I giochi logici di Raymond Smullyan

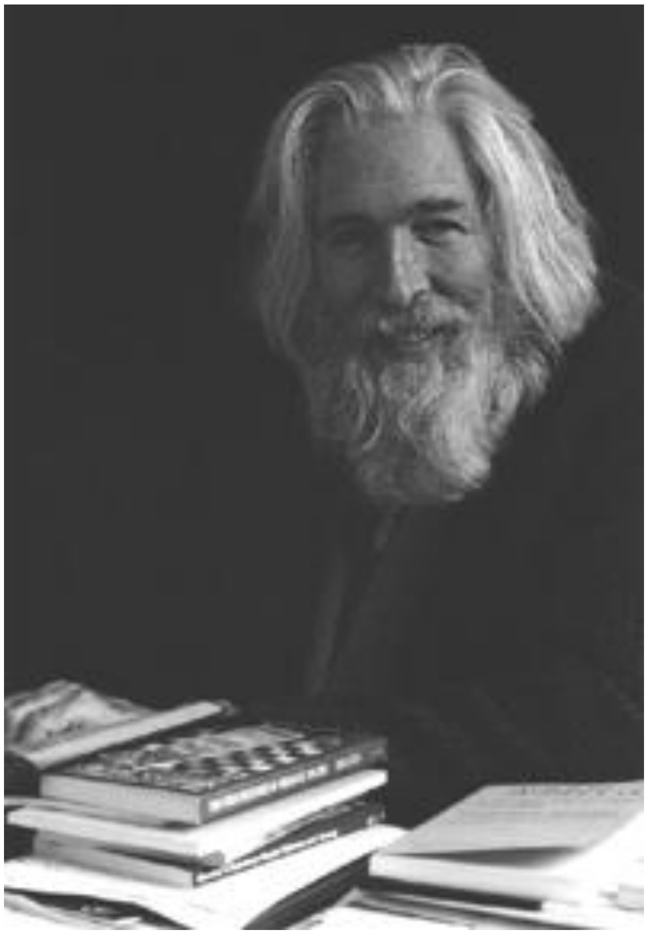
tra nozioni di base e limiti dell'informatica

Lorenzo Repetto
I.I.S. "Italo Calvino", Genova
lorenzo.repetto@istruzione.it

Circolo Filologico Milanese – Sezione Tecnica & Informatica

Milano, 17 marzo 2018

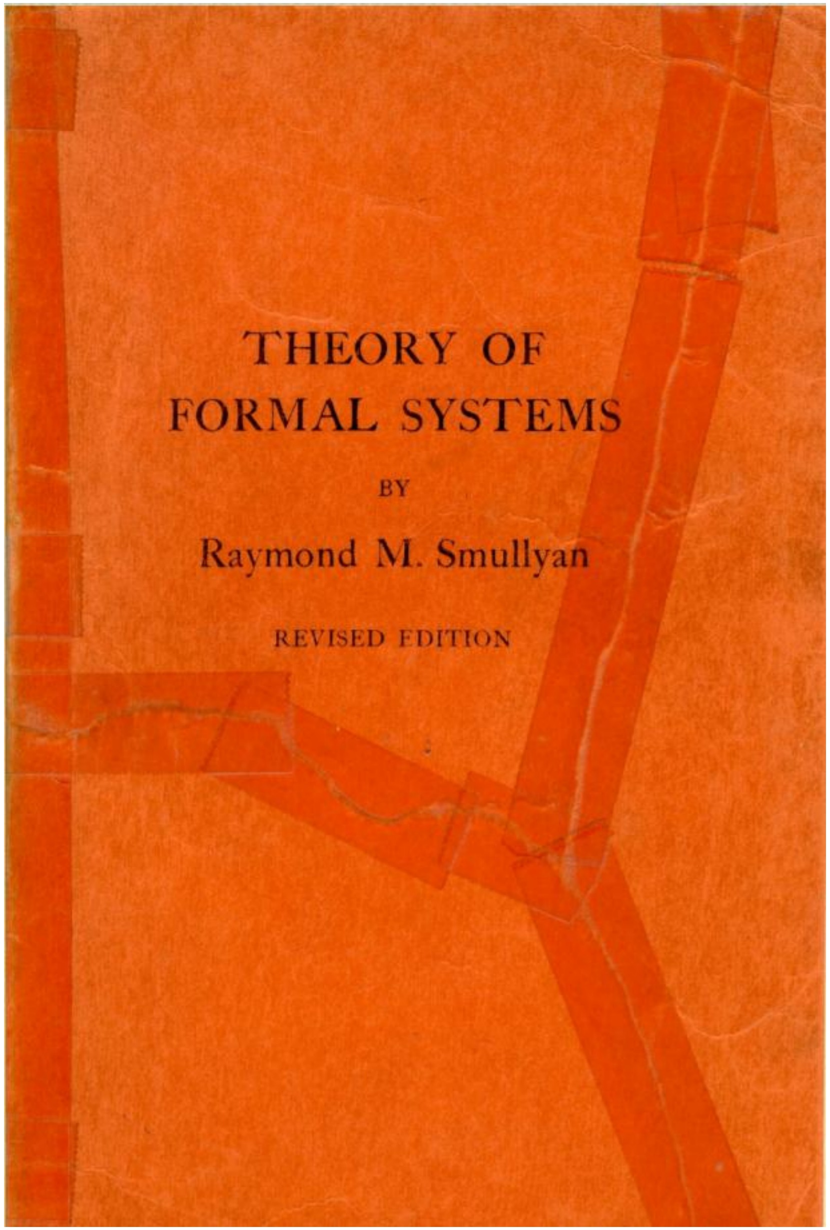
"Some people are always critical of vague statements. I tend rather to be critical of precise statements; they are the only ones which can correctly be labeled wrong."

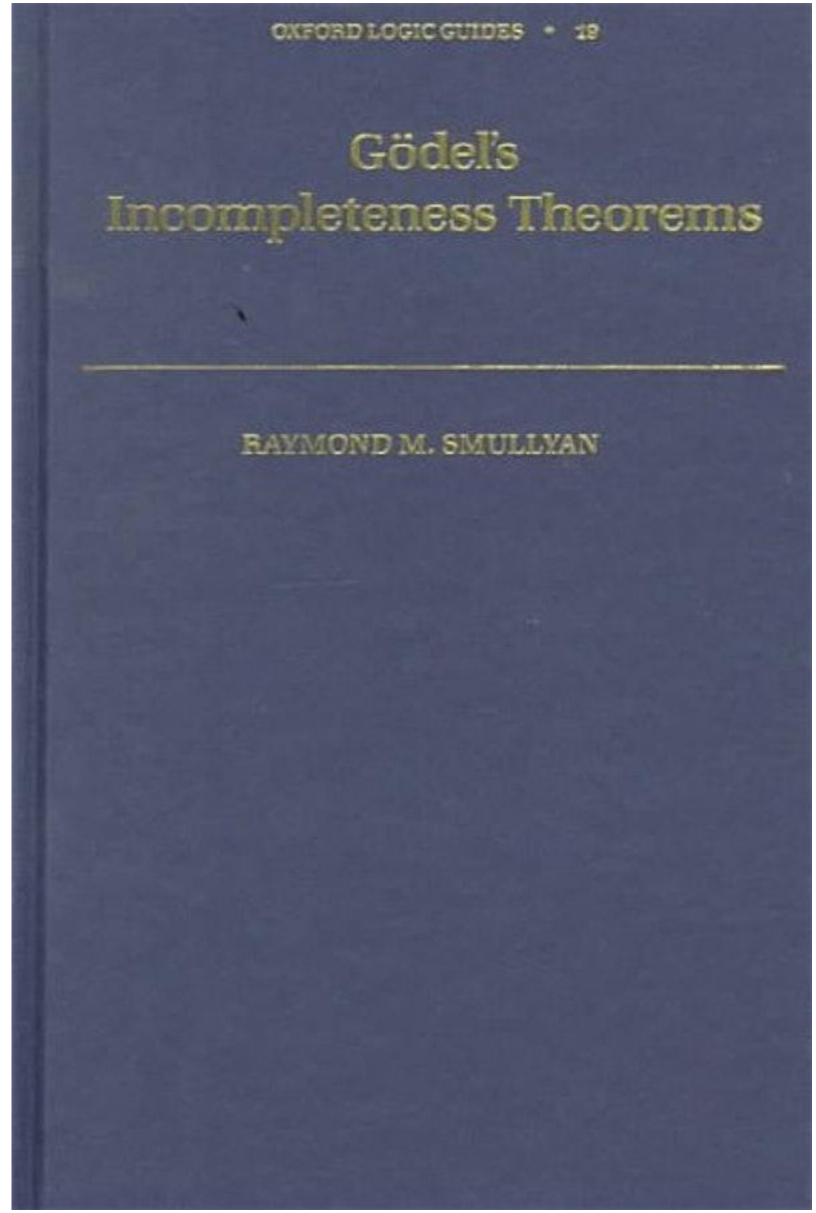


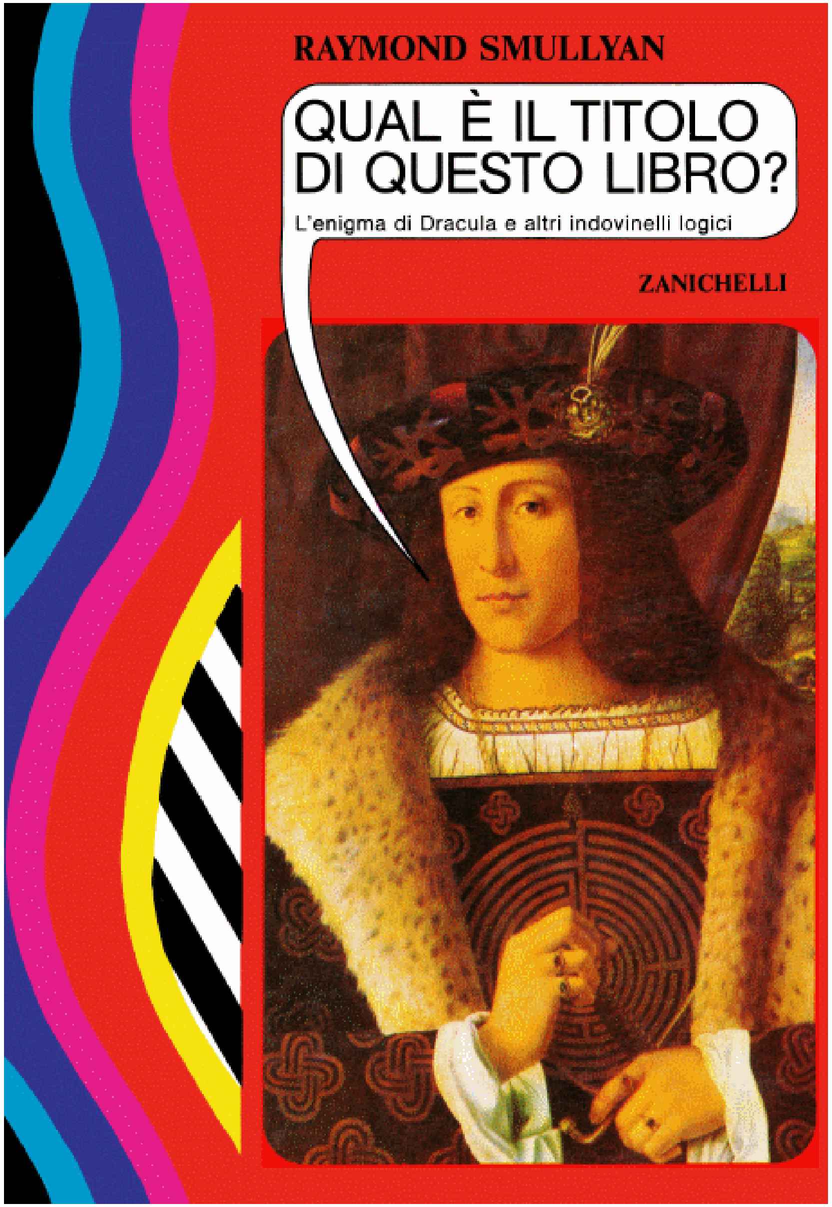
"Superstition brings bad luck."

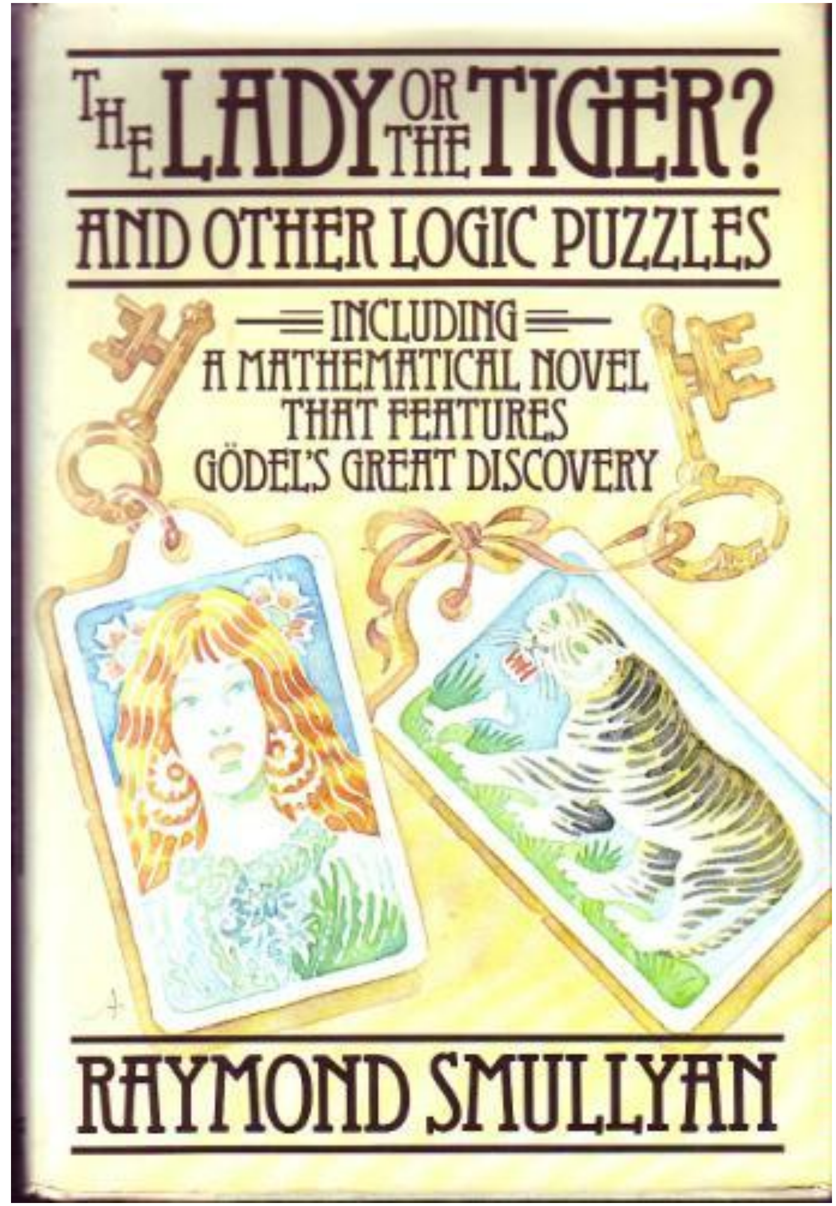
Raymond Merrill Smullyan
(1919 – 2017)

IN LIFE HE WAS INCORRIGIBLE.
IN DEATH, HE'S EVEN WORSE!


THEORY OF
FORMAL SYSTEMS
BY
Raymond M. Smullyan
REVISED EDITION

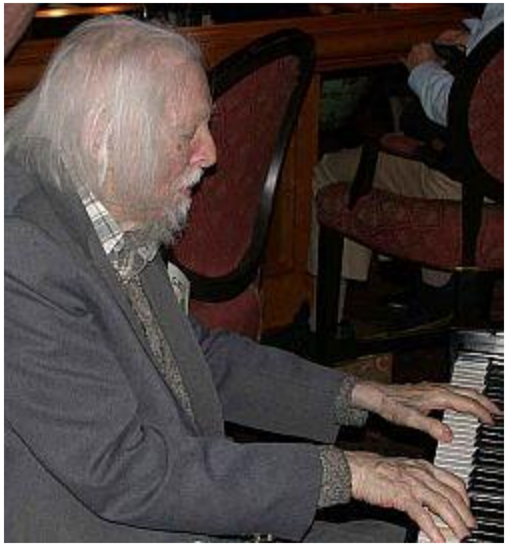

OXFORD LOGIC GUIDES • 19
Gödel's
Incompleteness Theorems
RAYMOND M. SMULLYAN

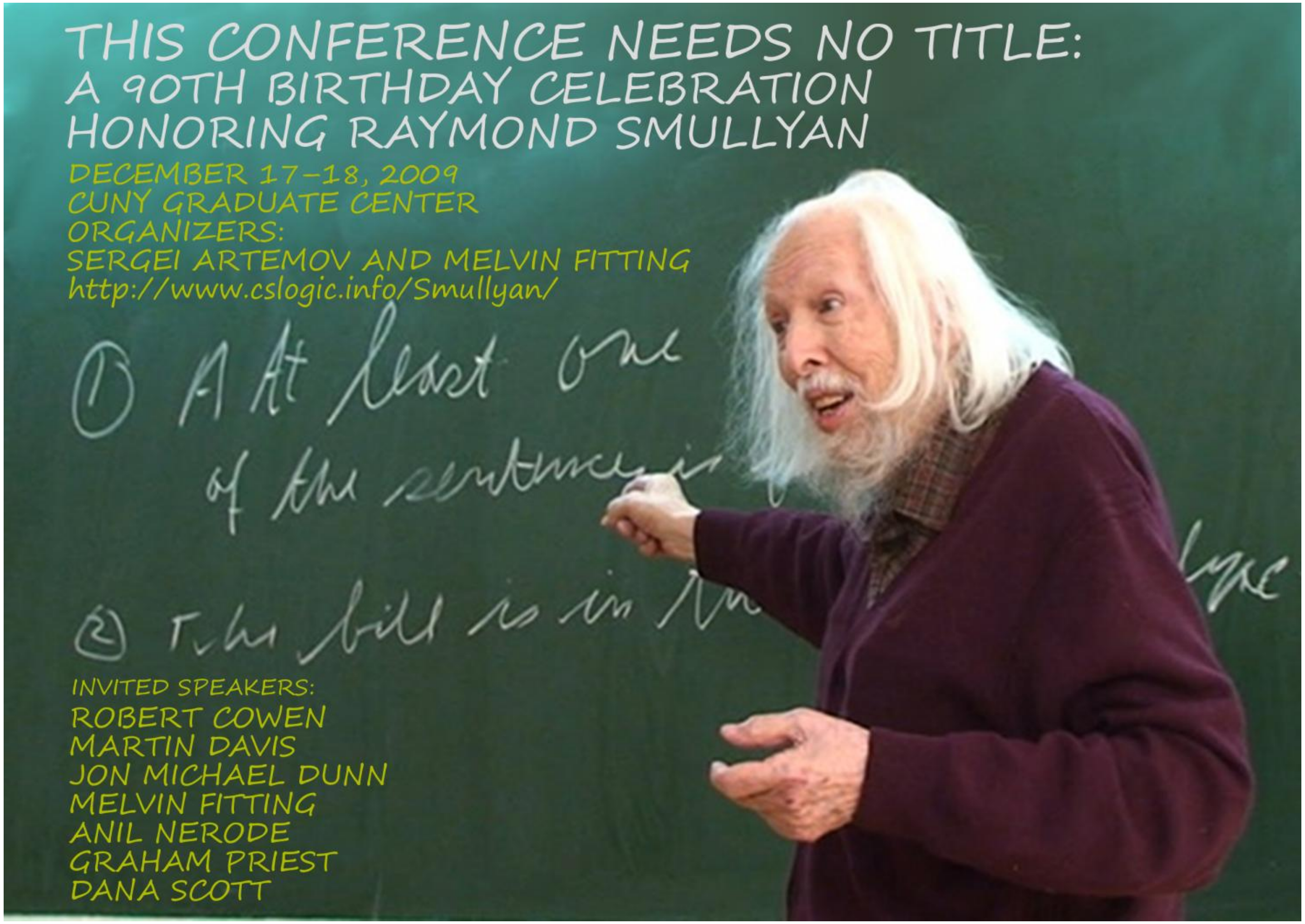

RAYMOND SMULLYAN
QUAL È IL TITOLO DI QUESTO LIBRO?
L'enigma di Dracula e altri indovinelli logici
ZANICHELLI


THE LADY OR THE TIGER?
AND OTHER LOGIC PUZZLES
INCLUDING A MATHEMATICAL NOVEL THAT FEATURES GÖDEL'S GREAT DISCOVERY
RAYMOND SMULLYAN

Con la sua futura moglie, Blanche …

- Ray: «Io farò un'affermazione. Se è vera, allora tu mi farai un autografo, altrimenti no.»
- Blanche: «D'accordo!»
- Ray: «Non mi darai né il tuo autografo, né un bacio.»





"I now introduce Professor Smullyan, who will prove to you that either he doesn't exist or you don't exist, but you won't know which."

(Melvin Fitting)

Una curiosa macchina numerica

- usa le cifre decimali 1, 2 e 3
- accetta in input una qualsiasi sequenza di cifre purché

- inizi con quante cifre 3 si vogliano, anche nessuna

- poi vi sia una cifra 2

- seguita da un numero arbitrario di cifre, almeno una

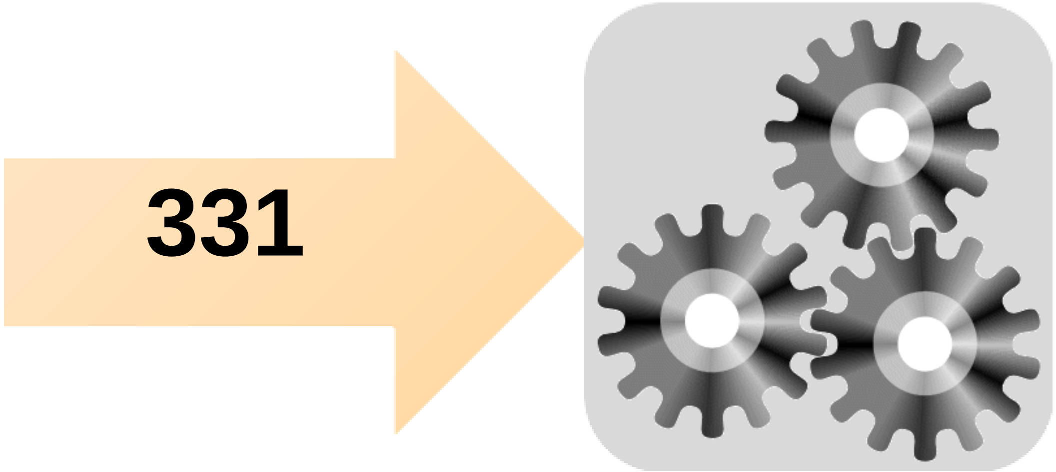
- esempi di sequenze accettate
  (= forniscono come risultato una sequenza non vuota)

213	222	321	3321
3323	3223	3232	32323

Come funziona?

- se riceve una sequenza che non è della forma accettata, allora

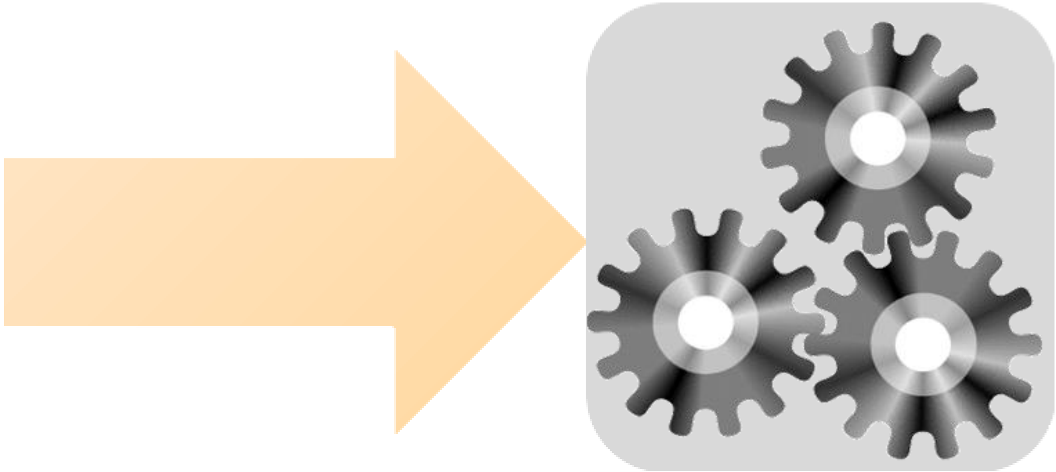
- si arresta senza dare alcun risultato
  (= sequenza in output vuota)



Come funziona?

- se riceve una sequenza che non è della forma accettata, allora

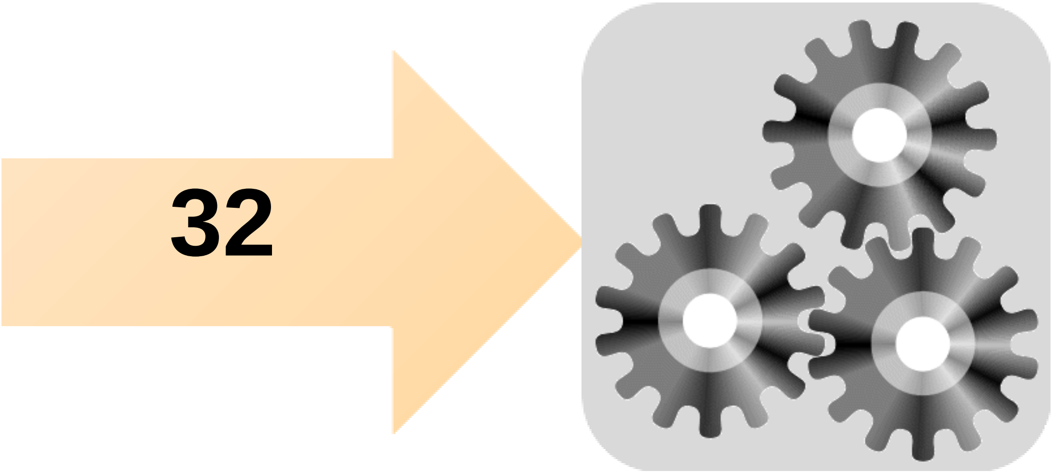
- si arresta senza dare alcun risultato
  (= sequenza in output vuota)



Come funziona?

▪ se riceve una sequenza che non è della forma accettata, allora

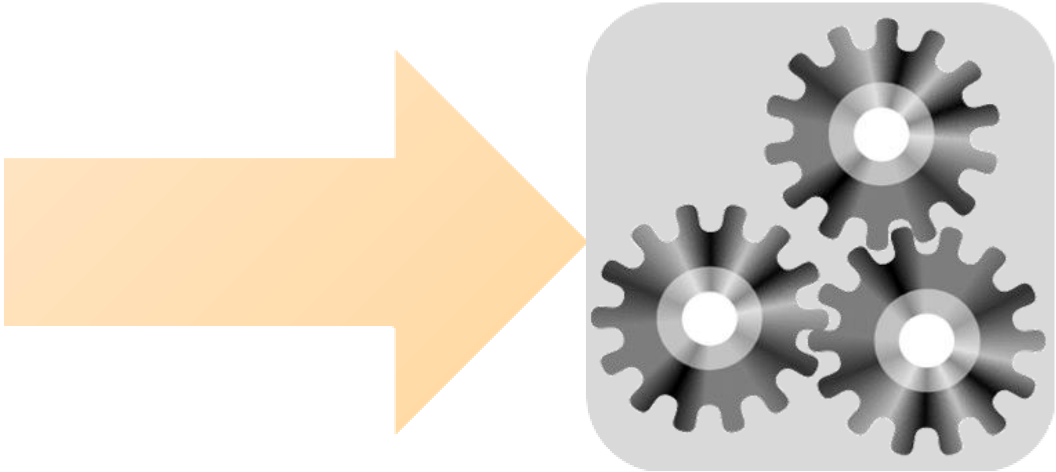
- si arresta senza dare alcun risultato
  (= sequenza in output vuota)



Come funziona?

▪ se riceve una sequenza che non è della forma accettata, allora

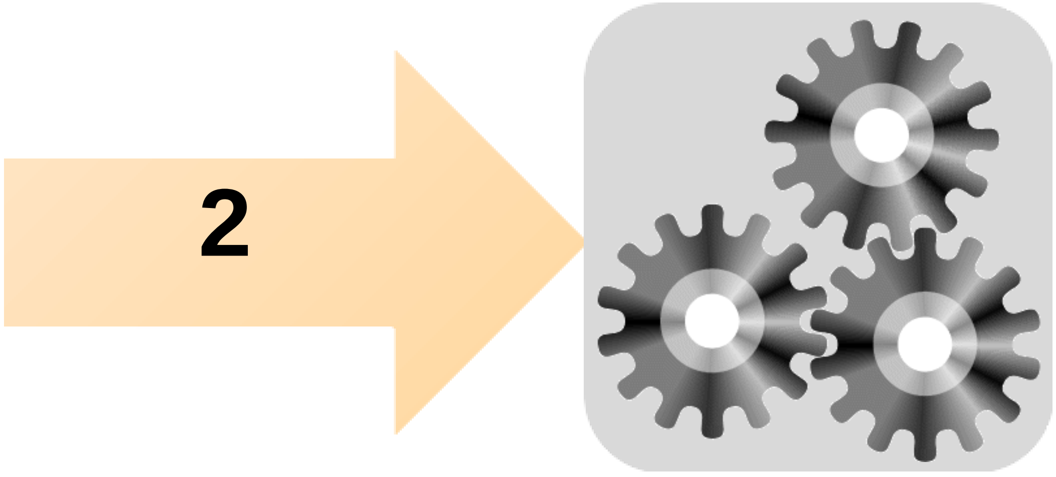
- si arresta senza dare alcun risultato
  (= sequenza in output vuota)



Come funziona?

▪ se riceve una sequenza che non è della forma accettata, allora

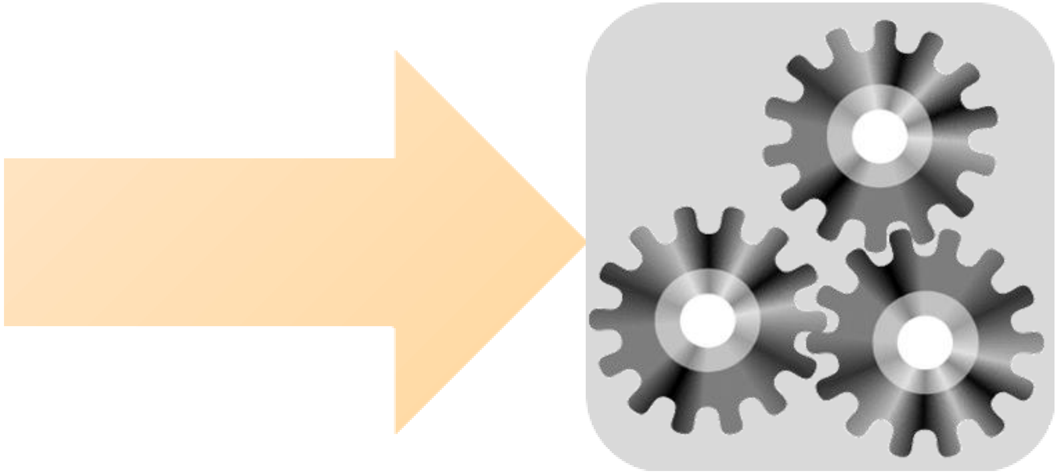
- si arresta senza dare alcun risultato
(= sequenza in output vuota)



Come funziona?

- se riceve una sequenza che non è della forma accettata, allora

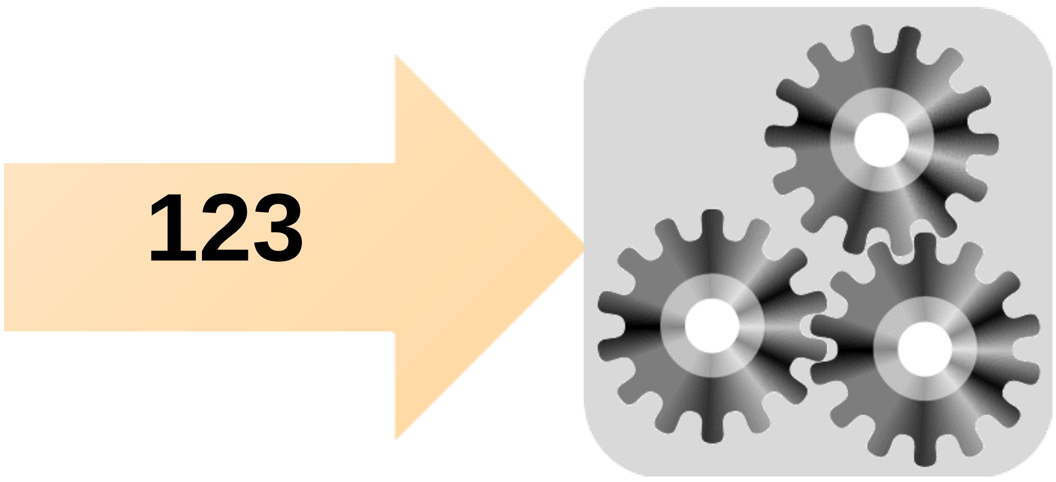
- si arresta senza dare alcun risultato
  (= sequenza in output vuota)



Come funziona?

- se riceve una sequenza che non è della forma accettata, allora

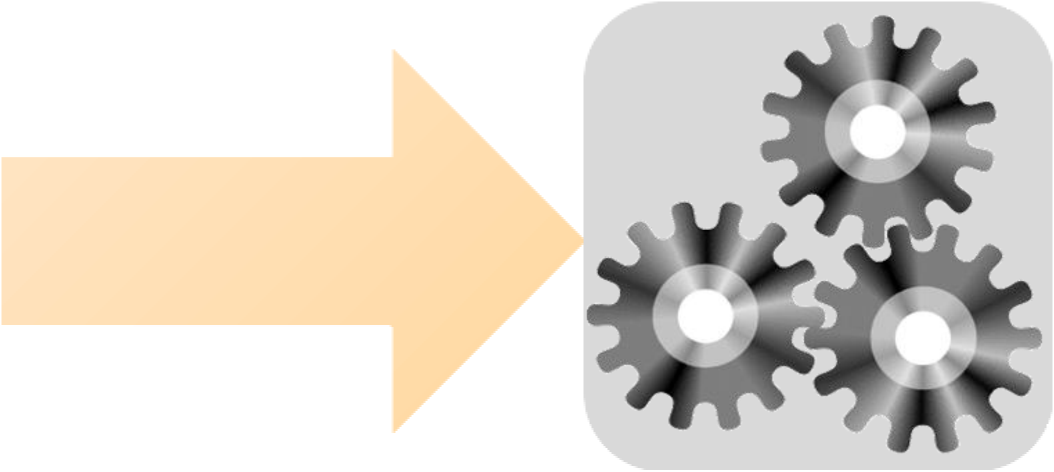
- si arresta senza dare alcun risultato
  (= sequenza in output vuota)



Come funziona?

- se riceve una sequenza che non è della forma accettata, allora

- si arresta senza dare alcun risultato
  (= sequenza in output vuota)



- se riceve 2 seguito dalla sequenza (non vuota) X, allora

\- produce in output proprio la sequenza X

- se riceve 2 seguito dalla sequenza (non vuota) X, allora

\- produce in output proprio la sequenza X

- se riceve 2 seguito dalla sequenza (non vuota) X, allora

\- produce in output proprio la sequenza X

- se riceve 2 seguito dalla sequenza (non vuota) X, allora

- produce in output proprio la sequenza X

- se riceve 2 seguito dalla sequenza (non vuota) X, allora

- produce in output proprio la sequenza X

- se riceve 2 seguito dalla sequenza (non vuota) X, allora

- produce in output proprio la sequenza X

- se riceve 3 seguito dalla sequenza (non vuota) X, allora

- prova a calcolare il proprio output come se ricevesse X

- se, prima o poi, da X ottiene la sequenza non vuota Y, allora dà come risultato finale la sequenza Y2Y

e poiché l’input 21 produce l’output 1 …

▪ se riceve 3 seguito dalla sequenza (non vuota) X, allora

- prova a calcolare il proprio output come se ricevesse X

- se, prima o poi, da X ottiene la sequenza non vuota Y, allora dà come risultato finale la sequenza Y2Y

… l’input 321 produce l’output 121

- se riceve 3 seguito dalla sequenza (non vuota) X, allora

- prova a calcolare il proprio output come se ricevesse X

- se, prima o poi, da X ottiene la sequenza non vuota Y, allora dà come risultato finale la sequenza Y2Y

e poiché, come abbiamo visto, l’input 321 produce l’output 121 …

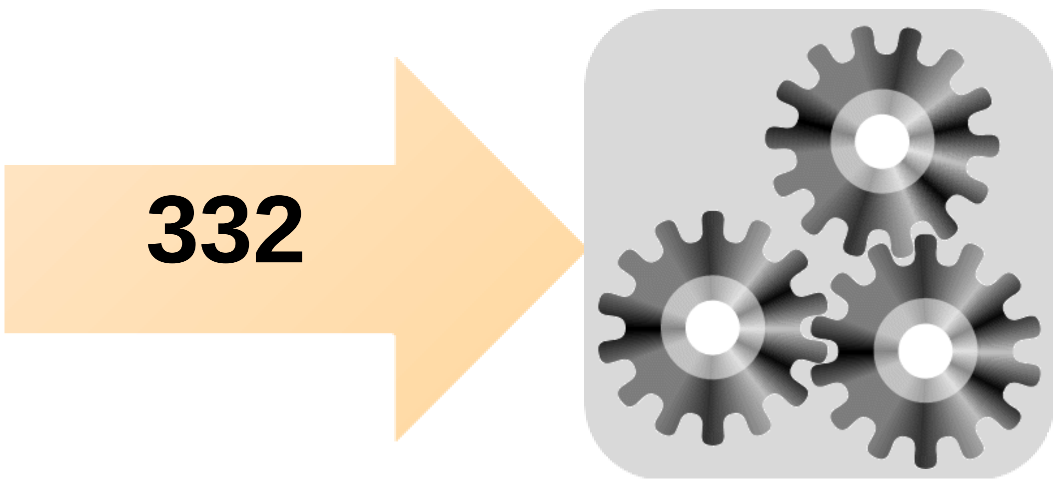
- se riceve 3 seguito dalla sequenza (non vuota) X, allora

- prova a calcolare il proprio output come se ricevesse X

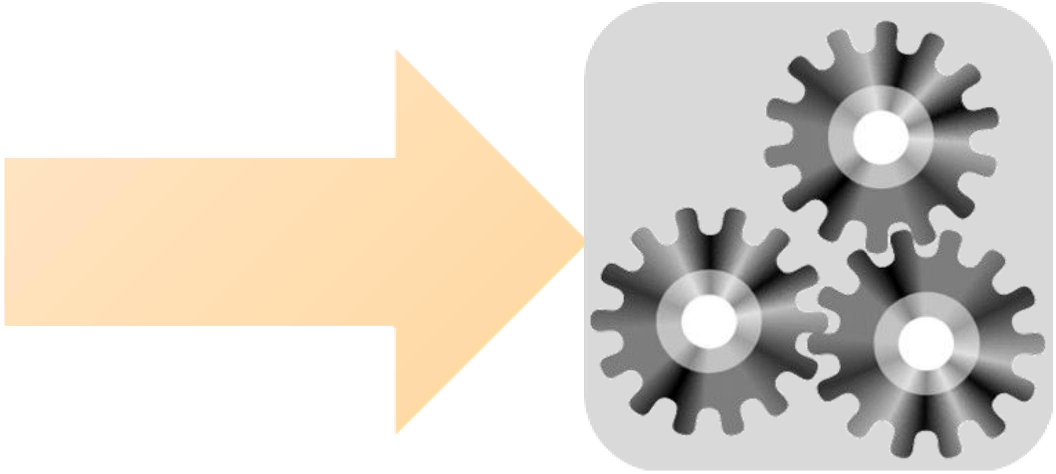
- se, prima o poi, da X ottiene la sequenza non vuota Y, allora dà come risultato finale la sequenza Y2Y

… l'input 3321 produce l'output 1212121

- se, invece, prima o poi, da X giunge a elaborare una sequenza che non è della forma accettata, allora la macchina si arresta senza dare alcun risultato



- se, invece, prima o poi, da X giunge a elaborare una sequenza che non è della forma accettata, allora la macchina si arresta senza dare alcun risultato

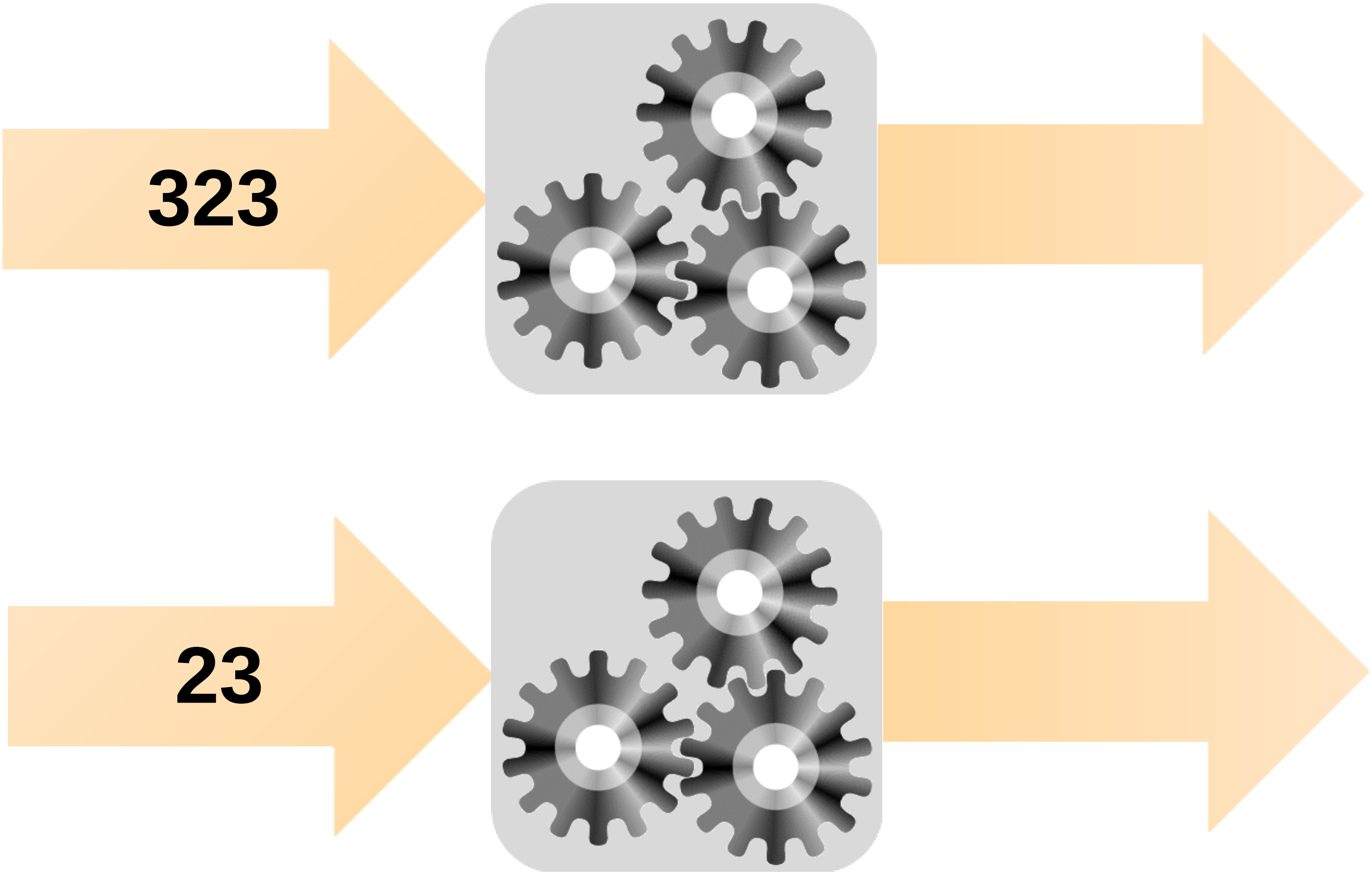


Che cosa produce la sequenza 3323?

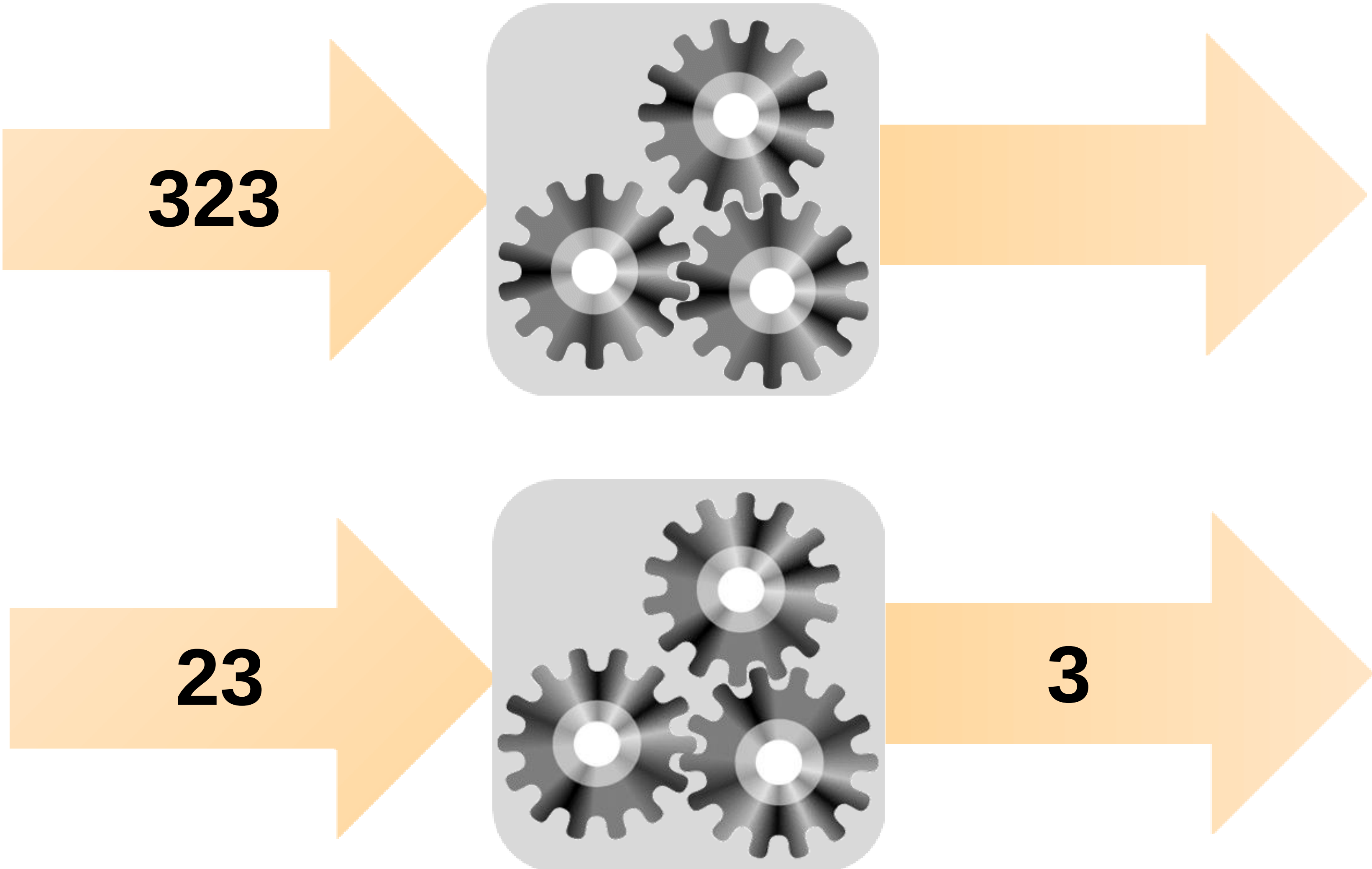
… si deve vedere che cosa produce la sequenza 323.

Se si ottiene una sequenza non vuota Y, allora l'output finale sarà Y2Y …

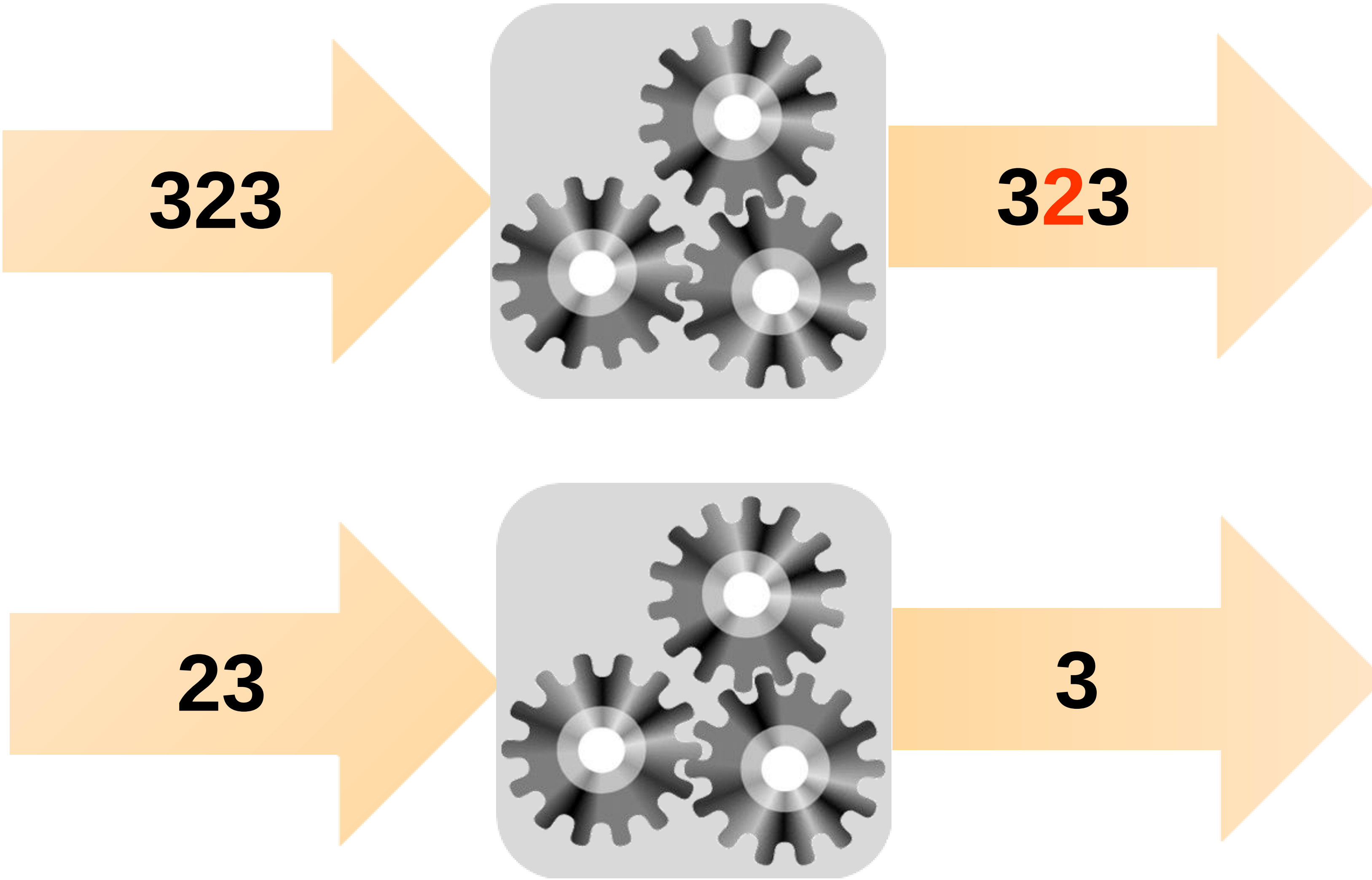
323 è la sola sequenza che produce sé stessa!



323 è la sola sequenza che produce sé stessa!



323 è la sola sequenza che produce sé stessa!



Che cosa produce la sequenza 3323?

… poiché la sequenza 323 produce sé stessa,

l'output finale è 3232323

Aggiungiamo un dispositivo …

Finché la macchina produce in uscita una sequenza non vuota, questa è riportata in ingresso come input successivo …

… Ora il processo può non terminare!

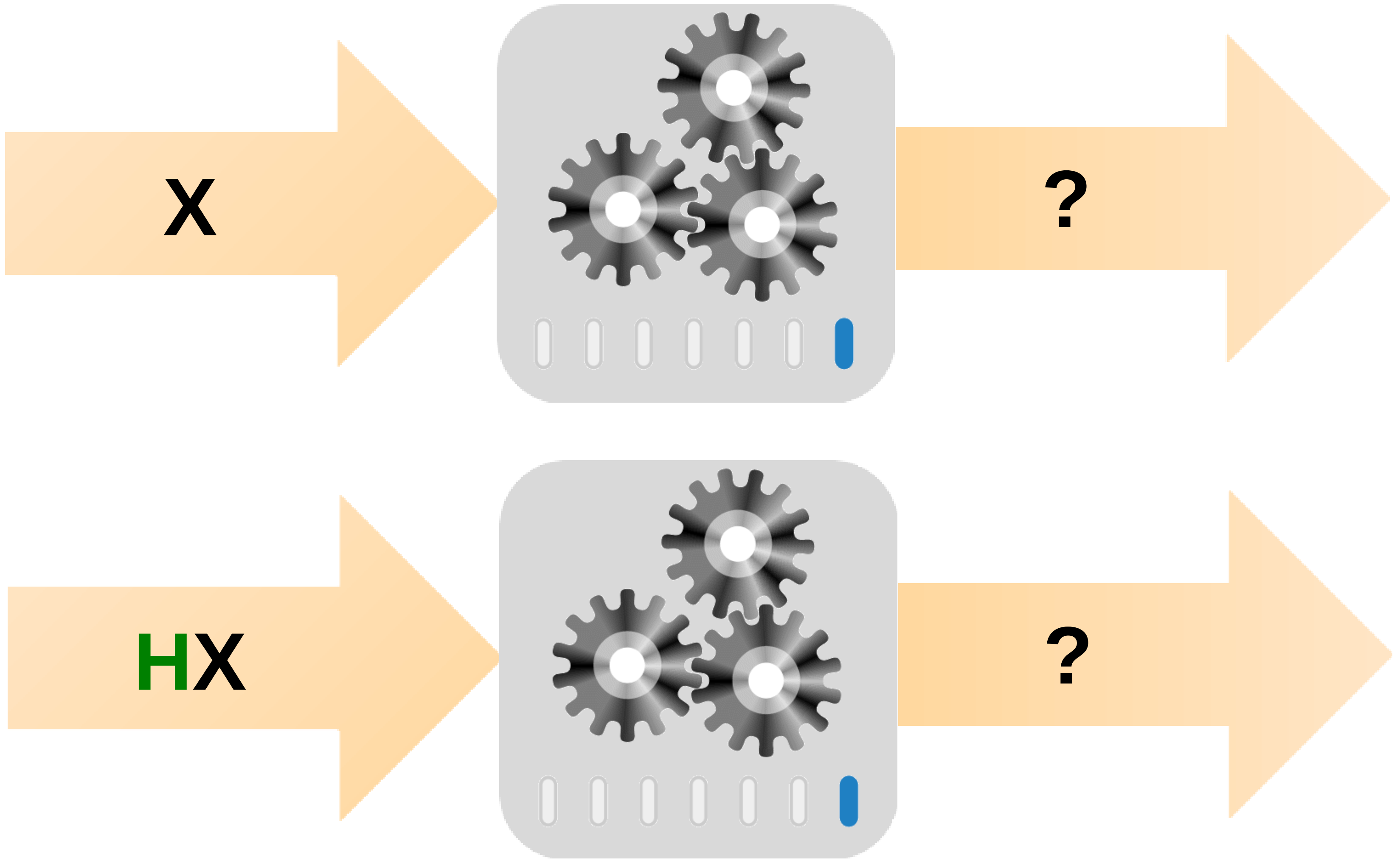
Esempi di input sui quali il processo non termina

- abbiamo visto che 323 → 323 …
- 3223 (poiché 223 → 23) → 23223 → 3223
- anche tutte le sequenze 32223, 322223, 3222223, … conducono a sé stesse e quindi il processo non termina
- 3232 → … → 32232 → … → 322232 → … → 3222232 → …
- 32323, 3232323, 323232323, … ciascuna di queste sequenze conduce, ad ogni passaggio successivo, a una sequenza della stessa forma e più lunga, e quindi la macchina non si fermerà …

Il problema dell'arresto

Esiste una sequenza H tale che, per ogni sequenza X, X e HX si comportino in maniera diversa rispetto alla terminazione?

Se così è, uno solo dei due processi termina …



Purtroppo non è così …

Qualunque sequenza H possiamo ipotizzare, basterà considerare la sequenza X = 32H3

Infatti, poiché 2H3 $\rightarrow$ H3,

32H3 $\rightarrow$ H32H3

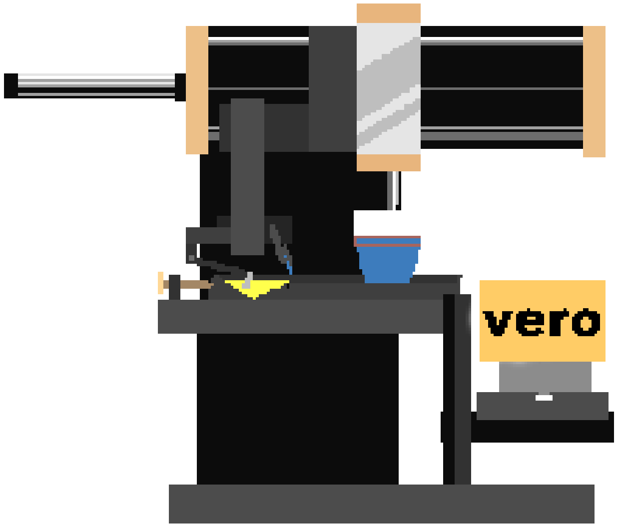
… e quindi

o su entrambe la macchina termina

o su entrambe la macchina non termina

Una macchina che parla di sé stessa

- elabora sequenze (finite e non vuote) di quattro simboli: S, N, A, −
- può stampare certe sequenze (stampabili), ma non altre
- una volta avviata, stamperà prima o poi ogni sequenza stampabile



Data una sequenza x, chiamiamo associata di x la sequenza x−x, e la indichiamo con A−x

Proposizioni:

sono tutte e sole le sequenze che hanno una di queste quattro forme

- S−x "x è stampabile"
- NS−x "x non è stampabile"
- SA−x "l'associata di x è stampabile"
- NSA−x "l'associata di x non è stampabile"

- S−AN “AN è stampabile”
- SA−N “l’associata di N (N−N) è stampabile”
- S−ANN “ANN è stampabile”
- NS−ANN “ANN non è stampabile”
- S−−NA “−NA è stampabile”
- S−− “− è stampabile”
- SA−− “l’associata di − (−−−) è stampabile”
- S−−−− idem: “−−− è stampabile”
- NSA−−S−A “l’associata di −S−A non è stampabile”
- NS−−S−A−−S−A idem: “−S−A−−S−A non è stampabile”

Per ogni sequenza x, valgono le quattro leggi della verità:

- S−x è vera sse x è stampabile
- NS−x è vera sse x non è stampabile
- SA−x è vera sse x−x è stampabile
- NSA−x è vera sse x−x non è stampabile

⇨ la macchina stampa proposizioni che fanno affermazioni su ciò che la macchina stessa può o non può stampare!

Assumiamo che la macchina sia corretta
cioè stampi soltanto proposizioni vere

- Se stampa S−x allora deve prima o poi stampare o aver già stampato anche x
- Se stampa SA−x allora deve stampare anche x−x
- Se stampa NS−x allora non può stampare S−x (non possono essere entrambe vere!) e nemmeno x (altrimenti avrebbe stampato una proposizione falsa, NS−x)

Una proposizione interessante …

Consideriamo la proposizione NSA−NSA

il cui significato è

“l’associata di NSA non è stampabile”

… ma l’associata di NSA è proprio NSA−NSA

NSA−NSA afferma la propria non-stampabilità: è vera sse non è stampabile

… e poiché la macchina è corretta, NSA−NSA è vera ma non-stampabile!

- nessuno può dire “io non sono un cavaliere” che equivale a “io sono un furfante” …

Sull’isola G …

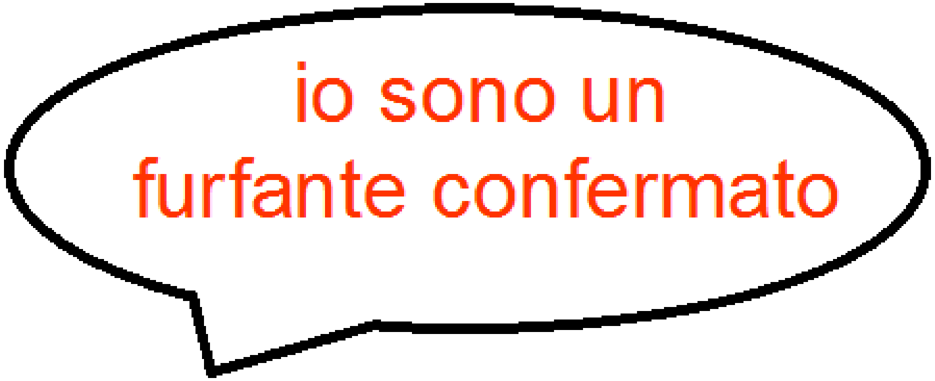
- … però è possibile che qualcuno dica:

io non sono un
cavaliere confermato

… e chi lo dice deve essere un cavaliere non confermato!

Sull'isola G …

- … è anche possibile che qualcuno dica:

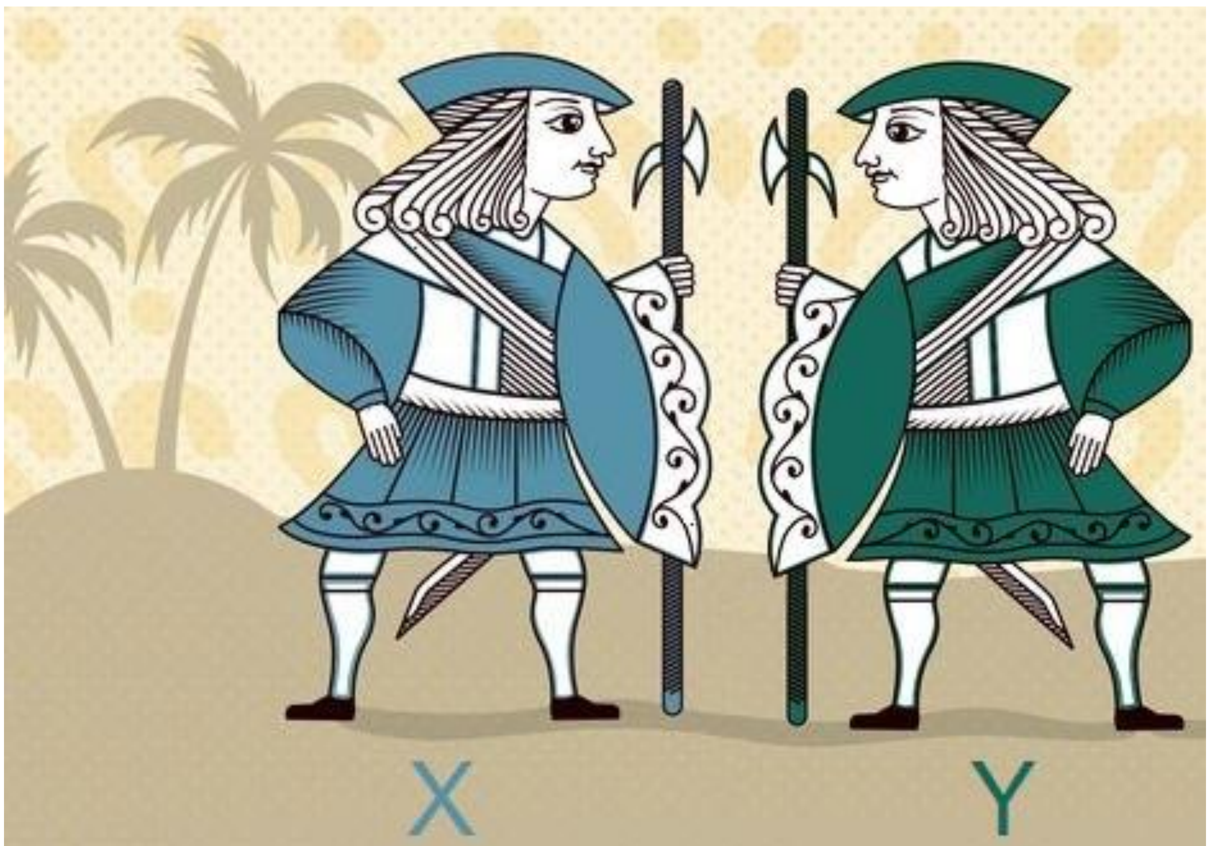


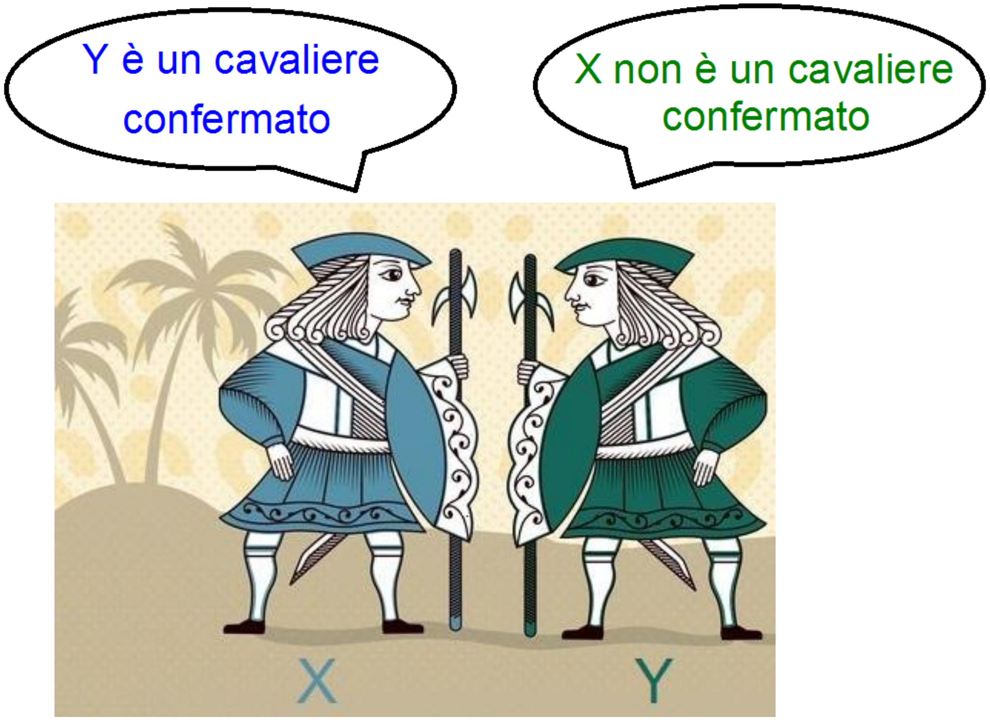
… e si tratta certamente di un furfante, ma non confermato!

Sull’isola G vi sono due abitanti, X e Y …

Y è un cavaliere confermato

X non è un cavaliere confermato





Che cosa possiamo dedurre?

- X non può essere un cavaliere confermato …

- se X è un cavaliere non confermato,

allora Y deve essere un cavaliere confermato

- se X è un furfante,

allora Y è un cavaliere non confermato

Conclusione:

uno e uno solo dei due è un cavaliere non confermato
… ma non possiamo sapere quale dei due lo sia!

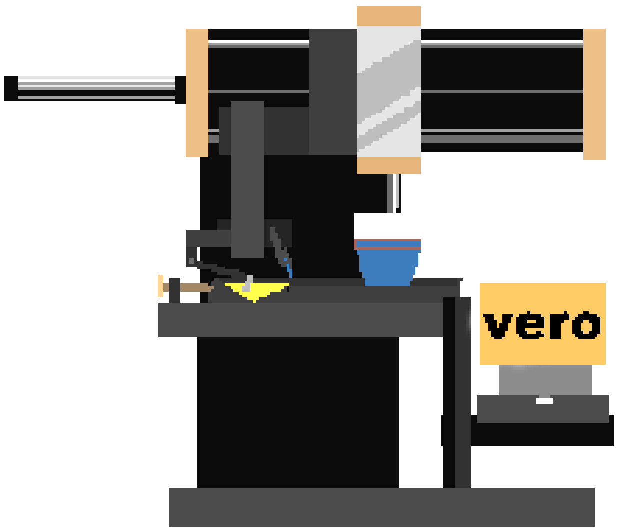
Analogamente, con la macchina che stampa verità …

- proposizione X: S−NSA−S−NSA

  “NSA−S−NSA (che è Y) è stampabile” …

- proposizione Y: NSA−S−NSA

  “l’associata di S−NSA (che è X) non è stampabile”



Analogamente, con la macchina che stampa verità …

- proposizione X: S−NSA−S−NSA

  "NSA−S−NSA (che è Y) è stampabile" …

- proposizione Y: NSA−S−NSA

  "l'associata di S−NSA (che è X) non è stampabile"

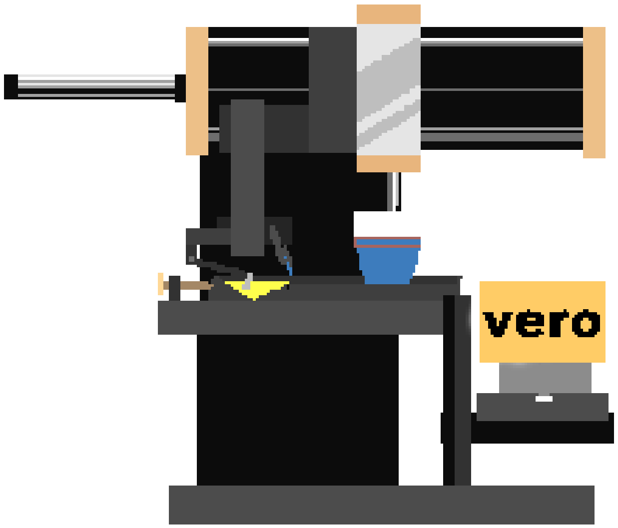
Due sono i casi possibili:

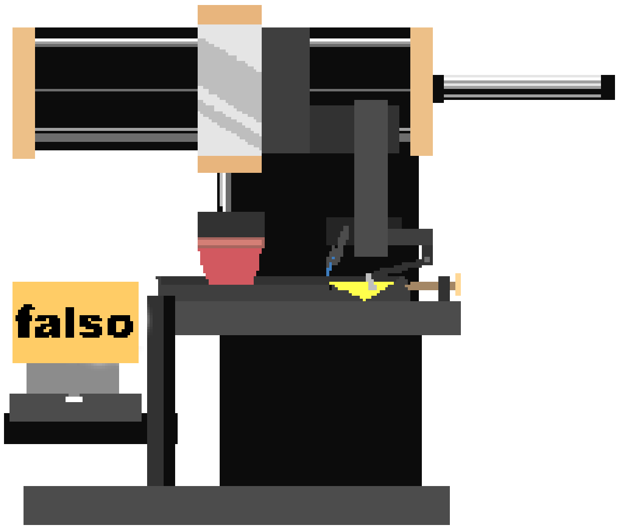
- X è vera e non stampabile, Y è vera e stampabile
- X è falsa e non stampabile, Y è vera e non stampabile

⇨ una tra X e Y è vera e non stampabile …
ma non v'è modo di stabilire quale delle due!

Due macchine M1 e M2 che parlano di sé stesse ... e ciascuna dell'altra

- M1 stampa soltanto proposizioni vere
- M2 stampa soltanto proposizioni false

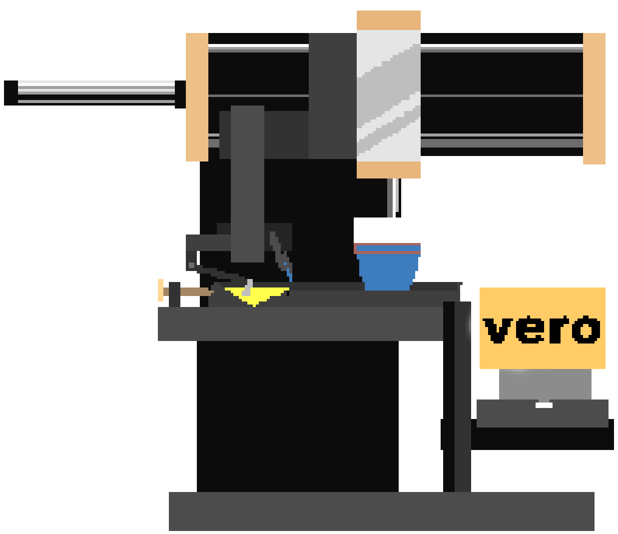




Aggiungiamo un quinto simbolo, R …

- S−x “x è stampabile da M1” ≡ “x è dimostrabile”
- R−x “x è stampabile da M2” ≡ “x è refutabile”

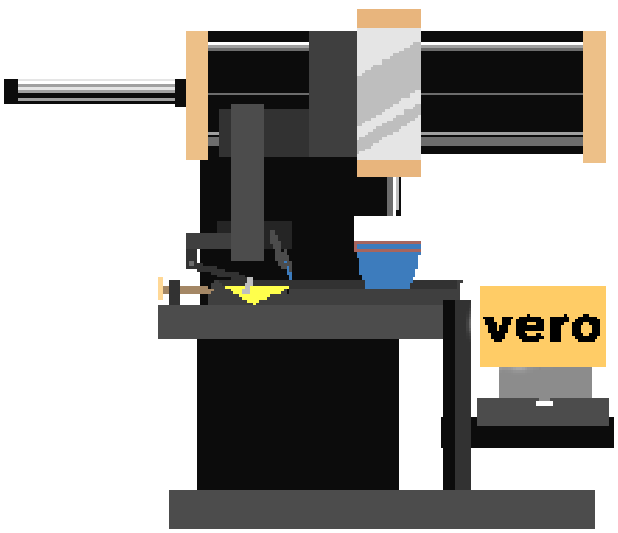
Che cosa afferma la proposizione RA−RA ?

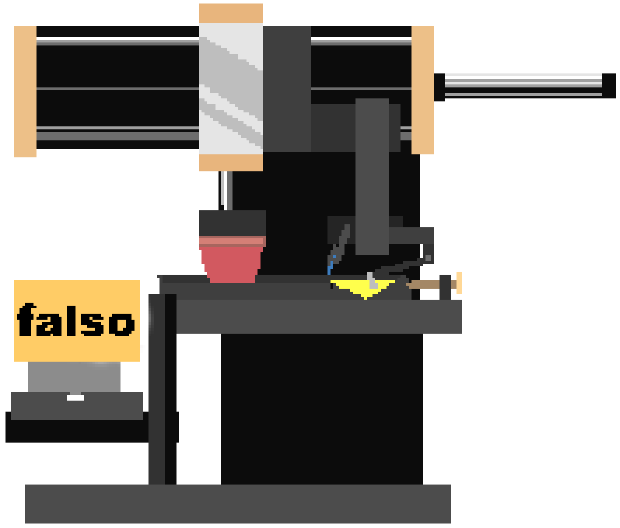


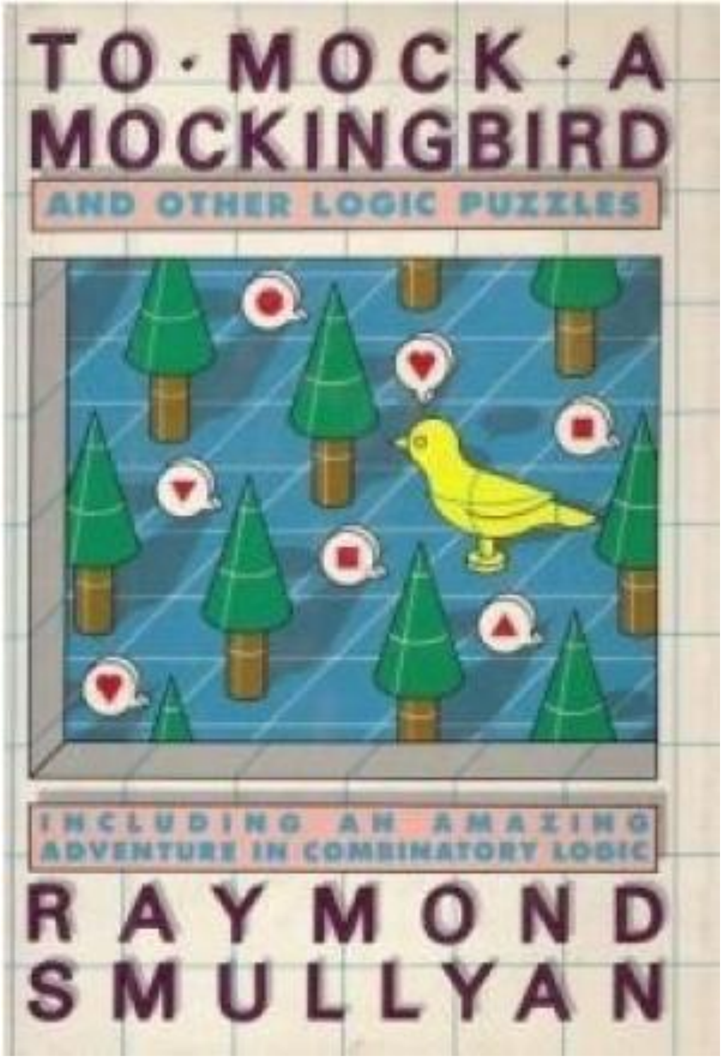
- RA−RA “l’associata di RA è refutabile”

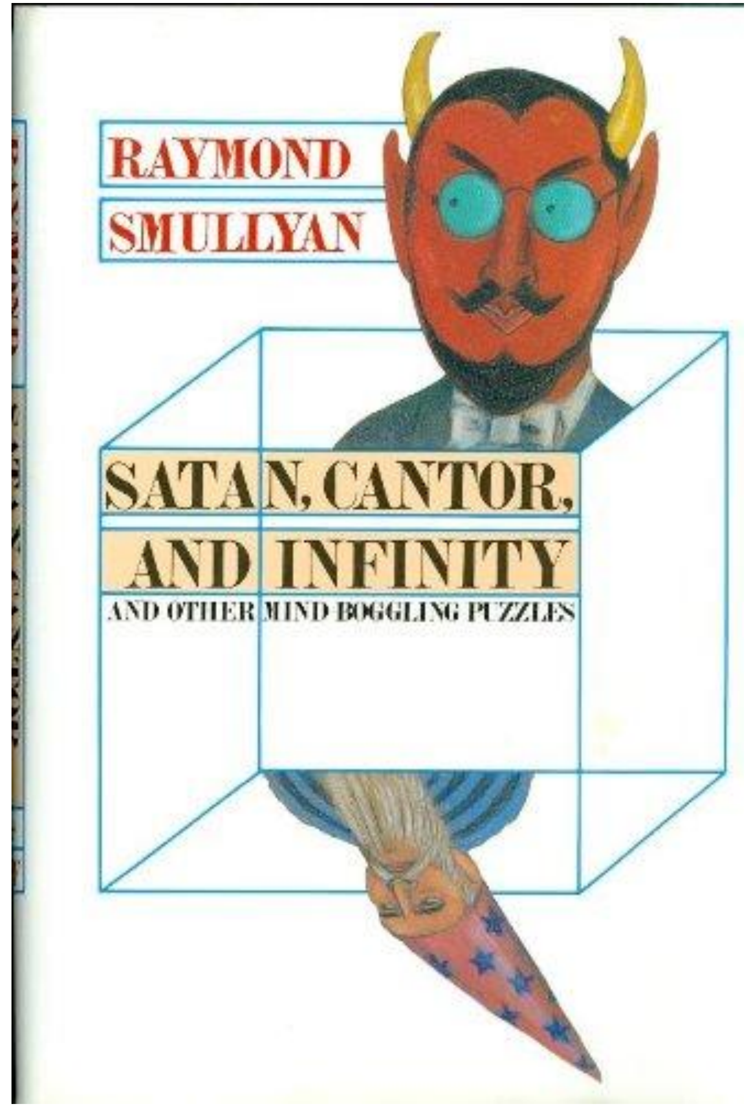
… ma l’associata di RA è proprio RA−RA

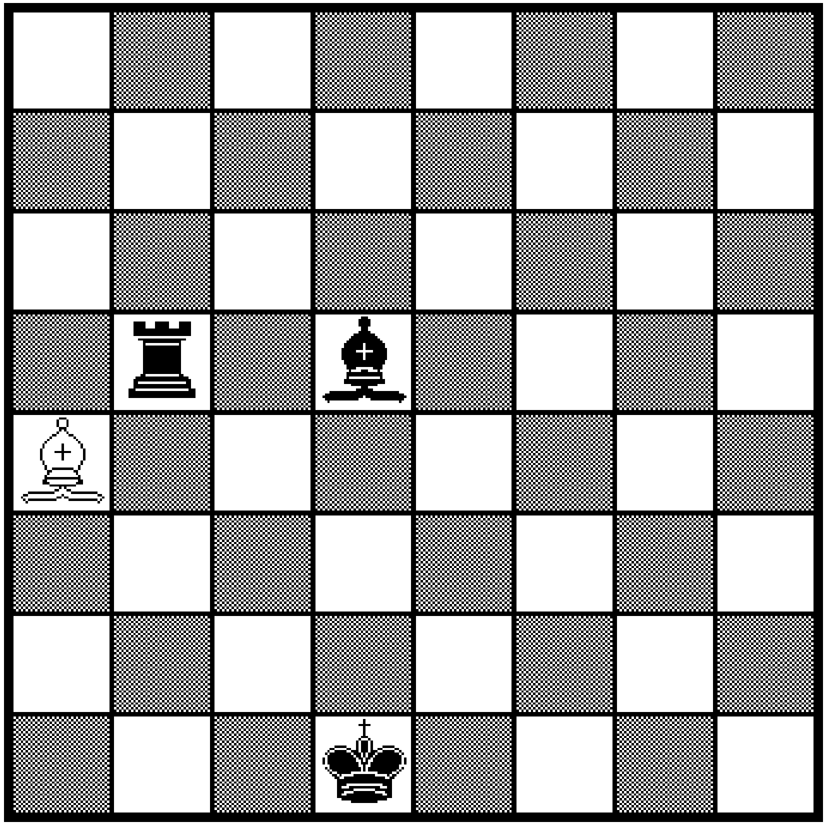
⇒ RA−RA è vera se e soltanto se è refutabile
… e poiché non può essere vera e refutabile,
dev’essere falsa ma non refutabile!





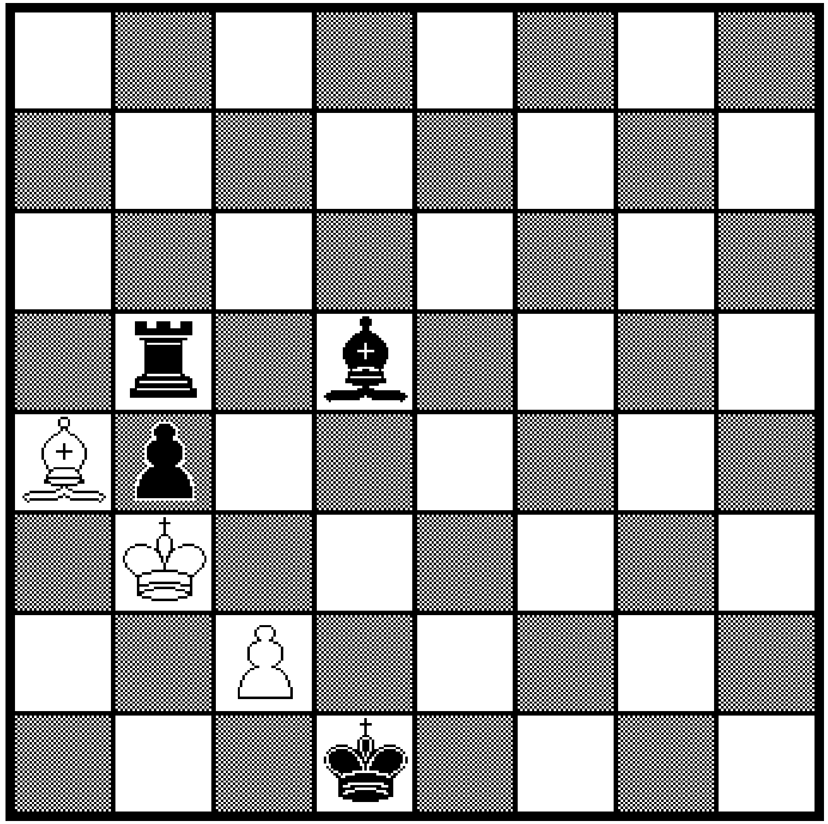

TO · MOCK · A
MOCKINGBIRD
AND OTHER LOGIC PUZZLES
INCLUDING AN AMAZING
ADVENTURE IN COMBINATORY LOGIC
RAYMOND
SMULLYAN


RAYMOND
SMULLYAN
SATAN, CANTOR,
AND INFINITY
AND OTHER MIND-BOGGLING PUZZLES



Il Re bianco è invisibile. Dove si trova?
E qual è stata l'ultima mossa?

(Manchester Guardian, 1957)



1. c2–c4, b4×c3 en passant + ; 2. Rb3×c3+


HILB V
HILB WXYZ
FCNL ABC
FCNL Ma-Mi

Alcune tappe fondamentali della logica

Le Scuole di Atene

- Accademia (Platone, IV sec. a. C.)

principio di non-contraddizione (o di consistenza): un enunciato e la sua negazione non possono essere entrambi veri

- Liceo (Aristotele, IV sec. a. C.)

funzionamento e uso dei quantificatori (nessuno, qualcuno, tutti)

- Stoà (Crisippo di Soli, III sec. a. C.)

logica proposizionale, con connettivi (negazione, congiunzione, disgiunzione, implicazione materiale) e regole di inferenza (modus ponens, modus tollens, …)

- Leibniz, 1676

Characteristica universalis: un linguaggio perfetto e puramente formale … la lingua degli odierni computer?

- Boole, 1847

le logiche di Aristotele e di Crisippo possono essere entrambe descritte da un'algebra con soli 0 e 1 …

- Frege, 1879

con la logica dei predicati (o delle relazioni) inizia la logica moderna …

- Post, 1920-21

completezza della logica proposizionale: i teoremi sono le tautologie

- Wittgenstein, 1921

tavole di verità

- Gödel, 1929-30

completezza della logica predicativa: i teoremi sono le formule valide (cioè vere in tutti i “mondi possibili” o, meglio, *modelli*) …

L'aritmetica di Peano del prim'ordine

1. $\forall x\ (\sim(s(x) = 0))$
2. $\forall x\ \forall y\ (s(x) = s(y) \rightarrow x = y)$

3. $\forall x\ (x + 0 = x)$
4. $\forall x\ \forall y\ (x + s(y) = s(x + y))$

5. $\forall x\ (x \cdot 0 = 0)$
6. $\forall x\ \forall y\ (x \cdot s(y) = (x \cdot y) + x)$

7. $(P(0) \wedge \forall x\ (P(x) \rightarrow P(s(x)))) \rightarrow \forall x\ P(x)$

schema d'assioma, per ogni P con una variabile libera *x* (eventuali altre variabili libere sono intese quantificate universalmente all'esterno)

- Gödel, 1931

incompletezza dell’aritmetica: nell’aritmetica esistono formule vere indimostrabili, e quindi l’aritmetica non è riducibile alla logica; inoltre, la coerenza dell’aritmetica deve essere provata “al di fuori” di essa …

- Turing, 1936

halting problem: non esiste un algoritmo per decidere se una formula arbitraria della logica predicativa è un teorema, ma soltanto per semi-deciderlo …

- Tarski, 1936

la verità non è definibile nel linguaggio, ma soltanto nel metalinguaggio: se nell’aritmetica fosse esprimile una definizione di verità, allora vi si potrebbe derivare il …

Paradosso del mentitore

- Eubulide di Mileto (IV sec. a. C.), discepolo di Euclide, fondatore della Scuola di Mégara:

« io sto mentendo » (ψευδόμενος)

- paradosso (contraddizione, antinomia): affermazione che non può essere né vera né falsa

- … è sempre autoreferenziale?

- Buridano (Jean Buridan, XIV sec.), scolastico

il problema non sta nell'auto-riferimento!

Socrate: « Platone dice il falso »

Platone: « Socrate dice il vero »

Se Platone dicesse « Socrate dice il falso » non vi sarebbe alcuna contraddizione.

- Jourdain, 1913

La frase seguente è falsa.

La frase precedente è vera.

Di per sé, ciascuna di esse può essere vera o falsa … tuttavia non possono stare insieme!

Tante varianti del “paradosso del mentitore”

- la più antica risale a Epimenide di Cnosso, cretese del VI sec. a. C. o precedente:

« [tutti] i cretesi sono [sempre] bugiardi »

… ma non è affatto un paradosso, è una frase falsa!

- Diogene Laerzio (III sec.), la mamma al coccodrillo:

« tu mangerai il mio bambino »

- Cervantes (1615), al ponte di Sancho Panza:

« voglio attraversare il ponte per farmi impiccare »

Tante varianti del “paradosso del mentitore”

- Grelling (1908)

Autologico: che si riferisce a sé stesso
Eterologico: che non si riferisce a sé stesso

« eterologico è autologico o eterologico? »

… un esempio riferito alla sola grammatica, svincolato da verità e falsità!

- Russell (1918)

In un villaggio c’è un solo barbiere, il quale rade tutti e soli gli abitanti che non si radono da soli …

« chi rade il barbiere? »

Tentativi di soluzione

- i paradossi sono privi di significato (Crisippo)
- distinzione tra uso e menzione di una frase (Aristotele)

« un monosillabo » ha sei sillabe

… però il paradosso si può ripresentare (Quine, 1962):

« è falsa se preceduta dalla sua menzione »
è falsa se preceduta dalla sua menzione

- piccolissima parentesi:

Domanda: Qual è il titolo di questo libro?
Risposta: « Qual è il titolo di questo libro? »

Tentativi di soluzione

- ogni affermazione va associata a un istante di tempo (Buridano, “precursore” della logica temporale)

… ma il problema rimane:

S.: « P. dirà il falso quando pronuncerà la frase seguente »

P.: « S. disse il vero quando pronunciò la frase precedente »

- oltre a “vero” e “falso” ci sono altri valori (intermedi) di verità … (ad esempio: logica fuzzy, anni 1960-70)

- distinzione tra linguaggio e metalinguaggio (Guglielmo di Occam, XIV sec.): è il tentativo più convincente, che sarà formalizzato da Tarski negli anni ’30 del Novecento

Linguaggio e metalinguaggio

- Linguaggio: la lingua di cui si sta parlando
- Metalinguaggio: la lingua usata per parlare del linguaggio

Quando si dice che qualcosa è vero o qualcos'altro è falso si fanno affermazioni nel metalinguaggio.

Limitazione "semantica" del linguaggio (Tarski):

se la definizione di verità del linguaggio si potesse esprimere all'interno del linguaggio stesso,

allora si potrebbero formulare paradossi o contraddizioni

→ la teoria sarebbe inconsistente

→ la definizione di verità deve stare fuori dal linguaggio

I giochi logici di Raymond Smullyan
tra nozioni di base e limiti dell'informatica

Circolo Filologico Milanese − 17 marzo 2018

Oggi vi parlerò di un grande e singolare personaggio, Raymond Smullyan, un uomo dall'ingegno a dir poco poliedrico: fu infatti matematico, logico, filosofo, scacchista, scrittore, saggista, prestigiatore e pianista.
In gioventù fu combattuto tra la musica e la matematica, disciplina che approfondì da autodidatta, ma in realtà la sua prima carriera fu quella di prestigiatore…

E in effetti un tocco di magia è riuscito a infonderlo appieno nei suoi numerosi libri, sia nelle opere accademiche sia soprattutto nei testi divulgativi, eccezionali libri di logica e matematica ricreative.
Tutti posseggono una caratteristica rara e mirabile: affrontano argomenti assai ardui non solo in modo chiaro e comprensibile, ma anche affascinante e piacevole.

Smullyan era un umorista, sempre pronto a battute scherzose, osservazioni sagaci, aforismi al limite del nonsense…

Smullyan fu allievo di Alonzo Church e ottenne il Ph.D. in matematica all'università di Princeton, dove rimase fino al 1961, anno in cui pubblicò l'elegante monografia *Theory of Formal Systems*, esemplare spiegazione − con varianti e miglioramenti − della teoria degli insiemi ricorsivamente enumerabili.
Gli insiemi *ricorsivi* sono caratterizzati da una proprietà, o predicato, decidibile per via automatica, cioè mediante un algoritmo: ad esempio, i numeri primi, i programmi sintatticamente corretti in un certo linguaggio di programmazione…
Gli insiemi *ricorsivamente enumerabili*, ma non ricorsivi, sono quelli legati a questioni, o predicati, soltanto *semidecidibili*: oltre al "problema dell'arresto" (il primo, significativo esempio di cui parleremo a breve), un altro problema famoso provatamente soltanto semidecidibile consiste nello stabilire se un'equazione a coefficienti interi con numero di variabili e grado arbitrari ammetta soluzioni intere.
Gli insiemi nemmeno ricorsivamente enumerabili sono quelli relativi a questioni *totalmente indecidibili* per via automatica: ad esempio, la non-terminazione (poiché è la negazione di un predicato semidecidibile ma non decidibile!); oppure: dati due programmi qualsiasi, stabilire se sono funzionalmente equivalenti…

Il punto di arrivo delle riflessioni di Smullyan sui classici teoremi limitativi della logica matematica, che lo hanno accompagnato lungo il corso della sua lunga vita, è costituito da un saggio sui teoremi di incompletezza di Gödel.
Già dagli anni trascorsi a Princeton, egli sosteneva che l'*incompletezza* si ritrova in sistemi formali più elementari di quelli considerati dal celebre logico austriaco Kurt Gödel nel suo fondamentale lavoro del 1931, e qui cercheremo di capire perché…

In seguito, Smullyan rilevò che almeno una parte dell'ammirazione suscitata dal teorema di Gödel dovrebbe spettare a quello di Tarski sull'*indefinibilità della verità*, secondo il nostro autore ugualmente rivoluzionario dal punto di vista filosofico, oltre che più facile da dimostrare: sotto certe ipotesi, un sistema coerente (o consistente, cioè che non dia luogo a contraddizioni) non può contenere in sé la definizione della propria verità, che va espressa in un opportuno *metalinguaggio*.

Quanto alle sue opere divulgative, citiamo anzitutto *What is the name of this book?* del 1978, pubblicato tre anni dopo in italiano dalla Zanichelli. Il capitolo conclusivo, che fra poco chiameremo in causa, era già dedicato a una prima trattazione in chiave giocosa del teorema di Gödel.
Martin Gardner, il prolifico curatore per quasi un quarto di secolo della rubrica di giochi matematici sulla rivista *Le Scienze*, riteneva questa raccolta di rompicapi la più originale, profonda e spiritosa mai scritta nel suo genere.
Attingeremo pure a *The lady or the tiger?* del 1982 (tradotto in italiano col titolo *Donna o tigre?* uscito ancora da Zanichelli nel 1985): qui entrano in scena diverse "macchine" che ci aiuteranno a capire sia il problema dell'arresto di Turing sia l'idea di Gödel « sotto la luce più chiara che io possa immaginare » (per citare le parole dello stesso Smullyan)…

In vecchiaia, ormai Professore Emerito di Filosofia all'Indiana University, tornò alla musica e incise alcune esecuzioni al pianoforte dei suoi pezzi classici preferiti, composti da Bach, Domenico Scarlatti e Schubert.
Si interessò pure di misticismo religioso e filosofie orientali, specialmente del taoismo e di come questo possa unificarsi con la logica e la matematica.

Tra il 1929 e il 1930, Gödel provò la *completezza della logica predicativa*. Che cosa significa? Se consideriamo una teoria, con i suoi assiomi propri (che sono enunciati, formule ben formate, composti con predicati, ossia funzioni a valori di verità, che esprimono sia proprietà di certi oggetti sia relazioni tra questi) e le classiche regole di inferenza (mediante le quali si possono dedurre le conseguenze di tali assiomi), e questa teoria è coerente (e dunque ammette almeno un modello, un "mondo possibile", pur astratto, in cui tutti gli assiomi siano simultaneamente veri), allora ciò che è dimostrabile per via automatica coincide con ciò che è vero in *tutti* i modelli.
Se però restringiamo l'indagine a modelli particolari (dove le verità possono essere "di più"), allora la completezza può essere perduta… Nel 1931, Gödel dimostrò l'*incompletezza dell'aritmetica e dei "sistemi affini"*: se vogliamo che la teoria comprenda le usuali proprietà dei numeri naturali, con le operazioni di addizione e moltiplicazione, e intendiamo riferirci al suo "modello standard", allora avremo a che fare con verità indimostrabili; dunque l'aritmetica non è riducibile alla logica predicativa. (*Nota:* l'aritmetica di Peano del prim'ordine, con sei assiomi e uno schema d'assioma propri, ha un'infinità di modelli, mutuamente non isomorfi.)
Contrariamente alle speranze di Hilbert, anche la coerenza della teoria non può essere provata all'interno della teoria stessa (secondo teorema di incompletezza)…

Una *macchina di Turing* è la definizione di un certo procedimento di calcolo, a partire da un input arbitrario, secondo un particolare formalismo. Nel 1936, Alan Turing mostrò la possibilità di definire, usando lo stesso formalismo, una “macchina universale” (di cui i moderni computer sono una realizzazione limitata, almeno per quanto concerne la disponibilità di memoria), in grado di emulare l’esecuzione di un algoritmo arbitrario (nella fattispecie, una qualsiasi macchina di Turing) su un input pur’esso arbitrario… Tuttavia, mostrò anche il rovescio della medaglia: non può esistere alcuna macchina che, a fronte di tali dati arbitrari, possa decidere se l’algoritmo dato terminerà il suo processo di calcolo sull’input dato: potrà soltanto semideciderlo, dicendo (prima o poi) “sì” tutte le volte che il processo terminerà.

Alonzo Church arrivò, nello stesso anno di Turing (il 1936), a provare che *la logica predicativa è indecidibile* (o, più precisamente, soltanto semidecidibile): non esiste alcun algoritmo in grado di stabilire se una sua formula arbitraria è *valida* (cioè vera in ogni possibile modello, ciò che qui coincide con la dimostrabilità, come aveva provato Gödel col teorema di completezza) oppure no; può solo semideciderlo.
Un tale algoritmo potrebbe infatti decidere, tra l’altro, le istanze del problema dell’arresto, giacché è possibile tradurre queste istanze in formule logiche.

Dunque, una limitazione sulla dimostrabilità appare già nella logica, e non soltanto nell’aritmetica come aveva provato Gödel col suo primo teorema di incompletezza. Nell’ambito della logica, non vi sono procedure effettive per refutare falsità, in generale.
Nell’ambito dell’aritmetica, invece, Gödel aveva già escluso persino la possibilità di dimostrare (automaticamente) tutte le verità: per fortuna, nella matematica, non basta, in generale, un procedimento di calcolo automatico, ma occorrono intuito e ingegno! Si tratta senza dubbio del contributo più significativo apportato alla logica dai tempi di Aristotele (IV secolo a. C.), anche in un senso culturale più ampio: ha fatto capire la distinzione tra verità e dimostrabilità!

Vorrei ricordare, infine, altri due (tra i tanti!) libri “abbordabili” di Smullyan: *To mock a mockingbird* (letteralmente: sbeffeggiare il tordo beffeggiatore) del 1985 (*Fare il verso al pappagallo*, Bompiani, 1990), dove si tratta di logica combinatoria (Schönfinkel e Curry, 1924-30: calcolo di due combinatori), che prelude al *lambda-calcolo* di Church (1931-32), prototipico linguaggio funzionale, universalmente espressivo. E poi *Satan, Cantor, and Infinity* del 1992 (*Satana, Cantor e l’infinito*, Bompiani, 1994) dove l’autore immagina un’isola popolata da robot “intelligenti”, ciascuno dei quali agisce in base a un programma (scritto al suo interno in un linguaggio con poche regole), sicché questi robot ne possono creare altri, fornendoli di un’intelligenza sufficiente per costruire ancora robot intelligenti, i quali a loro volta ne creano altri, e così via all’infinito… però, sul più bello, possono intervenire regole di distruzione! In questo libro Smullyan continua così il suo *excursus* sugli approcci più semplici – ma non per questo semplicistici – a Gödel e Tarski…

L. R.