;
- uso di assiomi per affermare l'esistenza di insiemi a partire da altri insiemi;
- uso di definizioni;

5.1 Terminologia utile

- **assioma**: un'ipotesi che assumiamo: potrebbe essere vera o falsa, ma siamo interessati solo alle situazioni in cui vale;
- **definizione**: solamente un'abbreviazione, non si ipotizza nulla;

6 La teoria degli insiemi Zermelo-Fraenkel

La teoria degli insiemi ZF si basa sull'**assioma di estensionalità** e sulla definizione di **essere sottoinsieme**.

6.1 Assioma di estensionalità

Due insiemi sono uguali sse hanno gli stessi elementi.

$$\forall X, \forall Y, (X = Y \Leftrightarrow \forall z.(z \in X \Leftrightarrow z \in Y))$$

Per ogni X e Y, X e Y sono uguali sse per ogni Z, Z appartiene a X sse Z appartiene a Y.

6.2 Definizione di essere sottoinsieme

X è sottoinsieme di Y se ogni suo elemento appartiene anche in Y.

$$X \subseteq Y := \forall z, (z \in X \Rightarrow z \in Y)$$

7 Lean

Lean è un linguaggio di programmazione funzionale usato per dimostrare teoremi.
In Lean ogni simbolo logico ha delle regole di Introduzione e di Eliminazione.

7.1 Per ogni ($\forall$)

7.1.1 Regola di introduzione

Scopo: Dimostrare che $\forall X, P(X)$. Per farlo si **dimostra** su un x generico.
<u>In Lean</u>: **assume x : set.** [**dimostrazione P(X)**]
Ovvero "assumendo che x sia un set (un insieme), etc...

7.1.2 Regola di eliminazione

Scopo: da un'ipotesi o un risultato intermedio concludo che P valga per un x a mia scelta.
<u>In Lean</u>: **by NOME_IPOTESI**
we proved CONCLUSIONE (NOME_RISULTATO_INTERMEDIO)

Attenzione: non è detto che il valore che ho scelto sia utile ai fini della dimostrazione!
es.:
avendo dimostrato che $\forall x$ x è un numero pari, posso concludere che 7 sia un numero pari.

7.2 Implicazione ($\Rightarrow$)

7.2.1 Regola di introduzione

Scopo: dimostrare $P \Rightarrow Q$ su un x generico.
In Lean: **suppose P as [NOME_IPOTESI].[dimostrazione Q]**

7.2.2 Regola di eliminazione

Scopo: da un'ipotesi o un risultato intermedio, concludo che Q valga per un x a mia scelta.
In Lean: **by [NOME_IPOTESI_PQ],**
[NOME_IPOTESI_P] we proved [Q]
as [NOME_RISULTATO_INTERMEDIO]

7.3 Esempio di codice Lean

```
theorem transitivity_inclusion: ∀A B C, A ⊆ B → B ⊆ C → A ⊆ C  := by
assume A: set                              //suppongo di avere tre insiemi generici A, B, C
assume B: set
assume C: set
suppose A ⊆ B that is equivalent to ∀Z. Z ∈ A → Z ∈ B as H1
      //suppongo che A ⊆ B; espando la definizione di ⊆ (per ogni Z, se Z ∈ A → Z ∈ B), chiamandola ipotesi H1
suppose B ⊆ C that is equivalent to ∀Z. Z ∈ B → Z ∈ C as H2
      //suppongo che B ⊆ C; espando la definizione di ⊆ (per ogni Z, se Z ∈ B → Z ∈ C), chiamandola ipotesi H2
we need to prove A ⊆ C that is equivalent to ∀W. W ∈ A → W ∈ C
      //devo dimostrare che A ⊆ C, che, espandendo la definizione di ⊆, è uguale a dimostrare che per ogni W, se
       W ∈ A → W ∈ C; la chiamao ipotesi H1
assume W: set                                        //sia W un generico elemento fissato
suppose W∈ A as K1                    //suppongo che W ∈ A (chiamandola ipotesi K1)
by H1, K1 we proved W ∈ B as K2
      //usando l'ipotesi H1, posso concludere che, poiché W∈ A (ipotesi K1), allora W ∈ B; la chiamo ipotesi K2
by H2, K2 done
      //usando l'ipotesi H2, posso concludere che, poiché W∈ B (ipotesi K2), allora W ∈ C

//dopo aver ipotizzato A ⊆ B → B ⊆ C, dimostro che per qualsiasi elemento W∈ A posso concludere che W∈ C.
Per fare ciò, utilizzo le regole di eliminazione del ∀ e del → .
```

Figure 1: Dimostrazione teorema di riflessività dell'inclusione.