Insiemi produttivi

Un insieme I 1è produttivo se e solo se $\bar{K} \preceq_f I \quad (K \preceq \bar{I})$

$$x \in K \iff f(x) \notin I$$

Questo non dice nulla su $\bar{I}$

$\bar{I}$ è produttivo se e solo se $K \preceq_f I \quad (\bar{K} \preceq_f \bar{I})$

Quindi i due casi possibili sono i seguenti:

$$I \begin{cases} \to K \preceq_f I \ \wedge\ I \in RE & (I \text{ creativo}, \ I \text{ produttivo}) \\ \text{o} \\ \to K \not\preceq_f I \ \wedge\ \bar{K} \preceq_f I & (I \text{ e } I \text{ produttivi}) \end{cases}$$

1. $A = \{x \mid W_x = \emptyset\}$

A è l'insieme di tutti i programmi che divergono su tutti gli input.
Bisogna verificare che:

$$y \in A \iff \forall z.\ \varphi_y(z)\uparrow$$

$$\bar{A} = \{x \mid W_x \neq \emptyset\} \Rightarrow y \in \bar{A} \iff \exists z.\ \varphi_y(z)\downarrow$$

Questo insieme è creativo. Bisogna dimostrare:

- $\bar{A} \in RE$
- $\bar{K} \preceq_f A \equiv K \preceq_f \bar{A} \iff K \preceq_f \{x \mid W_x \neq \emptyset\}$

Quindi siccome il complemento di A è creativo, A è produttivo.

2. $B = \{x \mid W_x \neq \mathbb{N}\}$

Questo è l'insieme dei programmi che non sono totali:

$$y \in B \iff \exists z.\ \varphi_y(z)\uparrow$$

La divergenza non è decidibile, quindi questo insieme è intuitivamente non RE.

$$\bar{B} = \{x \mid W_x = \mathbb{N}\}$$

$$y \in \bar{B} \iff \forall z.\ \varphi_y(z)\downarrow$$

Anche questo non è RE perchè bisogna provare ogni input.

Bisogna dimostrare entrambi gli insiemi:

- $B: \quad K \preceq_f \bar{B} \quad (\bar{K} \preceq B)$

$$\psi(x, y) = \begin{cases} 1 & x \in K \\ \uparrow & \text{altrimenti} \end{cases}$$

La funzione psi è parziale ricorsiva banalmente perchè il test di terminazione è l'appartenenza ad un insieme RE (Se la funzione fosse stata più complessa sarebbe servito mostrare lo pseudocodice).
Si può quindi applicare smn:

$\overset{smn}{\Rightarrow} \exists g$ totale ricorsiva . $\psi(x,y) = \varphi_{g(x)}(y)$

- $x \in K \overset{\text{def } \psi}{\Rightarrow} \forall y . \psi(x,y)\downarrow$

$\overset{smn}{\Rightarrow} \forall y\ \varphi_{g(x)}(y)\downarrow$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} W_{g(x)} = \mathbb{N} \underset{\text{def } B}{\Rightarrow} g(x) \notin B \equiv g(x) \in \overline{B}$

- $x \notin K \overset{\text{def } \psi}{\Rightarrow} \forall y . \psi(x,y)\uparrow$

$\overset{smn}{\Rightarrow} \forall y\ \varphi_{g(x)}(y)\uparrow$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} W_{g(x)} = \emptyset \underset{\text{def } B}{\Rightarrow} g(x) \notin \overline{B} \equiv g(x) \in B$

Questo dimostra che B è produttivo

- $\overline{B}$: $K \leq_F B \quad (\overline{K} \leq \overline{B})$

Dobbiamo creare una funzione che soddisfa: $x \in K \Leftrightarrow g(x) \in B$ quindi:

$x \notin K \Leftrightarrow \varphi_x(x)\uparrow \Leftrightarrow \forall n.\ \varphi_x(x)$ non termina in n passi

test non decidibile

stesso test in cui la decidibilità è separata dal test

$$\psi(x,y) = \begin{cases} 1 & x \in K \text{ non deciso in meno di y passi} \\ \uparrow & \text{altrimenti} \end{cases}$$

In questo modo si lega x appartenente a K (che non deve essere mai verificato) con il test sull'input.
Questa funzione è parziale ricorsiva (ma anche totale). L'algoritmo è il seguente:

```
input(x,y)
costruisci phi_x // Esiste una procedura effettiva ricorsiva

for z = 0 to y {
  esegui il prossimo passo di phi_x(y)
  if phi_x(x) ha terminato {
    while true { x = x }
  }
}
return 1
```

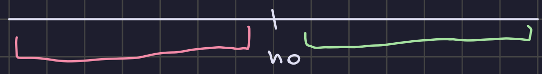
Questo algoritmo termina esattamente sugli input che rendono vera la condizione di terminazione, quindi psi è parziale ricorsiva e si può applicare smn:

$\exists g$ totale ricorsiva . $\psi(x,y) = \varphi_{g(x)}(y)$

Dimostriamo che questa funzione permette di avere la riduzione funzionale che ci interessa:

- $x \in K \overset{\text{def } K}{\Rightarrow} \varphi_x(x)\downarrow$

$\overset{\text{def terminazione}}{\Rightarrow} \quad \exists n_0 \,.\, \varphi_x(x)$ termina in n_0 passi



$y \in [0, n_0]$

phi_x(x) non termina in meno di y passi, quindi la condizione di terminazione è VERA

se y ≥ n_0 phi_x(x) ha terminato in meno di y passi, quindi la condizione di terminazione è FALSA

$\overset{\text{def } \psi}{\Rightarrow} \psi(x,y)\downarrow \iff y \in [0, n_0 - 1]$

$\overset{\text{smn}}{\Rightarrow} \varphi_{g(x)}(y)\downarrow \iff y \in [0, n_0 - 1]$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} W_{g(x)} = [0, n_0 - 1] \neq \mathbb{N}$

$\overset{\text{def } B}{\Rightarrow} g(x) \in B$

- $x \notin K \overset{\text{def } K}{\Rightarrow} \varphi_x(x)\uparrow$

$\overset{\text{def divergenza}}{\Rightarrow} \quad \forall n \,.\, \varphi_x(x)$ non termina in meno di n passi

$\overset{\text{def } \psi}{\Rightarrow} \forall y \,.\, \psi(x,y)\downarrow$

$\overset{\text{smn}}{\Rightarrow} \forall y \,.\, \varphi_{g(x)}(y)\downarrow$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} W_{g(x)} = \mathbb{N}$

$\overset{\text{def } B}{\Rightarrow} g(x) \notin B$

Abbiamo dimostrato che il complemento di B è produttivo

3. $C = \{ x \mid |range(\varphi_x)| \underbrace{< \omega}_{\text{finito}} \}$ Dimostriamo $\bar{K} \leq_F C$

In C troviamo tutti i programmi che hanno un insieme di output finito

$\bar{C} = \{ x \mid |range(\varphi_x)| = \omega \}$ Dimostriamo $\bar{K} \leq_F \bar{C}$

Nel complemento ottieniamo i programmi con range infinito

- $\bar{K} \leq_F C \quad (K \leq \bar{C})$

Se x appartiene a K vogliamo costruire una funzione con range finito

$$\psi(x,y) = \begin{cases} 1 & x \in K \\ \uparrow & \text{altrimenti} \end{cases}$$

Questa funzione non va bene perchè se x appartiene a K il range è finito $x \in K$ $\quad range(y) = \{1\}$ finito

Questa funzione non va bene perchè se x NON appartiene a K il range è finito $x \in K$ $\quad range(y) = \emptyset$ finito

Ciò che vogliamo è rendere il range infinito quando x appartiene a K, quindi la funzione corretta è:

$$\psi(x,y) = \begin{cases} y & x \in K \quad (x \in K \quad range(y) = \mathbb{N} \text{ infinito}) \\ \uparrow & \text{altrimenti} \quad (x \in K \quad range(y) = \emptyset \text{ finito}) \end{cases}$$

Questa funzione è parziale ricorsiva (completare). Quindi si può applicare smn:

$\overset{smn}{\Rightarrow} \exists g$ totale ricorsiva , $\psi(x,y) = \varphi_{g(x)}(y)$

- $x \in K \overset{def\ \psi}{\Rightarrow} \forall y .\ \psi(x,y) = y$

$\overset{smn}{\Rightarrow} \forall y\ \varphi_{g(x)}(y) = y$

$\overset{def\ range}{\Rightarrow} range(\varphi_{g(x)}) = \mathbb{N} \Rightarrow |range(\varphi_{g(x)})| = \omega$

$\overset{def\ C}{\Rightarrow} g(x) \notin C$

- $x \notin K \overset{def\ \psi}{\Rightarrow} \forall y .\ \psi(x,y) \uparrow$

$\overset{smn}{\Rightarrow} \forall y\ \varphi_{g(x)}(y) \uparrow$

$\overset{def\ range}{\Rightarrow} range(\varphi_{g(x)}) = \emptyset \Rightarrow |range(\varphi_{g(x)})| < \omega$

$\overset{def\ C}{\Rightarrow} g(x) \in C$

Abbiamo dimostrato che C è produttivo

- $\bar{K} \leq_f \bar{C}$ $\quad (K \leq C)$

Se $x \in K$ allora $g(x) \in C$ e vogliamo un range finito.

Se $x \notin K$ allora $g(x) \notin C$ e vogliamo un range infinito.

$$\psi(x,y) = \begin{cases} y & \text{se } x \in K \text{ non è deciso in meno di y passi} \\ \uparrow & \text{altrimenti} \end{cases}$$

Questa funzione è parziale ricorsiva (completare). Quindi si può applicare smn:

$\overset{smn}{\Rightarrow} \exists g$ totale ricorsiva . $\psi(x,y) = \varphi_{g(x)}(y)$

- $x \in K \overset{\text{def } K}{\Rightarrow} \varphi_x(x)\downarrow$

$\overset{\text{def} \downarrow}{\Rightarrow} \exists n_0 . \varphi_x(x)$ termina in n_0 passi

$\overset{\text{def } \psi}{\Rightarrow} \psi(x,y)\downarrow \iff y \in [0, n_0-1]$

$\overset{smn}{\Rightarrow} \varphi_{g(x)}(y)\downarrow \iff y \in [0, n_0-1]$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} |W_{g(x)}| = |[0, n_0-1]| < \omega$

$\overset{\text{def } C}{\Rightarrow} g(x) \in C$

- $x \notin K \overset{\text{def } K}{\Rightarrow} \varphi_x(x)\uparrow$

$\overset{\text{def} \downarrow}{\Rightarrow} \forall n \in \mathbb{N} . \varphi_x(x)$ non termina in n passi

$\overset{\text{def } \psi}{\Rightarrow} \forall y . \psi(x,y)\downarrow$

$\overset{smn}{\Rightarrow} \forall y \ \varphi_{g(x)}(y)\downarrow$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} |W_{g(x)}| = |\mathbb{N}| = \omega$

$\overset{\text{def } C}{\Rightarrow} g(x) \notin C$

Abbiamo dimostrato che il complemento di C è produttivo

4. $D = \{x \mid W_x = 2\mathbb{N}\} = \{x \mid W_x \subseteq 2\mathbb{N} \wedge W_x \supseteq \mathbb{N}\}$

$= \{x \mid (y \in W_x \Rightarrow y \in 2\mathbb{N}) \wedge (y \in 2\mathbb{N} \Rightarrow y \in W_x)\}$

$= \{x \mid (\varphi_x(y)\downarrow \Rightarrow y \in 2\mathbb{N}) \wedge (y \in 2\mathbb{N} \Rightarrow \varphi_x(y)\downarrow)\}$

D è l'insieme di tutti i programmi che terminano su soltanto input pari. Quindi non appartengono a D i programmi con almeno un input dispari:

$\forall y \in W_x$. y deve essere pari

- $\bar{K} \leq_F D \quad (K \leq_F \bar{D})$

Se $x \in K$ vogliamo $dom(\psi) \neq 2\mathbb{N}$

Potrebbe andare bene $\mathbb{N}$

Potrebbe andare bene $\emptyset$

Se $x \notin K$ vogliamo $dom(\Psi) = 2\mathbb{N}$

$$\Psi(x,y) = \begin{cases} 1 & \text{se } x \in K \ \vee \ y \in 2\mathbb{N} \\ \uparrow \end{cases}$$

Psi è parziale ricorsiva, forniamo l'algoritmo:

```
input(x,y)
costruisci phi_x
stop = 0
while !stop {
  esegui un passo di phi_x(y)
  if phi_x(y) ha terminato {
    if y pari {return 1}
  }
}
```

Questo codice è sbagliato perchè controlla y pari solo se x appartiene a k

Il codice corretto è il seguente:

```
input(x, y)
if y pari {
  return 1
} else {
  stop = false
  while !stop {
    esegui prossimo passi di phi_x(x)
    if phi_x(x) ha terminato {
      stop = true
    }
  }
}
return 1
```

Siccome questa funzione è parziale ricorsiva, si può applicare smn:

$\stackrel{smn}{\Rightarrow} \exists g$ totale ricorsiva , $\Psi(x,y) = \varphi_{g(x)}(y)$

- $x \in K \stackrel{def\ \Psi}{\Rightarrow} \forall y.\ \Psi(x,y)\downarrow$

$\stackrel{smn}{\Rightarrow} \forall y.\ \varphi_{g(x)}(y)\downarrow$

$\stackrel{def W_{g(x)}}{\Rightarrow} W_{g(x)} = \mathbb{N} \neq 2\mathbb{N}$

$\stackrel{def D}{\Rightarrow} g(x) \notin D$

- $x \notin K \stackrel{def\ \Psi}{\Rightarrow} \Psi(x,y)\downarrow \iff y \in 2\mathbb{N}$

$\stackrel{smn}{\Rightarrow} \varphi_{g(x)}(y)\downarrow \iff y \in 2\mathbb{N}$

$\stackrel{def W_{g(x)}}{\Rightarrow} W_{g(x)} = 2\mathbb{N}$

$\stackrel{def D}{\Rightarrow} g(x) \in D$

$\bar{D} = \{x \mid W_x \neq 2\mathbb{N}\}$ non RE

se $x \in K$ allora la funzione deve avere $dom(\psi) = 2\mathbb{N}$ $\notin \bar{D}$ $\nearrow \mathbb{N}$ $\searrow \emptyset$

se $x \notin K$ allora la funzione deve avere $dom(\psi) \neq 2\mathbb{N}$ $\in \bar{D}$

$$\psi(x,y) = \begin{cases} 1 & x \in K \wedge y \in 2\mathbb{N} \\ \uparrow & \text{altrimenti} \end{cases}$$

La funzione è parziale ricorsiva. Lo pseudocodice è il seguente:

```
input(x,y)
costruisci phi_x
while true {
  esegui il prossimo passo di phi_x(x)
  if phi_x(x) ha terminato && y è pari {
    return 1
  }
}
```

Siccome psi è parziale ricorsiva, si può applicare smn:

$\overset{smn}{\Rightarrow} \exists g$ totale ricorsiva , $\psi(x,y) = \varphi_{g(x)}(y)$

$- x \in K \overset{\text{def } \psi}{\Rightarrow} \psi(x,y)\downarrow \Leftrightarrow y \in 2\mathbb{N}$

$\overset{smn}{\Rightarrow} \varphi_{g(x)}(y)\downarrow \Leftrightarrow y \in 2\mathbb{N}$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} W_{g(x)} = 2\mathbb{N}$

$\overset{\text{def } D}{\Rightarrow} g(x) \in D$

$- x \notin K \overset{\text{def } \psi}{\Rightarrow} \psi(x,y)\downarrow \Leftrightarrow y \notin 2\mathbb{N}$

$\overset{smn}{\Rightarrow} \varphi_{g(x)}(y)\downarrow \Leftrightarrow y \notin 2\mathbb{N}$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} W_{g(x)} \neq 2\mathbb{N}$

$\overset{\text{def } D}{\Rightarrow} g(x) \notin D$

5. $E = \{x \mid 8\mathbb{N} \subseteq W_x \subseteq 2\mathbb{N}\} \to 8\mathbb{N} \subseteq 2\mathbb{N}$

$W_x \subseteq 2\mathbb{N}$ è sempre vero

$E \notin RE \wedge \bar{E} \notin RE$

- $\bar{K} \leq_f E$ $(K \leq_f \bar{E})$

$x \in K$ allora $W_x \not\supseteq 8\mathbb{N} \vee W_x \not\subseteq 2\mathbb{N}$ $\nearrow$ $\mathbb{N} \not\subseteq 2\mathbb{N}$ OK $\searrow$ $\emptyset \not\supseteq 8\mathbb{N}$ OK

$x \notin K$ allora $8\mathbb{N} \subseteq W_x \subseteq 2\mathbb{N}$ $\nearrow 8\mathbb{N}$ $\searrow 2\mathbb{N}$

$$\psi(x,y) = \begin{cases} 1 & x \in K \lor y \in 2\mathbb{N} \\ \uparrow & \text{altrimenti} \end{cases}$$

Completare fornendo l'algoritmo e dimostrando la riduzione

```
input(x,y)
if y è pari {
  return 1
}

costruisci phi_x
while true {
  esegui prossimo passi di phi_x(x)
  if phi_x(x) ha terminato {
    return 1
  }
}
```

Siccome psi è parziale ricorsiva, si può applicare smn:

$\overset{smn}{\Longrightarrow} \exists g$ totale ricorsiva , $\psi(x,y) = \varphi_{g(x)}(y)$

$- \; x \in K \overset{\text{def } \psi}{\Longrightarrow} \forall y.\ \psi(x,y)\downarrow$

$\overset{smn}{\Longrightarrow} \forall y.\ \varphi_{g(x)}(y)\downarrow$

$\overset{\text{def } W_{g(x)}}{\Longrightarrow} W_{g(x)} = \mathbb{N} \not\subseteq 8\mathbb{N}$

$\overset{\text{def } E}{\Longrightarrow} g(x) \notin E$

$- \; x \notin K \overset{\text{def } \psi}{\Longrightarrow} \psi(x,y)\downarrow \iff y \in 2\mathbb{N}$

$\overset{smn}{\Longrightarrow} \varphi_{g(x)}(y)\downarrow \iff y \in 2\mathbb{N}$

$\overset{\text{def } W_{g(x)}}{\Longrightarrow} W_{g(x)} = 2\mathbb{N} \supseteq 8\mathbb{N}$

$\overset{\text{def } E}{\Longrightarrow} g(x) \in E$

6. $F = \{x \mid W_x \subseteq 2\mathbb{N}\} \neq C \neq E$

Come insiemi F, C ed E sono diversi, ma la riduzione $\overline{K} \leq_c F$ funziona con la stessa psi

Completare

7. $G = \{x \mid 8\mathbb{N} \subseteq W_x\} \neq E$

se $x \in K$ allora $dom(\psi) \not\supseteq 8\mathbb{N}$ $\nearrow$ $\mathbb{N}$ non va bene

$\searrow$ $\emptyset \not\supseteq 8\mathbb{N}$ ✓

se $x \notin K$ allora $dom(\psi) \supseteq 8\mathbb{N} \to 8\mathbb{N}$ oppure $\mathbb{N}$ ✓

$$\psi(x,y) = \begin{cases} 1 & \text{se } x \in K \text{ non deciso in meno di y passi} \\ \uparrow & \text{altrimenti} \end{cases}$$

Psi è parziale ricorsiva (già vista prima), quindi si può applicare smn:

$\overset{smn}{\Rightarrow} \exists g$ totale ricorsiva . $\psi(x,y) = \varphi_{g(x)}(y)$

- $x \in K \Rightarrow \varphi_x(x)\downarrow$

$\Rightarrow \exists n_0 . \varphi_x(x)$ termina in n_0 passi

$\Rightarrow \psi(x,y)\downarrow \Leftrightarrow y \in [0, n_0 - 1]$

$\overset{smn}{\Rightarrow} \varphi_{g(x)}(y)\downarrow \Leftrightarrow y \in [0, n_0 - 1]$

$\Rightarrow W_{g(x)} = [0, n_0 - 1] \not\supseteq 8\mathbb{N}$ perchè $[0, n_0 - 1]$ è finito

$\Rightarrow g(x) \notin G$

- $x \notin K \Rightarrow \varphi_x(x)\uparrow$

$\Rightarrow \forall n . \varphi_x(x)$ non termina in n passi

$\Rightarrow \forall y . \psi(x,y)\downarrow$

$\overset{smn}{\Rightarrow} \forall y . \varphi_{g(x)}(y)\downarrow$

$\Rightarrow W_{g(x)} = \mathbb{N} \supseteq 8\mathbb{N}$

$\Rightarrow g(x) \in G$

Pensare al complemento

7. $H = \{ x \mid \varphi_x(2x+1) \neq 3x+3 \} = \left\{ x \mid \begin{cases} \varphi_x(2x+1)\downarrow \land \varphi_x(2x+1) \neq 3x+3 \\ \text{oppure} \\ \varphi_x(2x+1)\uparrow \end{cases} \right\}$

$\bar{H} = \{ x \mid \varphi_x(2x+1) = 3x+3 \}$

8. $I = \{x \mid \forall y .\ \varphi_x(y) = 4\}$

Il test è semidecidibile, ma si ha il per ogni, quindi non è RE.

$\bar{I} = \{x \mid \exists y .\ \varphi_x(y) \neq 4\}$

Il diverso, anche per un numero finito, è anche una divergenza:

$\varphi_x(y)\downarrow = 4 \ \vee \ \varphi_x(y)\uparrow$