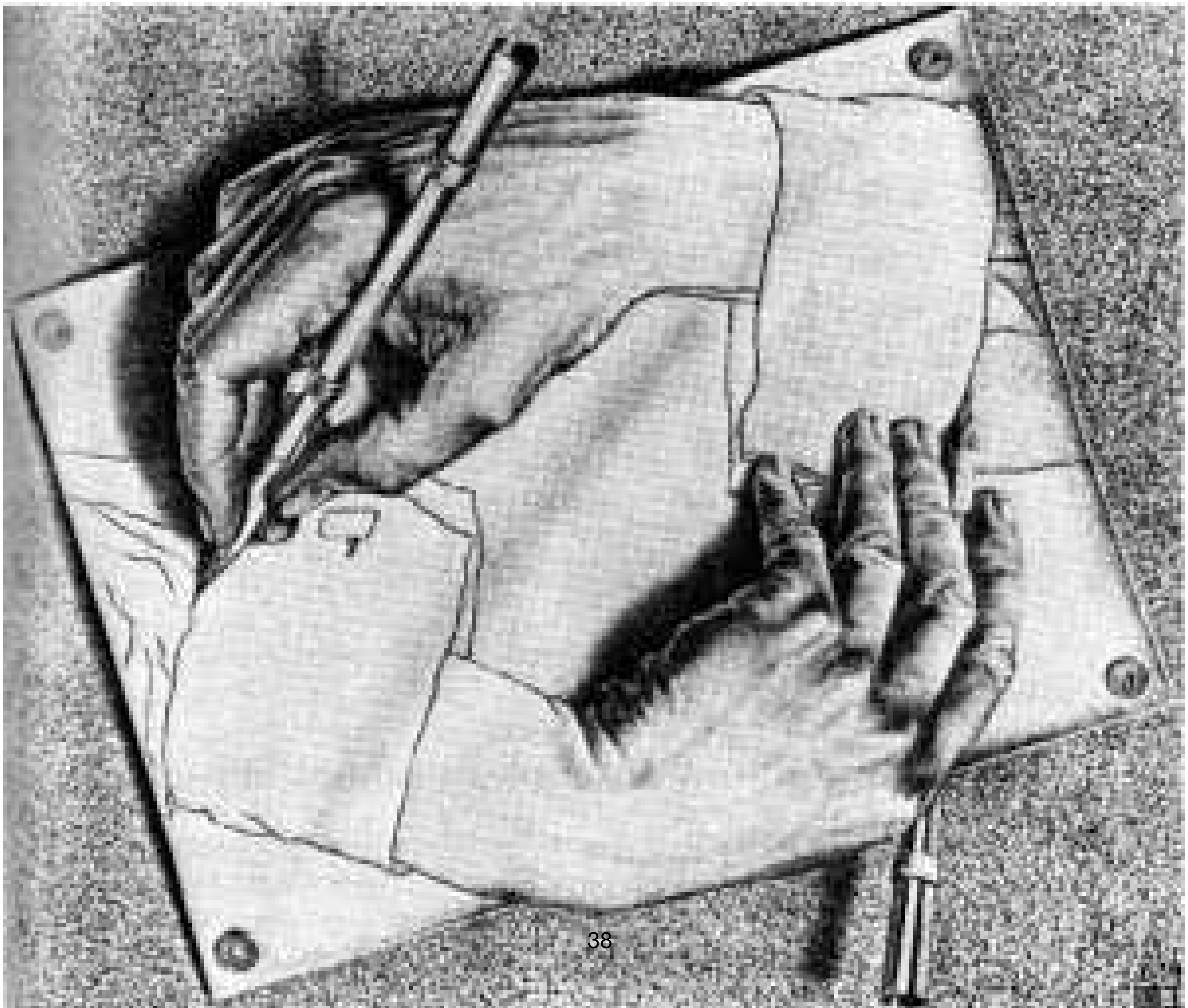
“Mani che disegnano” di Escher (1948)



2. Lezione Corso di Logica 2021/2022

1 ottobre 2021

Maria Emilia Maietti

email: maietti@math.unipd.it

regola delle tre C

studiare

con

Costanza

Concentrati

Cooperando con altri

Perchè studiare questo corso di logica?

logica = base per INTELLIGENZA ARTIFICIALE

Obiettivo di questo corso

offrire **STRUMENTI TEORICI**

per **istruire un robot a rispondere al test di logica**

Come istruire un robot a rispondere al test di logica?

linguaggio formale	= linguaggio di programmazione
derivazione formale	= programma

Domanda del Test di logica

Ammesso che

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

allora

“il programma X si ferma su qualche input”.

corretto sì no

Esempio di formalizzazione 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 puó formalizzare cosí

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

posto che:

F(x) = "il programma si ferma sull'input x"

Esempio di derivazione formale dell'asserzione formalizzata

derivazione = albero del tipo

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{\texttt{ax-id}}{B(x) \vdash B(x)}
      \qquad
      \dfrac{\dfrac{\texttt{ax-id}}{C(x) \vdash C(x)}}{C(x) \vdash \exists y\, C(y)}\ \exists\text{--re}
    }{B(x) \to C(x), B(x) \vdash \exists y\, C(y)}\ \to\text{--re}
  }{B(x) \to C(x), \forall x\, B(x) \vdash \exists y\, C(y)}\ \forall\text{--re}
}{
  \dfrac{B(x) \to C(x) \vdash \forall x\, B(x) \;\to\; \exists y\, C(y)}{\exists x(\, B(x) \to C(x)\,) \vdash \forall x\, B(x) \;\to\; \exists y\, C(y)}\ \exists\text{--F}
}\ \to\text{--F}
$$

Esempio di DEDUZIONE di verità aritmetica

il corso offre base teorica per

dedurre formalmente verità della teoria dell'aritmetica come ad esempio che vale

$$5 \cdot 1 = 5 \cdot 0 + 5$$

con un albero del tipo

$$\dfrac{\vdash \texttt{Ax 6.} \quad \dfrac{\dfrac{\dfrac{\texttt{ax-id}}{5 \cdot 1 = 5 \cdot 0 + 5 \vdash 5 \cdot 1 = 5 \cdot 0 + 5}}{\forall y \ \ (\, 5 \cdot s(y) = 5 \cdot y + 5\,) \vdash 5 \cdot 1 = 5 \cdot 0 + 5} \ \forall\!-\!\mathrm{S_v}}{\forall x \ \ \forall y \ \ (\, x \cdot s(y) = x \cdot y + x\,) \vdash 5 \cdot 1 = 5 \cdot 0 + 5} \ \forall\!-\!\mathrm{S_v}}{\vdash 5 \cdot 1 = 5 \cdot 0 + 5} \ \mathrm{comp}_{sx}$$

Obiettivi CONCETTUALI del nostro corso di logica

IMPARARE a DISTINGUERE tra

VERITÀ LOGICA	OPINIONE	CONTRADDIZIONE
(= tautologia)		(= paradosso)

via

PROCEDURE AUTOMATICHE

(robotizzabili)

o

PROCEDURE SEMI-AUTOMATICHE

(robotizzabili con interazione umana)

Obiettivi CONCETTUALI del nostro corso di logica

MPARARE a DISTINGUERE VERITÀ:	
relative ad un LOGICA (=verità logica)	relative ad una TEORIA (=verità teorica)

ove s'intende $\boxed{\text{TEORIA} = \text{LOGICA} + \text{assiomi}}$

classificare i quesiti logici

Proviamo a classificare

i quesiti del test di logica!

Che tipo di affermazione è

“se il tuo vicino di banco non è Napoleone ne segue che la radice quadrata non canta alla Scala di Milano”

ammesso che

“se la radice quadrata canta alla Scala di Milano allora il tuo vicino di banco è Napoleone”.

??

Che tipo di affermazione è

“se il tuo vicino di banco non è Napoleone ne segue che la radice quadrata non canta alla Scala di Milano”

ammesso che

“se la radice quadrata canta alla Scala di Milano allora il tuo vicino di banco è Napoleone”.

??

è verità logica o tautologia.

Che tipo di affermazione è

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

$$n + m = n + 1$$

??

Che tipo di affermazione è

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

$$n + m = n + 1$$

??

è verità aritmetica e NON logica

Che tipo di affermazione è

6. “Esistono bravi informatici che riescono a costruire un programma che attiva tutti e soli i programmi che non si attivano da sè.”

??

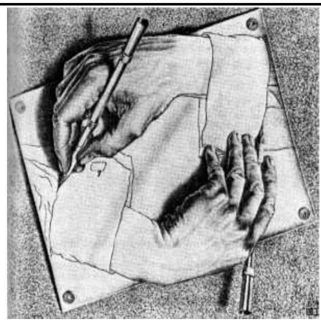
Che tipo di affermazione è

6. “Esistono bravi informatici che riescono a costruire un programma che attiva tutti e soli i programmi che non si attivano da sè.”

??

è una falsità logica

falsità logica è chiamata **PARADOSSO** o **CONTRADDIZIONE**



esiste una VERITÀ SCIENTIFICA ASSOLUTA?

NO, é relativa ad una teoria!!

e OVVIAMENTE ci sono 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), ...

ove s'intende $\boxed{\text{TEORIA} = \text{LOGICA} + \text{assiomi}}$

esempio di teoria

Supponi che le seguenti affermazioni siano valide

- "Se Carla 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 Carla 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 Carla non va in gita allora Beppe non ci va.”

corretto sì no

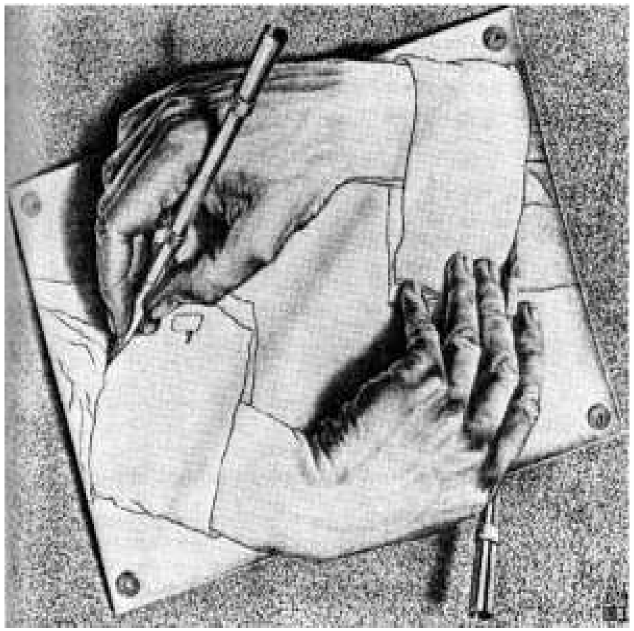
- “Carla va in gita.”

corretto sì no

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

corretto sì no

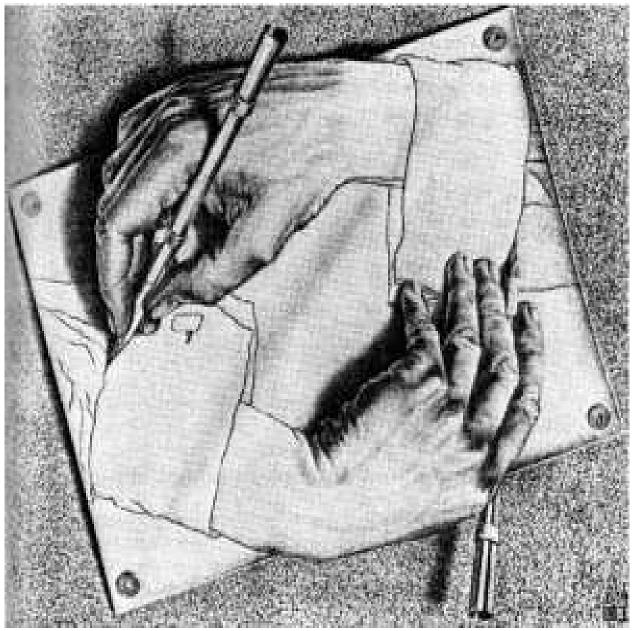
UTILITÀ PARADOSSI



servono per capire

CIÒ che NON si può affermare

UTILITÀ PARADOSSI



servono per capire

i PROGRAMMI che NON si possono costruire

Superiorità UOMO su MACCHINA

ad es. la falsità del paradosso

"Esistono bravi informatici che possono costruire un programma **ATT**

che attiva tutti e soli i programmi che non si attivano da sè"

dice che

"NON si può costruire

il programma **ATT**

rispetto alla nozione di MACCHINA attuale"

spiegazione paradosso dell'informatico

Se, il bravo informatico esistente *riuscisse a costruire un programma, che chiamiamo* **ATT**, *che attiva tutti e soli i programmi che non si attivano da sè*

allora da quanto assunto ne seguirebbe che

per ogni programma, che indichiamo con la lettera **P**, il programma **ATT**, per come è costruito, verifica la proprietà

proprietà (+++)	il programma **ATT** attiva **P** sse **P** **NON** attiva **P**

Ora, dato che **ATT** stesso è un programma, nella proprietà (+++) possiamo sostituire al posto di **P** il programma **ATT** stesso

ottenendo che vale per **ATT** la proprietà

proprietà (***)	il programma **ATT** attiva **ATT** sse **ATT** **NON** attiva **ATT**

Poi osserviamo che si possono verificare solo due casi:

o **ATT** *attiva* **ATT** oppure **ATT** **NON** *attiva* **ATT**.

Caso (1): si verifica che **ATT** *attiva* **ATT**.

In tal caso dalla proprietà (***) segue che pure **ATT NON** *attiva* **ATT** ovvero vale che

ATT *attiva* **ATT** e **ATT NON** *attiva* **ATT**

che è una contraddizione!! Quindi questo caso non può verificarsi.

Caso (2): si verifica che **ATT NON** *attiva* **ATT**.

Pure in tal caso dalla proprietà (***) segue che vale anche che **ATT** *attiva* **ATT** ovvero vale che

ATT NON *attiva* **ATT** e **ATT** *attiva* **ATT**

che è una contraddizione!!

Quindi anche questo caso non può verificarsi.

Conclusione:

siccome NON si possono verificare nè il caso (1) e nè il caso (2) che sono gli UNICI casi possibili riguardo all'esistenza di **ATT**- che per costituzione deve soddisfare la proprietà (***) - allora

NON possiamo costruire *il programma* **ATT**

in quanto **NON** *può esistere*!!!