Manuale pratico per il corso di Logica

Maria Emilia Maietti
Dipartimento di Matematica Pura ed Applicata
University of Padova
via Trieste n. 63 - 35121 Padova, Italy
maietti@math.unipd.it

13 aprile 2012

1 A che cosa serve un corso di logica?

Nei nostri discorsi quotidiani ci è certamente capitato di usare il sostantivo “logica” o l’aggettivo “logico”. Un tipico uso è in frasi del tipo: “Ti pare logico che uno si comporti in quel modo?”, “Ma con che logica ragiona quella persona?”. Probabilmente negli studi pre-universitari avete incontrato un uso più formale di queste parole, in particolare se avete fatto dei test di “logica” oppure se avete imparato a fare l’ “analisi logica” delle frasi italiane.

In questo corso faremo un uso molto specifico della parola “logica” completamente nuovo rispetto agli usi comuni. In particolare, la parola denoterà un *sistema formale* di regole fissate per poter *dedurre* la verità di certe asserzioni, scritte come *formule*, anche a partire dall’assumere come vere certe altre assunzioni.

L’attività del dedurre è simile all’attività del programmare. Il linguaggio di una logica, o sistema formale, può essere proprio pensato come un linguaggio di programmazione ove i programmi sono le deduzioni della verità di formule apartire dalla verità di un insieme (anche vuoto) di assunzioni. In poche parole con il presente corso di logica il lettore imparerà a programmare (o calcolare) *deduzioni* per *verificare* la *correttezza* di certe affermazioni in una certa logica.

Perché un informatico dovrebbe essere interessato a studiare un linguaggio per dedurre? Prima di rispondere a questa domanda, proponiamo al lettore il seguente test di logica.

Valuta la validità delle seguenti affermazioni e rispondi alle eventuali domande:

1. “Il programma

```
y = 1;
z = 0;
while  (z ≠ x)  {
          z = z + 1
          y = y * z;
   }
```

   non termina su input $x = 0$”

corretto **sì** **no**

2. "Il programma

$$
\begin{array}{l}
y = 1; \\
z = 0; \\
\texttt{while} \quad (z \neq x) \quad \{ \\
\qquad\qquad z = z + 1 \\
\qquad\qquad y = y * z; \\
\quad \}
\end{array}
$$

calcola in y il fattoriale su input x"

corretto **sì** **no**

3. Ammesso che

"**se la radice quadrata canta alla Scala di Milano allora il tuo vicino di banco è Napoleone**"

allora è vero che

"**se il tuo vicino di banco non è Napoleone ne segue che la radice quadrata non canta alla Scala di Milano**".

corretto **sì** **no**

4. **per ogni numero naturale** n **esiste un numero naturale** m **tale che**

$$n + m = n$$

corretto **sì** **no**

chi è questo m?

5. **per ogni numero naturale** n

$$2 + n = 1 + n$$

corretto **sì** **no**

6. Ammesso che

   “ **non si dia il caso che non esista input su cui il programma X si ferma**”

   allora è vero che

   “ **il programma X si ferma su qualche input**”.

   corretto **sì** **no**

7. “ **Dio esiste**”.

   corretto **sì** **no**

8. Ammesso che “ **non tutti i programmi siano utili e corretti**” allora è vero che “ **esiste un programma non utile o esiste un programma non corretto.**”

   corretto **sì** **no**

9. “**Ogni bravo informatico può costruire un programma che attiva tutti e soli i programmi che non si attivano da sè**”

   corretto **sì** **no**

10. Supponi che le seguenti affermazioni siano valide

    - “ **Se Claudia non va in gita allora Giovanni ci va.**”
    - “ **Beppe non va in gita se e solo se ci va Giovanni.**”
    - “ **Beppe va in gita se Claudia non va in gita.**”
    - “ **Non tutti vanno in gita.**”

    allora è vero che

    - “ **Qualcuno non va in gita.**”

      corretto **sì** **no**

    - “ **Se Giovanni non va in gita allora Beppe ci va.**”

      corretto **sì** **no**

    - “ **Se Claudia non va in gita allora Beppe non ci va.**”

      corretto **sì** **no**

- “ **Claudia va in gita.**”

  corretto **sì** **no**

- “ **Non si dà il caso che nessuno vada in gita.**”

  corretto **sì** **no**

Pensate che un robot, ovvero un computer (ovvero un programma) sarebbe in grado di risolvere il test di logica che avete appena eseguito in modo corretto e automatico (o semiautomatico/interattivo con il vostro aiuto)?

Il presente corso di logica pone le basi affinchè ciò possa realizzarsi, ovvero pone le basi per l'*intelligenza artificiale*.

Ma, allora, come si istruisce un robot a rispondere al test di logica?

L'idea di base è di istruirlo costruendo un *linguaggio formale* in cui poter *codificare* le asserzioni e provarne la *verità* tramite *derivazioni/deduzioni formali* che rappresentano la spiegazione *logica* del perchè certe asserzioni sono vere.

Anticipiamo qualche esempio di quel che faremo nel corso.

1.1 Esempio di codifica in linguaggio formale.

L'asserzione

Ammesso che “ **non si dia il caso che non esista input su cui il programma X si ferma**” allora è vero che “ **il programma X si ferma su qualche input**”.

si potrà formalizzare in tal modo nel sistema formale che useremo

$$\boxed{\neg\neg(\exists x\ F(x)) \vdash \exists x\ F(x)}$$

posto che:

$\mathbf{F(x)} \stackrel{def}{=}$ “ **il programma si ferma sull'input x**”

1.2 Esempio di derivazione formale

Una *derivazione* formale è un albero del tipo

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\textbf{ax-id}\\ B(x) \vdash B(x)\end{array} \qquad \dfrac{\begin{array}{c}\textbf{ax-id}\\ C(x) \vdash C(x)\end{array}}{C(x) \vdash \exists y\, C(y)}\ \exists\text{−re}}{B(x) \to C(x), B(x) \vdash \exists y\, C(y)}\ \to\text{−re}}{B(x) \to C(x), \forall x\, B(x) \vdash \exists y\, C(y)}\ \forall\text{−re}}{B(x) \to C(x) \vdash \forall x\, B(x) \ \ \to \ \ \exists y\, C(y)}\ \to\text{−F}}{\exists x(\, B(x) \to C(x)\,) \vdash \forall x\, B(x) \ \ \to \ \ \exists y\, C(y)}\ \exists\text{−F}$$

1.3 Scopi del corso

Gli scopi principali del corso sono

- imparare a *codificare asserzioni* in *linguaggio formale*
- imparare a *dedurre asserzioni* come *conseguenza logica* di altre *asserzioni* (tramite *alberi di derivazione*)

Ovvero il corso intende essere un'introduzione alla *deduzione formale/automatica.* Oltre a pensare all'arte del dedurre in modo analogico come un particolare tipo di *programmazione*, il nostro corso ha anche un aggancio concreto e specifico con il mondo della programmazione. Infatti lo studio della deduzione logico-formale pone le basi per la *verifica formale dei programmi*, ovvero per *formalizzare* la *correttezza (parziale)* di *programmi* rispetto ad una specifica, in modo *automatico/semiautomatico.*

Consideriamo il seguente programma ad esempio

```
y = 1;
z = 0;
while  (z ≠ x)  {
          z  =  z + 1
          y  =  y * z;
  }
```

Questo programma calcola in y il fattoriale di x.

Per verificare questo programma, o qualsiasi altro, in letteratura esiste un calcolo che mostriamo di seguito solo per dare l'idea della parte che noi tratteremo nel corso:

Esempio di calcolo *formale* per correttezza programmi

$$\frac{(\!|\phi|\!)\ C_1\ (\!|\eta|\!) \qquad (\!|\eta|\!)\ C_2\ (\!|\psi|\!)}{(\!|\phi|\!)\ C_1; C_2\ (\!|\psi|\!)}\ \text{Composition}$$

$$\frac{}{(\!|\psi[E/x]|\!)\ x = E\ (\!|\psi|\!)}\ \text{Assignment}$$

$$\frac{(\!|\phi \wedge B|\!)\ C_1\ (\!|\psi|\!) \qquad (\!|\phi \wedge \neg B|\!)\ C_2\ (\!|\psi|\!)}{(\!|\phi|\!)\ \texttt{if}\ B\ \{C_1\}\ \texttt{else}\ \{C_2\}\ (\!|\psi|\!)}\ \text{If-statement}$$

$$\frac{(\!|\psi \wedge B|\!)\ C\ (\!|\psi|\!)}{(\!|\psi|\!)\ \texttt{while}\ B\ \{C\}\ (\!|\psi \wedge \neg B|\!)}\ \text{Partial-while}$$

$$\frac{\vdash_{\mathrm{AR}} \phi' \to \phi \qquad (\!|\phi|\!)\ C\ (\!|\psi|\!) \qquad \vdash_{\mathrm{AR}} \psi \to \psi'}{(\!|\phi'|\!)\ C\ (\!|\psi'|\!)}\ \text{Implied}$$

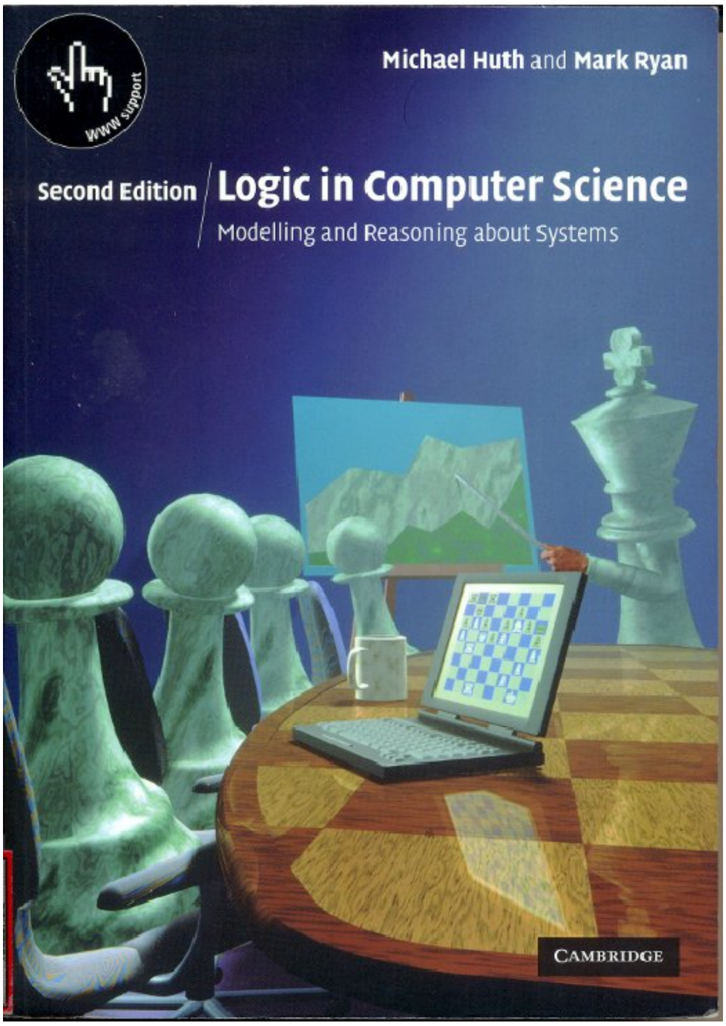
Figure 4.1. Proof rules for partial correctness of Hoare triples.

IMPARERETE A DERIVARE FORMALMENTE QUESTE PARTI LOGICHE NEL CORSO

Esempio di verifica *formale* del programma fattoriale

$$
\dfrac{
\dfrac{
\dfrac{(\!|1=1|\!)\ \texttt{y = 1}\ (\!|y=1|\!)}{(\!|\top|\!)\ \texttt{y = 1}\ (\!|y=1|\!)}\,i
\quad
\dfrac{(\!|y=1\wedge 0=0|\!)\ \texttt{z = 0}\ (\!|y=1\wedge z=0|\!)}{(\!|y=1|\!)\ \texttt{z = 0}\ (\!|y=1\wedge z=0|\!)}\,i
}{(\!|\top|\!)\ \texttt{y = 1; z = 0}\ (\!|y=1\wedge z=0|\!)}\,c
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{(\!|y\cdot(z+1)=(z+1)!|\!)\ \texttt{z = z+1}\ (\!|y\cdot z=z!|\!)}{(\!|y=z!\wedge z\neq x|\!)\ \texttt{z = z+1}\ (\!|y\cdot z=z!|\!)}\,i
\quad
(\!|y\cdot z=z!|\!)\ \texttt{y = y*z}\ (\!|y=z!|\!)
}{(\!|y=z!\wedge z\neq x|\!)\ \texttt{z = z+1; y = y*z}\ (\!|y=z!|\!)}\,c
}{(\!|y=z!|\!)\ \texttt{while (z != x) \{z = z+1; y = y*z\}}\ (\!|y=z!\wedge z=x|\!)}\,w
}{(\!|y=1\wedge z=0|\!)\ \texttt{while (z != x) \{z = z+1; y = y*z\}}\ (\!|y=x!|\!)}\,i
}{(\!|\top|\!)\ \texttt{y = 1; z = 0; while (z != x) \{z = z+1; y = y*z\}}\ (\!|y=x!|\!)}\,c
$$

Riferimento bibliografico per la verifica dei programmi



1.4 Cosa imparerà il lettore nel corso?

Imparerà il concetto di **verità formale** (o **validità formale**), la quale può essere:

- una **verità** *relativa* ad una **logica**, per cui si parla di **verità logica**.

Infatti esistono **molte logiche**. Per esempio nella letteratura corrente si trovano le seguenti logiche: **logica classica, logica intuzionista, logica quantistica, logica lineare, logica modale, logica temporale.....**

Noi, in questo corso, studieremo solo la **logica classica** e le sue teorie.

- una **verità** *relativa* ad una **teoria**, per cui si parla di **verità extra-logica** della teoria bla bla ....

  dove

teoria = logica + assiomi

  ovvero una teoria è un'estensione di una logica con assiomi.

  Esistono molte **teorie** in ogni campo del sapere. Per esempio: **teoria della computabilità (in informatica), teoria dell'aritmetica (in matematica), teoria relativistica (in fisica), teoria dell'evoluzione (in biologia), ...**

1.4.1 Esempi di verità logiche

Ad esempio l'asserzione complessa

> Ammesso che
> " **non si dia il caso che non esista input su cui il programma X si ferma**"
> allora è vero che
> " **il programma X si ferma su qualche input**".

è corretta in logica classica (ma non in logiche alternative come quella intuizionista. Per approfondimenti si veda il libro di G. Sambin "**Per Istruire Un Robot**: **(ovvero, come costruirsi una logica)**".

Altro esempio è l'asserzione

> Ammesso che
> " **se la radice quadrata canta alla Scala di Milano allora il tuo vicino di banco è Napoleone** "
> allora è vero che
> "**se il tuo vicino di banco non è Napoleone ne segue che la radice quadrata**
> **non canta alla Scala di Milano**".

che è corretta formalmente ma senza significato perchè la proposizione "la radice quadrata canta alla Scala di Milano" non ha senso semanticamente (a meno di ulteriori specifiche come il fatto che "radice quadrata" è un nome per un cantante o gruppo di cantanti, ma allora dovrebbe andare con la maiuscola "Radice quadrata"..).

1.4.2 Esempi di verità extra-logiche

L'asserzione

per ogni numero naturale n **esiste un numero naturale** m **tale che** $n + m = n$

è corretta nella teoria dell'aritmetica, ma non è verità logica.

L'asserzione

"**Dio esiste**"

è corretta nella teoria della dottrina cristiana ma non è una verità logica.

1.4.3 Procedure formali del corso

Nel corso verrà fornita una **procedura di verifica formale** per classificare le **asserzioni formalizzate** in **logica classica** come:

- **verità logiche**:
  per es. esercizio 3. o 6. o 8. del test
- **falsità logiche, ovvero paradossi logici**:
  per es. esercizio 9. del test
- **opinioni logiche, cioé asserzioni nè vere, nè false per la logica**:
  per es. esercizio 7. del test

1.4.4 Verifica formale delle verità dell'aritmetica di Peano

Nel corso imparerete anche a verificare *formalmente* le **verità** della **teoria dell'aritmetica di Peano** come ad esempio che vale

$$\mathbf{5 \cdot 1 = 5 \cdot 0 + 5}$$

con un albero del tipo

$$\dfrac{\vdash \mathbf{Ax\ 6.} \qquad \dfrac{\dfrac{\begin{array}{c}\mathbf{ax\text{-}id}\\ 5\cdot 1 = 5\cdot 0+5 \vdash 5\cdot 1 = 5\cdot 0+5\end{array}}{\forall y\ (5\cdot s(y) = 5\cdot y+5) \vdash 5\cdot 1=5\cdot 0+5}\ \forall\text{−re}}{\forall x\ \forall y\ (x\cdot s(y)=x\cdot y+x)\vdash 5\cdot 1=5\cdot 0+5}\ \forall\text{−re}}{\vdash 5\cdot 1=5\cdot 0+5}\ \mathrm{comp}_{sx}$$

Si noterà che $\mathbf{5 \cdot 1 = 5 \cdot 0 + 5}$ NON è una verità della logica classica (servono infatti gli assiomi dell'aritmetica di Peano per derivarlo).

1.4.5 Esempio di paradosso (o contraddizione) con stessa forma esercizio 9. del test

L'asserzione dell'esercizio 9. del test precedente ha la stessa forma logica del famoso Paradosso di Russel:

"**Nel villaggio di Cantù tutti c'è un unico barbiere che rade solo gli uomini che non si radono da sé**."

Si vede che questa affermazione è un paradosso in quanto afferma una contraddizione: l'**esistenza** di tale barbiere. Questo barbiere, infatti, deve radersi ma non può farlo.

il barbiere di Cantù rade se stesso
sse
non si rade da sè

⇒ l'**esistenza** di un tal barbiere porta ad una **CONTRADDIZIONE**

e analogamente

l'esistenza del programma dell'esercizio 9. del test
che **attiva se stesso**
sse
non si attiva da sè

porta ad una **contraddizione** con lo stesso tipo logico di ragionamento (o meglio *forma logica*) del paradosso del barbiere.

Ora il lettore provi ad inventare paradossi con la stessa forma... Durante avrà modo di studiare la forma logica precisa di questi paradossi.

1.5 Utilità dello studio della logica per lo sviluppo delle scienze

Diamo di seguito delle motivazioni sull'utilità dello studio della logica per uno scienziato o per un amante della conoscenza.

1.5.1 Una possibile obiezione allo studio di un corso di logica

Una possibile obiezione allo studio della logica è il suo carattare astratto e lontano dai saperi scientifici. Una prova a supporto di ciò può apparire il fatto che non compare come materia di studio negli insegnamenti pre-universitari.

Uno quindi potrebbe può dire: "**a me interessano solo le verità di alcune teorie scientifiche particolari**" come

- la **teoria dei sistemi operativi e delle reti** (in informatica)
- la **teoria delle equazioni differenziali** (in matematica)
- la **teoria delle stringhe** (in fisica)
- la **teoria della dottrina cristiana** (in teologia)
- etc....

e "mi interessano eventuali applicazioni di queste teorie all'intelligenza artificiale"!

Ma allora: *ha senso studiare logica pura oltreché teorie?* Sì per questi motivi

- I **paradossi logici** sono **falsi** in **ogni** teoria scientifica.
- le scienze devono dare per scontato le **VERITÀ LOGICHE**.

In altre parole nel momento in cui uno scienziato studia una certa teoria è bene che abbia presente le verità logiche e i paradossi in modo da non perdere tempo a verificarli, in quanto le verita' logiche sono valide a priori (e basta la logica per riconoscerle) e i paradossi sono falsi a priori. Quindi, ha senso che uno scienziato verifichi la validità di asserzioni tramite **DEDUZIONI LOGICHE** all'interno della sua teoria solo se queste asserzioni rientrano tra le **opinioni logiche** (quindi nè verità logiche, nè paradossi).

Ad esempio, non ha senso che un informatico provi a **costruire un programma che attiva tutti e soli i programmi che non si attivano da sè**.

Concludiamo dicendo che il corso di logica che tratteremo dà pure un contributo scientifico concreto e specifico alla scienza dell'informazione perchè *offre le basi per rendere la logica applicabile all'intelligenza artificiale.*

1.6 Come affrontare l'esame di logica

È *indispensabile* fare molti esercizi, poichè l'esame si basa soprattutto sul **ragionamento** e non sulla **memoria**.

1.6.1 Difficoltà del corso:

Lo studio della logica è molto astratto, più della matematica ....

In **logica**, come in **matematica**
non si sa **di cosa si parla**
nè se **ciò di cui si parla** sia **vero**.
(Russell)