Logica Monadica (MFO/MSO)

Solo per linguaggi

Caratteristiche:

- Var. sono def. su un sottoinsieme finito di $\mathbb{N}$
  $\{0, 1, \ldots, n-1\}$, dove n è la lung. della stringa

- formula φ può essere:
  - $\neg \varphi$
  - $\varphi_1 \wedge \varphi_2$
  - $\forall x (\varphi)$
  - $i(x)$ $\quad \forall i \in I$
    in pos. x c'è la lettera "i"
    $b(3)$
  - $x < y$

- $\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$
- $\varphi_1 \Rightarrow \varphi_2 = \neg\varphi_1 \vee \varphi_2$
- $\exists x\,(\varphi) = \neg\forall x\,(\neg\varphi)$
- $x = y$
- $x \leq y$
- costante 0
- Pred. di successore $y = S(x) \equiv y = x+1$
- costanti num. $1, 2, \ldots$
- last(x)

- $y = x + k$
- $y = x - k$

es.: stringhe che iniziano con "a" e hanno almeno 3 "b"

(MFO)

$$a(0) \wedge \exists x_1, x_2, x_3 \left(b(x_1) \wedge b(x_2) \wedge b(x_3) \wedge x_1 \neq x_2 \wedge x_2 \neq x_3 \wedge x_1 \neq x_3 \right)$$

in pos. 0 c'è "a"

$a(x) \equiv$ in pos. x c'è "a"

$b(x) \equiv$ in pos. x c'è "b"

Es:

$$L = a^+ b^+ = a a^* b b^*$$

$$aaabbbb \in L$$

$y < x \qquad x \leq y \leq \ell$

$$a(0) \wedge \exists \ell \, (last(\ell) \wedge b(\ell) \wedge \exists x \, (b(x) \wedge \forall y \, (y < x \Rightarrow a(y)) \wedge \forall y \, (x \leq y \leq \ell \Rightarrow b(y))))$$

es.: $L = a^{2k+1}$, $k \geq 0$

non è possibile utilizzare MFO

Utilizziamo la MSO:

problema

$$\exists P \left(P(0) \land \forall x \left(P(x) \Leftrightarrow \neg P(x+1) \right) \land \forall x \left(a(x) \right) \land \exists \ell \left(last(\ell) \land P(\ell) \right) \right)$$

$$\begin{array}{ccccc} a & a & a & a & a \\ 0 & 1 & 2 & 3 & 4 \end{array} \in L$$

$$L = a^{2k+1}$$

Utilizziamo la MSO:

$$\exists P \left(P(0) \wedge \forall x \left(\neg last(x) \Rightarrow \left(P(x) \Leftrightarrow \neg P(x+1) \right) \right) \wedge \forall x \left(a(x) \right) \right.$$

$$\left. \wedge \; \exists \ell \left(last(\ell) \wedge P(\ell) \right) \right)$$

$a \; a \; a \; a \; a \in L$

$0 \; 1 \; 2 \; 3 \; 4$

es.:

$$L = (e^+ h \mid la)^+$$

MFO

▷ $e(\emptyset) \vee l(\emptyset)$

▷ $\exists x \, (last(x) \wedge (h(x) \vee a(x)))$

▷ $\forall x \, (e(x) \Rightarrow (e(x+1) \vee h(x+1)))$

▷ $\forall x \, (l(x) \Rightarrow a(x+1))$

▷ $\forall x \, (h(x) \Rightarrow (e(x+1) \vee l(x+1)))$ ←

▷ $\forall x \, (a(x) \Rightarrow (e(x+1) \vee l(x+1)))$ ←

non vanno bene se sono "last"

$$L = (e^+ h \mid la)^+$$

- $e(0) \vee l(0)$
- $\exists x\,(last(x) \wedge (h(x) \vee a(x)))$
- $\forall x\,(e(x) \Rightarrow (e(x+1) \vee h(x+1)))$
- $\forall x\,(l(x) \Rightarrow a(x+1))$
- $\forall x\,((h(x) \wedge \neg last(x)) \Rightarrow (e(x+1) \vee l(x+1)))$
- $\forall x\,((a(x) \wedge \neg last(x)) \Rightarrow (e(x+1) \vee l(x+1)))$

Calcolabilità

Tesi di Church:

1. Non c'è formalismo per modellare calcolo meccanico più potente delle TM

2. Ogni algoritmo può essere codificato con TM (o formalismo equiv.)

Sinonimi: algo, TM, programma, procedura, ...

Le MT possono essere enumerate $\{MT\} \overset{\exists}{\longleftrightarrow} \mathbb{N}$

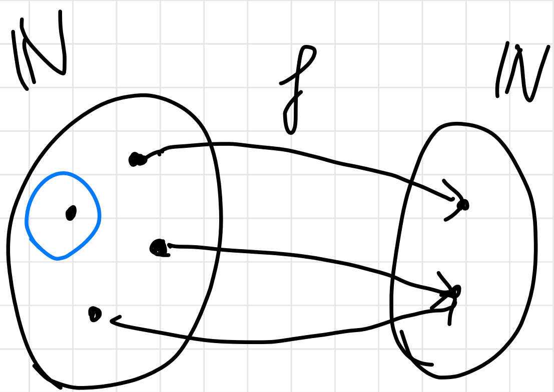
Problema: richiesta di calcolo (algoritmico) di $f : \mathbb{N} \to \mathbb{N}$

(oppure insiemi numerabili)

in generale parziale

$$\exists n \; (f(n) = \perp)$$

la fun. è indef.



$\exists$ algo. (MT, prog.) che calcola f $\iff$ f computabile (o calcolabile) $\iff$ Problema risolvibile (calcolabile)

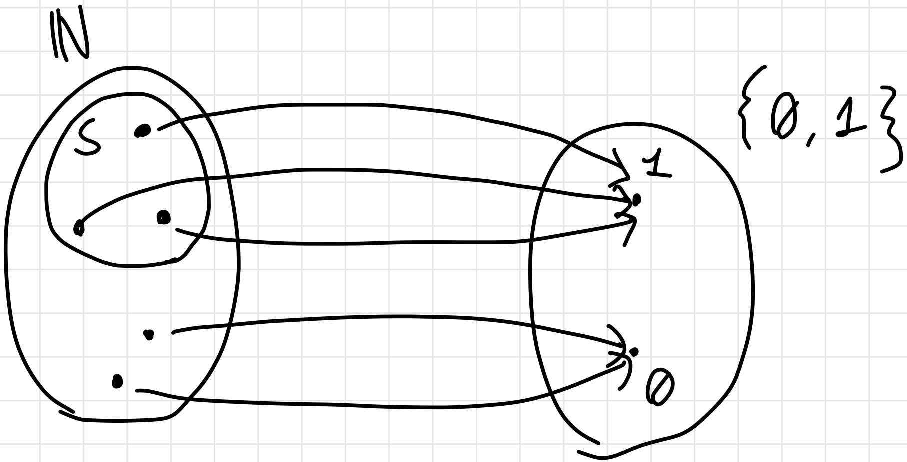
$$\left|\{f : \mathbb{N} \to \mathbb{N}\}\right| \geq \left|\{f : \mathbb{N} \to \{0,1\}\}\right|$$

$$= |\mathcal{P}(\mathbb{N})| = 2^{|\mathbb{N}|} > |\mathbb{N}| = |\{MT\}|$$

$$2^{|\mathbb{N}|} = |\mathbb{R}|$$

Decidibilità

$$\mathbb{N} \longrightarrow \{0, 1\}$$



$$\mathbb{1}_S(n) = \begin{cases} 1 & \text{se } n \in S \\ 0 & \text{se } n \notin S \end{cases}$$

$\mathbb{1}_S$ computabile $\iff$ $\exists$ algo che calcola $\mathbb{1}_S$ $\iff$ Prob. decidibile

$\iff$ S ricorsivo (decidibile)

$$\mathbb{I}'_S(n) = \begin{cases} 1 & n \in S \\ \perp & n \notin S \end{cases}$$

$\mathbb{I}'_S$ calcolab ($\Leftrightarrow \exists$ algo per $\mathbb{I}'_S$) $\Leftrightarrow$ Pr. semi-decidibile

$\Leftrightarrow$ S ricorsivamente enumerabile (semi-dec.)

$1/2 + 1/2 = 1$:

S ric. $\iff$ S r. enum. $\wedge$ S^c r. enum.

Corollario :

$\neg S$ ric. :

S r.e. $\wedge$ $\neg S^c$ r.e.

$\neg S$ r.e. $\wedge$ S^c r.e.

$\neg S$ r.e. $\wedge \neg S^c$ r.e.

Teo. di Rice :

$$F = \{ \text{tutte le f. computabili} \}$$
$$= \{ f_i \mid i \in \mathbb{N} \}$$

$f_i \equiv$ f. calcolata da i-esima TM

$P \subseteq F$ sottoinsieme di f. comp. che soddisfano una certa proprietà.

$$S = \{ x \mid f_x \in P \}$$

S ricorsivo (Pr. dec.) $\iff (P = \emptyset \land P = F)$

Per usare teo. di Rice:

1. Il problema riguarda MT (o form. equiv.) generiche
2. Il problema riguarda proprietà della fun. calcolata

Cercare di formalizzare il problema con una formula logica che usa solo f_M

Riduzione:

Dim. decid. di problema sconosciuto:

non lo conosco

$$P_1 \text{ sospetto dec.} \xrightarrow{\text{rid.}} P_2 \text{ noto dec.}$$

r calcolab.

$$x \in P_1 \iff r(x) \in P_2$$

Risolvi P_1 (x) :

$y \leftarrow r(x)$ // r calc.

sol $\leftarrow$ Risolvi P_2 (y) // Risolvi P_2 esiste perché P_2 dec.

return sol

es.:

P_1 : data una generica gramm. reg. G, dire se $L(G) = \emptyset$

P_2 : dato un automa FSA A, dire se $L(A) = \emptyset$ noto dec.

$r(G)$: ritorna algoritmicamente FSA A corrispondente $(L(A) = L(G))$

Risolvi $P_1(G)$:

$A \leftarrow r(G)$

$sol \leftarrow$ Risolvi $P_2(A)$ $\Rightarrow$ P_1 dec.

return sol

Rid. per indec. : P_2 sconosciuto sospetto indec.

$$P_1 \text{ noto } \underline{\text{INDEC}}. \xrightarrow{\quad r \quad} P_2 \text{ sospetto indec.}$$

Per assurdo, sia P_2 dec.
Allora $\exists$ RisolviP_2.

Posso scrivere :

```
Risolvi P_1 (x) :
    y <- r(x)
    sol <- Risolvi P_2 (y)
    return sol
```

assurdo

$\Rightarrow$ P_2 indec.

es.: P_2 = Problema: stabilire se un generico programma A accede a var. non inizializzate.

$$P_1 = \text{Halting Problem (noto indec.)} \xrightarrow{\ r\ } P_2$$

Risolvi HP(A: generico programma):

$A' \leftarrow r(A)$

sol $\leftarrow$ Risolvi $P_2(A')$

return sol

assurdo

$r(A)$:

$A' = \{ A;$

$y = x;$ // var. non iniz. $\}$

return A'

Diagonalizzazione: dimostra indec. ragionando per assurdo

"Questa frase è falsa"

Lista di cose da fare:

1. Capire se il prob. è chiuso. ((Capire qual è l'input del pr.)
2. Provare Rice (se si può applicare)
3. Ragionare sul problema
   (Riduzione, trovare algo., diagonalizzazione)

es.: Problema: sapere se una generica $f : \mathbb{N} \to \mathbb{N}$ calcolabile questa f. soddisfa:

$$\forall x \left(\left(x < 0 \Rightarrow f(x) \neq 5 \right) \land \right.$$

$$\left. \left(x \geq 0 \Rightarrow \left(f(x) > 37 \lor \left(f(x) \neq \bot \Rightarrow \left(f(x) < 100 \right) \right) \right) \right) \right)$$

INPUT: f $\longrightarrow$ non è chiuso

Si può applicare Rice.

$$\forall x \left(\left(\cancel{(x < 0 \Rightarrow f(x) \neq 5) \wedge} \right. \right.$$

$$\left. \left. \left(\cancel{x > 0 \Rightarrow} \left(f(x) > 37 \vee \left(f(x) \neq \perp \Rightarrow \left(f(x) < 100 \right) \right) \right) \right) \right) \right)$$

$$\forall x \left(f(x) > 37 \vee \left(f(x) \neq \perp \Rightarrow \left(f(x) < 100 \right) \right) \right)$$

CASO 1 : $f(x) = \perp$

$$\forall x \left(f(x) > 37 \vee \top \right) = \top$$

CASO 2 : $f(x) \neq \perp$

$$\forall x \left(f(x) > 37 \vee f(x) < 100 \right) = \top$$

$$P = F = \{ f_x \} = \{ \text{tutte le f. calc.} \}$$

$$\Downarrow$$

Dec. per Rice

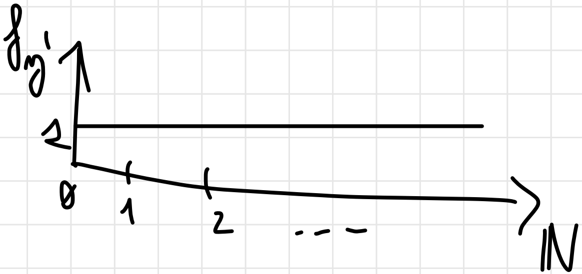
Prob: Sia data in input una fun. calcolabile f_y si stabilisca se Dominio di $f_y \equiv 2\mathbb{N}$ (insieme dei pari)

$$\forall x \left(\underbrace{f_y(x) \neq \perp}_{\text{fun. def. in } x} \iff \underbrace{\exists z \, (x = 2z)}_{x \text{ pari}} \right)$$

INPUT: f_y

Si può usare Rice? Sì

$$f_{y'}(x) = 1 \quad \forall x \in \mathbb{N} \qquad f_{y'} \notin P$$



$$f_{y''}(x) = \begin{cases} 1 & \text{se } x \text{ pari} \\ \bot & \text{altrimenti} \end{cases} \qquad f_{y''} \in P$$

$$(P \neq \emptyset \wedge P \neq F) \Rightarrow \text{indec. per Rice}$$