Teoria della Computabilità

Guido Gherardi*

Agosto 2017

1 Funzioni ricorsive

1.1 Funzioni ricorsive primitive

Le funzioni ricorsive primitive costituiscono la più piccola classe di funzioni sui numeri naturali che include le funzioni iniziali ed è chiusa per composizione e ricorsione primitiva, dove:

Funzioni inziali

- la funzione 0-aria C_0^0, $C_0^0 = 0$
- la funzione di successore, $Succ : \mathbb{N} \to \mathbb{N}$, $Succ(x) = x + 1$
- per ogni $n \geq 1$ e $1 \leq i \leq n$ la funzione di proiezione $P_i^n : \mathbb{N}^n \to \mathbb{N}$, $P_i^n(x_1, ..., x_n) = x_i$

Composizione

Date $h : \mathbb{N}^m \to \mathbb{N}$ e $g_i : \mathbb{N}^n \to \mathbb{N}$, per $1 \leq i \leq m$, la funzione $f : \mathbb{N}^n \to \mathbb{N}$ è ottenuta per *composizione* dalle $g_1, ..., g_m$ ed h qualora $f(x_1, ..., x_n) = h(g_1(x_1, ..., x_n), ..., g_m(x_1, .., x_n))$ (scriviamo anche che $f = h \circ (g_1, ..., g_m)$)

Ricorsione

Date $g : \mathbb{N}^n \to \mathbb{N}$ e $h : \mathbb{N}^{n+2} \to \mathbb{N}$, la funzione $f : \mathbb{N}^{n+1} \to \mathbb{N}$ è ottenuta per *ricorsione* da g e da h qualora

$$f(x_1, ..., x_n, 0) = g(x_1, ..., x_n)$$

$$f(x_1, ..., x_n, y_{n+1}) = h(x_1, ..., x_n, y, f(y, x_1, ..., x_n))$$

(scriviamo $f := Rec(g, h)$)

Esempio 1.1

- Identità*:* $Id(x) = x = P_1^1(x)$
- Funzioni costanti 0*:*

$$C_0^{n+1}(x_1, ..., x_n, 0) := C_0^n(x_1, ..., x_n) = C_0^0$$

$$C_0^{n+1}(x_1, ..., x_n, y + 1) := P_{n+2}^{n+2}(x_1, ..., x_n, y, C_0^{n+1}(y, x_1, ..., x_n))$$

*Dipartimento FILCOM Università di Bologna

- Funzioni costanti *k:*

$$C^n_k(x_1, ..., x_n) := \underbrace{succ \circ succ... \circ succ}_{k\text{-}volte}(C^n_0(x_1, ..., x_n))$$

(nel seguito identificheremo ciascun $k \in \mathbb{N}$ con C^0_k)

- Somma*:*

$$Somma(x, 0) := x = Id(x)$$

$$Somma(x, y + 1) := (y + x) + 1 = Succ(P^3_3(x, y, Somma(y, x)))$$

- Predecessore*:*

$$Pred(0) := 0$$

$$Pred(y + 1) := y = P^2_1(y, Pred(y))$$

- Differenza troncata*:*

$$x \dot{-} 0 := x = Id(x)$$

$$x \dot{-} (y + 1) := Pred(P^3_3(x, y, x \dot{-} y)))$$

- Segno*:*

$$sg(0) := 0$$

$$sg(y + 1) := C^2_1(y, sg(y))$$

- Cosegno*:*

$$cosg := C^1_1 \dot{-} sg$$

Definizione 1.2 (iterazione) *Date $g : \mathbb{N}^n \to \mathbb{N}$ e $h : \mathbb{N}^{n+1} \to \mathbb{N}$, la funzione $f : \mathbb{N}^{n+1} \to \mathbb{N}$ è definita per* iterazione *da g ed h qualora*

$$f(x_1, ..., x_n, 0) := g(x_1, ..., x_n)$$

$$f(x_1, ..., x_n, y + 1) := h(x_1, ..., x_n, f(x_1, ..., x_n, y))$$

Proposizione 1.3 *Se $f : \mathbb{N}^{n+1} \to \mathbb{N}$ è definita per* iterazione *da $g : \mathbb{N}^n \to \mathbb{N}$ e $h : \mathbb{N}^{n+1} \to \mathbb{N}$ ricorsive primitive, allora f è ricorsiva primitiva.*

Dim.: Si può mostrare che f è definita per ricorsione come

$$f(x_1, ..., x_n, 0) := g(x_1, .., x_n)$$

$$f(x_1, ..., x_n, y + 1) := h^*(x_1, ..., x_n, y, f(x_1, ..., x_n, y))$$

dove $h^*(x_1, ..., x_n, x_{n+1}, x_{n+2}) =_{def} h(P^{n+2}_1(x_1, ..., x_{n+2}), ..., P^{n+2}_n(x_1, ..., x_{n+2}), P^{n+2}_{n+2}(x_1, ..., x_{n+2}))$ (h^* è pertanto definita come composizione di funzioni ricorsive primitive). □

Esempio 1.4 *1.* Prodotto*:*

$$x \cdot 0 := 0 = C^1_0(x)$$

$$x \cdot (y + 1) := Somma(x, x \cdot y)$$

2. Esponenziazione*:*

$$x^0 := 1 = C_1^1(x)$$
$$x^{y+1} = x \cdot x^y$$

3. Fattoriale*:*

$$0! = 1$$
$$(y+1)! = Succ(y) \cdot y!$$

Esercizio 1.5 *Ridefinisci le funzioni precedentemente definite per ricorsione tramite iterazione.*

Definizione 1.6 (operatore di minimalizzazione limitato) *Sia data una funzione $g : \mathbb{N}^{n+1} \to \mathbb{N}$. Allora la funzione $f : \mathbb{N}^{n+1} \to \mathbb{N}$,*

$$f(x_1, ..., x_n, x_{n+1}) := \min(\{x_{n+1}\} \cup \{\, i \in \mathbb{N} \mid i < x_{n+1} \;\&\; g(x_1, ..., x_n, i) = 0 \,\}$$

$$= \begin{cases} \min(\{\, i \in \mathbb{N} \mid i < x_{n+1} \;\&\; g(x_1, ..., x_n, i) = 0 \,\} & \textit{se almeno un tale } i \textit{ esiste} \\ x_{n+1} & \textit{altrimenti} \end{cases}$$

è detta essere ottenuta da g tramite μ-operatore limitato.

Lemma 1.7 *Sia $f : \mathbb{N}^{n+1} \to \mathbb{N}$ ottenuta mediante μ-operatore limitato da $g : \mathbb{N}^{n+1} \to \mathbb{N}$ ricorsiva primitiva. Allora f è ricorsiva primitiva.*

Dim.: Abbiamo che f è ottenuta per iterazione da g mediante

$$f(x_1, .., x_n, 0) := 0 = C_0^n(x_1, .., x_n)$$

$$f(x_1, ..., x_n, y{+}1) := Somma(f(x_1, ..., x_n, y), sg(g(x_1, ..., x_n, f(x_1, ..., x_n, y)))) =$$
$$= Somma(P_{n+1}^{n+1}(x_1, ..., x_n, f(x_1, ..., x_n, y)), sg(g(x_1, ..., x_n, f(x_1, ..., x_n, y))))$$

□

Definizione 1.8 (funzione di pseudocoppie di Cantor) *Definiamo la funzione $\pi : \mathbb{N}^2 \to \mathbb{N}$ di pseudocoppie di Cantor come*

$$\pi(n, m) := \frac{1}{2}(n+m)(n+m+1) + m$$

Indichiamo $\pi(m, n)$ anche come $\langle n, m \rangle$.

Teorema 1.9 *La funzione $\pi : \mathbb{N}^2 \to \mathbb{N}$ enumera iniettivamente tutte le coppie di numeri naturali, ed è ricorsiva primitiva, così come lo sono sue inverse $\pi_i : \mathbb{N} \to \mathbb{N}$, $i \in \{0, 1\}$ tali che $\pi_1\langle n, m \rangle := n$, $\pi_2\langle n, m \rangle := m$.*

Dim.: La funzione π enumera le coppie nel seguente modo:

(0,0),	0,
(1,0), (0,1)	1, 2,
(2,0), (1,1), (0,2),	3, 4, 5,
(3,0), (2,1), (1,2), (0,3)	6, 7, 8, 9
(4,0), (3,1), (2,2) (1,3), (0,4),	10, 11, 12, 13, 14,
...	...

Vediamo che π è ricorsiva primitiva:

- Nella z-esima riga si trovano quindi tutte le coppie (x, y) con $x + y = z$.
- Il numero che corrisponde al primo elemento della riga $z \in \mathbb{N}$, ovvero alla coppia $(z, 0)$, è
$$f(z) := \sum_{i=0}^{z-1}(i+1) = \sum_{i=1}^{z} i.$$
- $f : \mathbb{N} \to \mathbb{N}$ è ricorsiva primitiva, essendo definibile per ricorsione come segue:
$$f(0) = 0$$
$$f(z+1) = f(z) + (z+1) = Succ(Somma(z, f(z)))$$
- Ciascuna coppia (x, y) nella z-esima riga sarà associata al numero
$$\pi(x, y) = f(z) + y = f(x+y) + y = Somma(f(Somma(x, y)), P_2^2(x, y)),$$
quindi π è ricorsiva primitiva.
- Per induzione si può dimostrare che $f(z) = \frac{1}{2}z(z+1)$, quindi $\pi(x, y) = \frac{1}{2}(x+y)(x+y+1) + y$.

Vediamo ora che le inverse π_1 e π_2 sono ricorsive primitive:

- Sia dato $m \in \mathbb{N}$.
- Troviamo la riga a cui appartiene l'm-esima coppia. Dobbiamo cioè trovare $z \in \mathbb{N}$ tale che
$$f(z) \leq m < f(z+1).$$
Dobbiamo cioè individuare
$$h(m) := \min(\{m\} \cup \{\, i \in \mathbb{N} \mid i < m \ \& \ m < f(i+1) \,\}),$$
ovvero
$$\min(\{m\} \cup \{\, i \in \mathbb{N} \mid i < m \ \& \ g(m, i) = 0 \,\})$$
per
$$g(m, i) := cosg(f(i+1) \dot{-} m).$$
La funzione $h : \mathbb{N} \to \mathbb{N}$ è ricorsiva primitiva per il Lemma 1.7.
- Una volta trovata la riga z a cui la coppia corripondente ad m appartiene, troviamo la colonna y a cui appartiene tramite $y = m \dot{-} f(z)$.
- $m := \langle x, y \rangle$ con x tale che $x + y = z$.
- $\pi_2(m) = m \dot{-} f(z) = m \dot{-} f(h(m)) = y$, $\pi_1(m) = x = z \dot{-} y = z - \pi_2(m)$.
- $\pi_2 = id \dot{-} (f \circ h), \pi_1 = id - \pi_2$ sono ricorsive primitive perché composizione di funzioni ricorsive primitive.

$\square$

Estendiamo poi le pseudocoppie di Cantor a successioni finite di lunghezza arbitraria mediante definizione per induzione:

$$\langle n_1 \rangle := n_1$$

$$\langle n_1, ..., n_{k+1} \rangle := \langle n_1, \langle n_2 ..., n_{k+1} \rangle \rangle$$

Tramite le *pseudo-ennuple* di Cantor l'arietà di una funzione ricorsiva (primitiva) non è rilevante.

Esercizio 1.10 *La funzione* $f : \mathbb{N}^n \to \mathbb{N}$ *è definita per casi sulla base delle funzioni* $g_i : \mathbb{N}^n \to \mathbb{N}$ *e* $h_i : \mathbb{N}^n \to \mathbb{N}$ *per* $1 \leq i \leq m$ *se*

$$f(x_1, ..., x_n) := \begin{cases} h_1(x_1, ..., x_n) & \textit{se } g_1(x_1, ..., x_n) = 0 \\ \dots\dots \\ h_m(x_1, ..., x_n) & \textit{se } g_m(x_1, ..., x_n) = 0 \end{cases}$$

dove $g_1(x_1, ..., x_n) = 0$ *per esattamente un* $i \in \{1, ..., m\}$ *(scriviamo* $f := D(h_1, ..., h_m, g_1, ..., g_m)$*). Dimostrare che se* $h_1, ..., h_m, g_1, ..., g_m$ *sono ricorsive primitive, allora anche* f *lo è.*

1.2 Operatore di minimalizzazione e funzioni parziali ricorsive

Sulla base dei precedenti esempi si potrebbe essere tentati di evincere che tutte le funzioni (intuivamente) calcolabili siano ricorsive primitive. Tuttavia una semplice osservazione ci porta a comprendere come questo non sia il caso. Infatti le funzioni ricorsive primitive sono tutte *totali*, ovvero definite sull'intero dominio dei numeri naturali $\mathbb{N}$, mentre esistono funzioni calcolabili che hanno come dominio un sottoinsieme proprio dei numeri naturali, ad esempio $\sqrt{} :\subseteq \mathbb{N} \to \mathbb{N}$ che associa ad un numero $n \in \mathbb{N}$ la sua radice quadrata $\sqrt{n}$ posto che essa sia un numero naturale. Per trattare le funzioni calcolabili abbiamo quindi bisogno di estendere il concetto di funzione ricorsiva primitiva. Innanzitutto, visto che abbiamo a che fare con funzioni parziali, utilizziamo la notazione $f(x) \downarrow$ se $x \in \mathrm{dom}(f)$, ed esiste quindi un $y \in \mathbb{N}$ tale che $f(x) = y$, e $f(x) \uparrow$ se questo non è il caso.

Definizione 1.11 *Sia data una funzione* $g :\subseteq \mathbb{N}^{n+1} \to \mathbb{N}$*. Definiamo una funzione* $f :\subseteq \mathbb{N}^n \to \mathbb{N}$ *nel seguente modo:*

$$f(x_1, ..., x_n) := \begin{cases} y & \textit{se } (\exists y \in \mathbb{N}) \Big(g(x_1, ..., x_n, y) = 0 \;\&\; (\forall z < y)\, g(x_1, ..., x_n, z) \downarrow \neq 0 \Big) \\ \uparrow & \textit{altrimenti.} \end{cases}$$

Allora diciamo che f *è ottenuta per* operatore di minimalizzazione *(*μ-operatore*) da* g*, e scriviamo*

$$f(x) := (\mu y)[g(x_1, ..., x_n, y) = 0],$$

o più semplicemente $f := \mu(g)$*).*

La classe delle *funzioni (parziali) ricorsive* è la più piccola classe di funzioni che contiene le funzioni iniziali, ed è chiusa per composizione, ricorsione e minimalizzazione[1].

La funzione $\surd$ può essere definita mediante minimalizzazione. Dato $x \in \mathbb{N}$ dobbiamo trovare il minimo $y \in \mathbb{N}$ tale che $y^2 = x$ (se tale y esiste). Ciò può essere espresso mediante il μ-operatore nel seguente modo:

$$\sqrt{x} := (\mu y)\ [(y^2 \dot{-} x) + (x \dot{-} y^2) = 0].$$

In realtà esistono anche funzioni ricorsive totali che non sono ricorsive primitive. Un esempio storicamente celebre è dato dalla funzione di Ackermann.

2 Macchine di Turing

Una macchina di Turing è un tipo di macchina ideale che viene utilizzata per eseguire *computazioni*. Essa è costituita da un *hardware* e da un *software*.

L'hardware di una macchina di Turing è lo stesso per tutte le macchine ed è costituito da un nastro di lunghezza infinita e da una testina di lettura/scrittura capace di muoversi lungo tale nastro. Tale nastro è costituito da una successione infinita di celle che possono contenere un simbolo di un alfabeto determinato. Spesso si considera l'alfabeto binario $\Sigma := \{0, 1\}$.

Il software è invece l'elemento caratterizzante la singola macchina di Turing: due macchine di Turing distinte avranno due software differenti. Tale software è il *programma* registrato nella memoria della macchina. Il programma di una macchina di Turing è dato da una successione finita di quadruple del tipo:

$$qijTq' \in Q \times (\Sigma \cup \{\flat\}) \times (\Sigma \cup \{\flat\}) \times \{D, S, I\} \times Q.$$

Qui $Q := \{q_1, ..., q_n, q_0\}$ è l'insieme degli *stati interni* della macchina. In esso si distinguono lo stato iniziale q_1 e quello finale q_0. All'inizio di qualsiasi computazione la macchina di Turing si troverà nello stato iniziale q_1.

$\flat$ è un simbolo speciale, detto "*blank*", che indica una cella vuota, mentre D, S, I indicano rispettivamente "destra, sinistra, immobile".

L'istruzione $qijTq'$ viene considerata solo quando la macchina si trova nello stato q e il simbolo letto dalla testina in quel momento è proprio i, se $i \in \{0, 1\}$, oppure nessun simbolo se $i = \flat$.

La macchina di Turing è deterministica, nel senso che vi è al massimo una sola istruzione che cominci con qi, pertanto ciò che la macchina farà in tale condizione è univocamente determinato. Ciò sarà innanzitutto rimpiazzare i con j; come casi particolari, qualora $i = j$, questo corrisponde a lasciare inalterato il contenuto della cella (sia esso anche il nulla), mentre se $i \neq j = \flat$ allora il contenuto della cella viene cancellato. La testina si muoverà poi esattamente di una cella verso sinistra, verso destra, o rimarrà immobile a seconda di quale tra D, S, I il simbolo T sia. Infine la macchina entrerà nello stato q'.

Lo stato q_0 è lo stato finale. Se la macchina raggiunge tale stato, si ferma istantaneamente.

[1] Per uniformità anche le funzioni totali sono considerate esempi di funzioni parziali: infatti per definizione una funzione è parziale se il suo dominio è un sottoinsieme dell'insieme totale, ma non necessariamente un sottoinsieme proprio.

L'idea di fondo è che una macchina di Turing calcolerà una determinata funzione n-aria f. All'inizio della computazione n stringhe binarie sono pertanto scritte sul nastro (di lunghezza ≥ 1). La testina si trova sulla cella che contiene la prima cifra della prima stringa e la macchina si trova nello stato iniziale q_1. La macchina di Turing segue le istruzioni contenute nel programma. Se (i) lo stato finale q_0 viene raggiunto, (ii) sul nastro si trova (esattamente) una stringa binaria, e (iii) la testina si trova sull'ultima cifra di tale stringa, allora tale stringa rappresenta un output valido. Se le stringhe iniziali denotavano (in binario) i numeri $k_1, ..., k_n$, e la stringa finale denota il numero k, allora la macchina mappa $(k_1, ..., k_n)$ su k, cioè $f(k_1, ..., k_n) = k$. La funzione f sarà detta quindi *Turing-computabile.*

Teorema 2.1 *Le funzioni ricorsive sono Turing-computabili.*

Diamo una giustificazione *intuitiva* (!!!) di tale asserto.

La funzione iniziale C_0^0 è computabile mediante una macchina di Turing che scrive sul nastro vuoto (quindi contenente 0 argomenti iniziali) la stringa 0 (senza bisogno che la testina si muova).

La funzione successore è computabile da una macchina di Turing che scorra tutto la stringa binaria iniziale data in input spostando la testina verso destra fino all'ultima cifra. Se questa è uno 0 la macchina la sostituisce con 1. Se questa è un 1 allora la macchina deve eseguire i riporti spostando la testina verso sinistra e sostituendo ciascun 1 incontrato con 0. La prima occorrenza di 0 che la testina incontra muovendosi verso sinistra viene trasformata in 1. Se tale occorrenza non esiste, allora la testina scriverà un 1 nella cella vuota che precede la stringa numerica.

La funzione di proiezione P_i^n è banalmente computabile da una macchina che cancelli tutte le n stringhe ricevute in input eccetto l'i-esima.

La composizione $f := h \circ (g_1, ..., g_m)$ applicata a $k_1, ..., k_n$ (numeri codificati in binario) è computabile da una macchina che prima trasformi l'input $(k_1, ..., k_n)$ in $g_1(k_1, ..., k_n), ..., g_m(k_1, ..., k_n)$ scrivendo provvisoriamente tali valori (in binario) sul nastro al posto dell'input iniziale (separati da occorrenze di blank), e poi applichi la funzione h all'ennupla di tali valori. La sola stringa rimanente sul nastro al termine della computazione sarà (la codifica in binario di) $h(g_1(k_1, ..., k_n), ..., g_m(k_1, ..., k_n))$.

Per calcolare una funzione $f := Rec(g, h)$ definita per ricorsione occorre una macchina che generi $f(k_1, ..., k_n, 0)$ applicando g ad $(k_1, ..., k_n)$ tale che $f(k_1, ..., k_n, 0) = g(k_1, ..., k_n)$. Tale valore, scritto (provvisoriamente) sul nastro viene poi utilizzato dalla macchina per calcolare $f(k_1, ..., k_n, 1)$ come $h(k_1, ..., k_n, 0, f(k_1, ..., k_n, 0))$. L'operazione viene poi ripetuta fino al raggiungimento di $f(k_1, ..., k_n, m)$ per l'm desiderato.

La funzione $f := (\mu)g$ infine viene eseguita da una macchina di Turing che dati $k_1, ..., k_n$ verifichi se $g(k_1, ..., k_n, 0) = 0$, ovvero calcoli il valore di $(g(k_1, ..., k_n, 0)\dot{-}0) + (0\dot{-}g(k_1, ..., k_n, 0))$. Se tale valore è definito ed è diverso da 0, allora la macchina esegue il test analogo $g(k_1, ..., k_n, 1) = 0$, andando alla ricerca del minimo k tale che $g(k_1, ..., k_n, k) = 0$. Se tale k viene trovato, la macchina lascia alla fine sul nastro solo (la codifica binaria di) k, altrimenti, la computazione non termina.

In realtà tutti i diversi paradigmi di computabilità presentati nel 1936 (macchine di Turing, funzioni ricorsive di Kleene, λ-calcolo di Church) sono risultati essere tra loro equivalenti.

Questo fatto ha supportato la seguente tesi, centrale in Teoria della Computabilità e in Filosofia della Computabilità:

Tesi 2.2 (Tesi di Church-Turing) *Le funzioni computabili (sui numeri naturali) sono esattamente le funzioni Turing-computabili.*

Questa tesi cerca quindi di dare una caratterizzazione specifica al concetto altrimenti puramente *intuitivo* di "computabile": cosa significa essere computabile? Significa essere computabile con una macchina di Turing. La moderna Teoria della Computabilità si basa su tale tesi, e spesso viene utilizzata indirettamente per svolgere le dimostrazioni in tale teoria, sulla base del fatto che tutto ciò che suona "plausibilmente computabile" lo sia in effetti mediante una macchina di Turing, per cui si omette di scrivere programmi (che di per sé sarebbero particolamente ostici da gestire). In fondo la stessa dimostrazione intuitiva che abbiamo dato dell'enunciato del Teorema 2.1 è basata indirettamente su tale tesi.

Si noti che non ha propriamente senso chiedersi se tale tesi sia "dimostrabile", nel senso che essa si basa sull'unione di un concetto vago con uno specifico, e si tratta pertanto piuttosto, come dicevamo, di una caratterizzazione.

In queste dispense utilizzero quindi i termini "funzione ricorsiva" e "funzione computabile" (sui numeri naturali), come sinonimi.

Le conoscenze attuali, ma ancor più il buon senso ci confermano la sostenibilità di tale tesi (non vogliamo dire la "verità"), tuttavia essa potrebbe essere teoricamente falsificata qualora si riuscisse a trovare una funzione che potremmo riconoscere come computabile (ad esempio in quanto simulata da un dispositivo meccanico o da un processo fisico o naturale) e al contempo non fosse Turing-computabile.

3 Insiemi decidibili e (co-)ricorsivamente enumerabili

3.1 Insiemi decidibili

Definizione 3.1 (Insiemi decidibili) *Un insieme $M \subseteq \mathbb{N}$ è detto* decidibile *se la sua funzione caratteristica $\chi_M : \mathbb{N} \to \{0, 1\}$,*

$$\chi_M(x) := \begin{cases} 1 & se\ x \in M \\ 0 & se\ x \notin M \end{cases}$$

è ricorsiva.

In pratica, M è decidibile se possiamo stabilire quando un qualsiasi $n \in \mathbb{N}$ è un suo elemento oppure no.

Molti degli insiemi elementari studiati correntemente in aritmetica sono decibili, ad esempio $\mathbb{N}$, $\emptyset$, l'insieme dei numeri pari, quello dei numeri dispari, l'insieme dei quadrati, quello dei numeri primi, sono tutti insiemi decidibili. Ad esempio la funzione caratteristica dell'insieme dei numeri dispari è data da

$$Disp(n) = n \dot{-} [2 \cdot Pred((\mu y)[2y > n])] = n \dot{-} [2 \cdot Pred((\mu y)[cosg(2y \dot{-} n) = 0]].$$

La definizione si estende banalmente al caso $M \subseteq \mathbb{N}^n$ (quindi ai *predicati* in $\mathbb{N}$) mediante pseudo-ennuple di Cantor, per cui sono decidibili ad esempio anche le coppie di elementi identici, o quelle in cui il primo elemento è minore del secondo, e così via.

Ad esempio, la funzione caratteristica del predicato d'identità è data da

$$cosg((x \dot{-} y) + (y \dot{-} x)).$$

Proposizione 3.2 *Siano $M, N \subseteq \mathbb{N}$ decidibili. Allora sono decidibili anche*

1. $M \cap N$
2. $M \cup N$
3. $\mathbb{N} \setminus M$.

Dim.:

1. $\chi_{M \cap N} = \chi_M \cdot \chi_N$
2. $\chi_{M \cup N} = sg(\chi_M + \chi_N)$
3. $\chi_{\mathbb{N} \setminus M} = 1 \dot{-} \chi_M$.

□

Da questa proposizione (formulata per $\mathbb{N}^n$) segue la chiusura per operatori booleani dei predicati decidibili.

Tuttavia una semplice considerazione legata alla cardinalità degli insiemi mostra che la maggior parte degli insiemi *non* è decidibile. Una macchina di Turing è identificata dal suo programma, e questo è una successione finita di istruzioni su un un insieme numerabile (addirittura finito) di caratteri. Perciò l'insieme delle macchine di Turing è numerabile, e pertanto lo è anche quello delle funzioni caratteristiche computabili. Tuttavia $Card(\wp(\mathbb{N})) > Card(\mathbb{N})$.

3.2 Insiemi ricorsivamente enumerabili e co-ricorsivamente enumerabili

Definizione 3.3 (Insiemi ricorsivamente enumerabili (r.e.)) *Un insieme M è* ricorsivamente enumerabile *(*r.e.*) se esiste una funzione parziale ricorsiva $f :\subseteq \mathbb{N} \to \mathbb{N}$ tale che $M := \mathrm{dom}(f)$.*

Esempio 3.4 *1.* $M = \emptyset = \mathrm{dom}(\lambda x. \uparrow) = \mathrm{dom}(x + (\mu y)[Succ(y) = 0])$

2. $M = \mathbb{N} = \mathrm{dom}(C^1_0)$

3. $M := \{\, n \in \mathbb{N} \mid (\exists m \in \mathbb{N})\, m = \sqrt{n}\,\} = \mathrm{dom}(\lambda x.[(\mu y)[\, y^2 = x]])$

In realtà è facile vedere che i precendenti esempi sono anche insiemi decidibili. Abbiamo infatti:

Proposizione 3.5 *Un insieme decidibile è anche ricorsivamente enumerabile.*

DIM.: Sia M decidibile. Consideriamo allora la funzione $f :\subseteq \mathbb{N} \to \mathbb{N}$ tale che $f(x) := (\mu y)[C_1^1(y) \dot{-} \chi_M(x) = 0]$. □

Tuttavia esistono insiemi r.e. che non sono decidibili. Come dicevamo, gli insiemi utilizzati in artitmetica sono (solitamente) decidibili, quindi non è immediato mostrare dei concreti esempi di insiemi r.e. non decidibili. Questa è stata una scoperta della Teoria della Computabilità.

Per arrivare a questo, ritorniamo alla considerazione già fatta che le macchine di Turing sono numerabili, in quanto l'insieme dei loro programmi (in quanto liste finite di istruzioni su un insiene numerabile, addirituttura finito, di caratteri). È routine mostrare che le macchine di Turing possono essere enumerate secono una procedura meccanica di *Gödelizzazione*, per cui è possibile meccanicamente associare a ciascun programma un numero, e viceversa, risalire al programma che corrisponde ad un certo numero. A questa enumerazione $M_0, M_1, ..., M_n, ...$ delle macchine di Turing corrisponderà un'enumerazione della totalità $\varphi_0, \varphi_1, ..., \varphi_n, ...$ delle funzioni parziali ricorsive. Non si tratta ovviamente di un'enumerazione iniettiva, ovvero senza ripetizioni, questo perché data una qualsiasi macchina di Turing, vi sono infinite altre macchine di Turing che calcolano la sua stessa funzione (è sufficiente infarcire il programma con un numero arbitrario di istruzioni inutili o ridondanti).

Un celebre risultato dimostrato da Turing nel suo articolo del 1936 è l'esistenza di una **macchina di Turing universale** (**utm property**) capace di simulare qualsiasi altra macchina di Turing (compresa sé stessa!). Questa macchina prende in input una (pseudo)coppia $\langle i, n \rangle$ e simula la macchina M_i applicata sull'input n, ovvero calcola $\varphi_i(n)$ (che può anche divergere).

Al contrario, un suo celebre risultato "negativo" riguarda il **problema della fermata**: è possibile in generale sapere se la computazione della macchina M_i converge sull'input n (codificato in binario)? In altre parole, possiamo sapere per ogni $i, n \in \mathbb{N}$ se il valore di $\varphi_i(n)$ esiste?

Teorema 3.6 *L'insieme* Halting Set $H := \{ \langle i, n \rangle \mid \varphi_i(n) \downarrow \}$ *è r.e. ma non è decidibile.*

DIM.: Si può vedere che H è r.e. utilizzando una macchina universale. Sia dato quindi $\langle i, n \rangle$. Utilizziamo una maccina di Turing universale M per simulare M_i su n. Se $\varphi_i(n) \downarrow$, la macchina restituisce il valore di $\varphi_i(n)$, altrimenti ovviamente non restituisce nulla. Per cui $\langle i, n \rangle \in H$ sse $U(\langle n, n \rangle) \downarrow$, dove U è la funzione calcolata da M.

Per vedere che l'insieme non è decidibile, utilizziamo il metodo di diagonalizzazione alla Cantor. Supponiamo quindi per assurdo che H sia decidibile, ovvero che $\chi_H : \mathbb{N} \to \{0, 1\}$ sia ricorsiva. Consideriamo allora la funzione $\varphi : \mathbb{N} \to \mathbb{N}$ tale che

$$\varphi(x) := \begin{cases} \varphi_x(x) + 1 & \text{se } \varphi_x(x) \downarrow \text{, cioè } \chi_H(\langle x, x \rangle) = 1 \\ 0 & \text{se } \varphi_x(x) \uparrow \text{, cioè } \chi_H(\langle x, x \rangle) = 0 \end{cases}$$

Questa funzione è ricorsiva (immediato via Tesi di Church-Turing, alternativamente si adatti banalmente la soluzione dell'Esercizio 1.10 al caso delle funzioni ricorsive totali). Esiste pertanto un $k \in \mathbb{N}$ tale che $\varphi = \varphi_k$. Abbiamo allora $\varphi_k(k) = \varphi(k) \neq \varphi_k(k)$. Contraddizione. $\square$

Notazione 3.7 *Per ogni $e \in \mathbb{N}$ poniamo*

$$W_e =_{def} \{\, n \in \mathbb{N} \mid n \in \mathrm{dom}(\varphi_e) \,\}$$

$W_{e,s} =_{def} \{ n \in \mathbb{N} \mid M_e$ *converge in meno di s passi di computazione sull'input n (producendo $\varphi_e(n)$* $\}$

Abbiamo ora la seguente fondamentale caratterizzazione degli insiemi r.e.:

Teorema 3.8 (Listing Theorem) *$M \subseteq \mathbb{N}$ è r.e. sse $M = \emptyset$ o è listato da una funzione ricorsiva totale φ, ossia $M = \{\varphi(0), \varphi(1), ..., \varphi(n), ...\} = \mathrm{cod}(\varphi)$.*

Dim.: $\Leftarrow$) Sia $M := \mathrm{cod}(\varphi)$ per φ totale ricorsiva. Definiamo ora $\psi :\subseteq \mathbb{N} \to \mathbb{N}$ tale che

$$\psi(x) := (\mu y)[\varphi(y) = x].$$

Allora ψ è parziale ricorsiva e $\mathrm{dom}(\psi) = \{\, n \in \mathbb{N} \mid (\exists y)\ \varphi(y) = n \,\} = M$.

$\Rightarrow$) Sia $M \neq \emptyset$ r.e., più precisamente sia $M := W_e$. Giacché $M \neq \emptyset$ esiste un $a \in M$. Definiamo ora φ totale ricorsiva nel seguente modo:

$$\varphi\langle x, s\rangle := \begin{cases} x & \text{se } x \in W_{e,s+1} \\ a & \text{altrimenti.} \end{cases}$$

Abbiamo che $\mathrm{cod}(\varphi) \subseteq M$, giacché $n \in \mathrm{cod}(\varphi)$ implica $n = a \in M$ o $n \in W_{e,s} \subseteq W_e = M$ per un certo s. Ma d'altronde $M \subseteq \mathrm{cod}(\varphi)$ perché se $n \in M = W_e$ allora esiste un s tale che $n \in W_{e,s}$ e quindi $\varphi(\langle n, s\rangle) = n$. $\square$

Sebbene quindi la classe degli insiemi r.e. contenga più elementi della classe degli insiemi, una considerazione analoga a quella degli insiemi decidibili mostra che la maggior parte degli insiemi non è neppure ricorsivamente enumerabile.

Un esempio concreto di insieme che non è r.e. è il complementare dell'Halting Set.

Definizione 3.9 (Insiemi co-r.e.) *Un insieme M è detto* co-ricorsivamente enumerabile *(*co-r.e*) se è il complementare di un insieme r.e.*

$\mathbb{N} \setminus H$ è quindi un insieme co-r.e. Dobbiamo mostrare che non è r.e.

Definizione 3.10 (Insiemi ricorsivi) *Un insieme $M \subseteq \mathbb{N}$ è detto* ricorsivo *sse M è sia r.e. che co-r.e.*

Teorema 3.11 *$M \subseteq \mathbb{N}$ è decidibile sse M è ricorsivo.*

Dim.: $\Rightarrow$) Mostriamo che M è sia r.e. che co-r.e., ovvero che sia M che $\mathbb{N} \setminus M$ sono r.e. Abbiamo già visto che M decidibile implica M r.e. (Proposizione 3.5). In maniera analoga si dimostra che $\mathbb{N} \setminus M$ è r.e. mediante la Proposizione 3.2.3.
$\Leftarrow$) Usare il Teorema 3.8 applicato in parallelo sia ad M che a $\mathbb{N} \setminus M$ in quanto insiemi r.e.. $\square$

Proposizione 3.12 $\mathbb{N} \setminus H$ *non è ricorsivamente enumerabile.*

Dim.: Segue dal Teorema 3.11, considerando che H è r.e. ma non è decidibile per il Teorema 3.6. $\square$

4 Spazi rappresentati

I numeri naturali sono "oggetti finiti", nel senso che una successione finita di bit è sufficiente a codificarli (es. sviluppo binario, sviluppo decimale). Ovviamente anche i numeri interi sono "oggetti finiti", così come i numeri razionali: questi ultimi possono essere codificati come coppie di numeri naturali (numeratore e denominatore).

Al contrario i numeri reali, a causa della presenza dei numeri irrazionali, hanno bisogno di successioni infinite di bit per essere univocamente determinati (questo vale quindi anche per i numeri dei tipi precedenti quando immersi all'interno del campo dei numeri reali).

Giacché le successioni richieste sono di lunghezza infinita, l'unica cosa a cui possiamo affidarci nell'ambito di una computazione concreta sono i segmenti iniziali finiti di tali stringhe. Siccome richiedere alla macchina il raggiungimento di uno stato finale d'arresto in questo caso non avrebbe senso, anzi la computazione in linea di principio non deve mai arrestarsi, richiediamo almeno che i segmenti iniziali della successione infinita prodotti dalla macchina siano affidabili. Per questa ragione, vorremmo che essi non vengano mai corretti. Tuttavia, una macchina di Turing può avere bisogno di eseguire dei calcoli provvisori durante la computazione, i cui risultati vengano poi superati e quindi cancellati. Per questa ragione è naturale considerare in questo caso una macchina di Turing con più nastri: uno di input di sola lettura, sulla quale la macchina riceve una successione infinita non modificabile come input, uno di lavoro, sulla quale la macchina possa eseguire tutte le operazioni ammesse per una macchina di Turing tradizionale (compresa la modifica e la cancellazione di cifre), ed infine un nastro di sola scrittura dove la macchina scriva man mano l'output estendendolo all'infinito e dove nessuna correzione possa mai essere eseguita (per questa ragione si può supporre, senza perdita di generalità, che le testine sui nastri di input e di output possano solamente muoversi verso destra). Le macchine di questo tipo sono dette TTE (Type-2 Theory of Effectivity).

Si osservi che questo modello di computazione è realistico: una computazione di questo tipo può essere ridotta (e nei fatti così è) ad una successione di computazioni con macchine di Turing tradizionali (Type-1): se una macchina TTE riceve in input una successione infinita p e produce come output un'altra successione infinita q, vi sarà una macchina di Turing tradizionale che trasforma segmenti iniziali w dell'input p in segmenti iniziali v dell'output q.

Proprio come nel caso delle funzioni con oggetti finiti, anche nel caso delle funzioni con oggetti infiniti l'arietà non è molto importante, grazie all'estensione delle pseudo-ennuple di Cantor al caso delle successioni infinite: per $p_0, p_1, p_2, ...$ successioni infinite di bit, dove $p_i := p_i(0)p_i(1)p_i(2)....$ per ogni $i \in \mathbb{N}$, abbiamo

$$\langle p_1, ..., p_n \rangle := p_1(0)...p_n(0)p_1(1)...p_n(1)p_1(2)...p_n(2)...$$

$$\langle p_0, p_1, p_2, ... \rangle \langle i, j \rangle := p_i(j)$$

Un esempio ben conosciuto di codifica dei numeri reali è dato dagli sviluppi binari o decimali *infiniti*.

Tuttavia, sebbene largamente conosciuti, gli sviluppi binari o decimali o in qualsiasi altra base dei numeri reali non costituiscono un metodo soddisfacente di codifica dal punto di vista della teoria della computabilità. Vediamo il caso degli sviluppi decimali:

> Nessuna macchina TTE moltiplica frazioni decimali infinite per 3.
>
> Assumiamo che qualche macchina M di tipo 2 computi la funzione reale $x \mapsto 3 \cdot x$. Allora deve operare correttamente sul nome $0,333...$ di $\frac{1}{3}$ come input. L'output sarà $0,999...$ o $1,000....$ Consideriamo il primo caso. Sia k il numero di passi che M esegue prima di scrivere il prefisso "0," sul nastro di output. Durante questo periodo la macchina legge non più di un numero finito di simboli, diciamo il prefisso $0,w$, sul nastro di input. Consideriamo ora la successione di input $0,w999...$ che rappresenta un numero reale $x > \frac{1}{3}$. Allora anche su tale input la macchina M comincerà l'output con "0,". Ma poiché $3 \cdot x > 1$, M non lavora correttamente su quest'input (contraddizione). Nel secondo caso l'argomento è simile. [Wei00, p. 18]

L'argomento si può adattare anche agli sviluppi nelle altre basi, come la binaria.

Esercizio 4.1 *Dimostrare che la somma non è computabile sui numeri reali codificati in binario o in base 10.*

Come osserva quindi giustamente Weihrauch:

> Un concetto di computabilità per i numeri reali, rispetto al quale neppure la somma e la moltiplicanzione risultino funzioni computabili, non è molto utile e sarebbe difficilmente accettata dai matematici numerici o dagli studiosi di teoria della computabilità. [Weih18]

Si rende quindi necessaria una teoria che sappia determinare buoni metodi di codifica per le strutture matematiche con "oggetti infiniti".

4.1 Elementi di Topologia

Definizione 4.2 (Spazio topologico) *Uno* spazio topologico *è una coppia* (X, τ_X) *dove* X *è un insieme e* $\tau_X \subseteq \wp(X)$ *è una* topologia *su* X*, ovvero una collezione di insiemi di* X *tale che:*

1. τ_X *è chiusa per unioni arbitrarie, ovvero se* $O_i \in \tau_X$ *per ogni* $i \in I$*, allora* $\bigcup_{i \in \mathbb{N}} O_i \in \tau_X$
2. τ_X *è chiusa per intersezioni finite, ovvero se* $O_i \in \tau_X$ *per ogni* $i \in I$ *con* I *finito, allora* $\bigcap_{i \in \mathbb{N}} O_i \in \tau_X$

Gli elementi di τ_X *sono detti* insiemi aperti.
Un sottoinsieme $A \subseteq X$ *è detto* chiuso *se è il complementare di qualche insieme aperto.*

Osservazione 4.3 *Dalla condizione 1 segue che* $\emptyset \in \tau_X$*.*
Dalla condizione 2 segue che $X \in \tau_X$*.*
Pertanto gli insiemi $\emptyset$ *e* X *sono sia aperti che chiusi.*

Esempio 4.4 *Dato* X*, la topologia più povera su* X *è data da* $\{\emptyset, X\}$*.*

Definizione 4.5 (Base e sottobase) *Una* base *per uno spazio topologico* (X, τ_X) *è data da una collezione* $\{O_i\}_{i \in I}$ *di sottoinsiemi di* τ_X *tali che per ogni* $O \in \tau_X$ *esiste* $J \subseteq I$ *tale che* $O = \bigcup_{i \in J} O_i$.

Una base per τ_X *è detta* generare τ_X.

Una sottobase *per uno spazio topologico* (X, τ_X) *è data da una collezione* $\{O_i\}_{i \in I}$ *di sottoinsiemi di* τ_X *tale che la chiusura per intersezioni finite in* $\{O_i\}_{i \in I}$ *è una base per* τ_X.

> **Esempio 4.6** *Dato* X *la topologia più ricca su* X *ha come base* $\{\, \{x\} \mid x \in X \,\}$*; quest'ultima è detta* topologia discreta.
>
> *La topologia ordinaria sui numeri reali ha come base la collezione di tutti gli intervalli aperti in* $\mathbb{R}$*, ovvero la collezione di tutti gli insiemi del tipo* $]a, b[:= \{\, c \in \mathbb{R} \mid a < c < b \,\}$ *per* $a \leq b \in \mathbb{R}$.
>
> *Gli intervalli chiusi* $[a, b] := \{\, c \in \mathbb{R} \mid a \leq c \leq b \,\}$ *per* $a, b \in \mathbb{R}$ *sono tutti insiemi chiusi rispetto a tale topologia.*

Spesso per semplicità identificheremo uno spazio topologico (X, τ_X) con l'insieme X, lasciando per implicita τ_X. Coerentemente, una funzione $f : (X, \tau_X) \to (Y, \tau_Y)$ sarà una funzione $f : X \to Y$ sugli insiemi corrispondenti.

- **Definizione 4.7 (Funzioni continue)** *Una funzione* $f : X \to Y$ *tra spazi topologici* X, Y *è* continua *se per ogni* $x \in X$ *e per ogni aperto* $O \subseteq Y$ *tale che* $f(x) \in O$ *esiste un aperto* $U \subseteq X$ *tale che* $x \in U$ *e* $f(U) \subseteq O$.

Notazione 4.8 *Con* 0^n *(risp.* 1^n*) indichiamo una successione di* n *occorrenze di 0 (risp. 1) per* $n \in \mathbb{N}$.

Con $\{0, 1\}^{\mathbb{N}}$ *indichiamo l'insieme delle successioni* binarie *infinite* $p = p(0)p(1)p(2)...$*, mentre con* $\{0, 1\}^{<\mathbb{N}}$ *quello delle succesioni* binarie *finite (parole)* $w = w(0)...w(|w|-1)$*, dove* $|w|$ *indica la lunghezza di* w.

La successione costantemente 0 (risp. 1) verrà indicata con $0^{\mathbb{N}}$ *(risp.* $1^{\mathbb{N}}$*).*

Data una parola $w \in \{0, 1\}^{<\mathbb{N}}$ *ed una successione* $p \in \{0, 1\}^{\mathbb{N}}$*, scriviamo* $w \sqsubseteq p$ *quando* w *è una segmento iniziale di* p*. In questo caso scriviamo anche che* $p = wp'$ *per* $p' \in \{0, 1\}^{\mathbb{N}}$ *tale che* $p'(n) = p(n + |w|)$ *per ogni* $n \in \mathbb{N}$.

Scriviamo anche $w\{0, 1\}^{\mathbb{N}}$ *per indicare l'insieme* $\left\{\, p \in \{0, 1\}^{\mathbb{N}} \mid w \sqsubseteq p \,\right\}$.

Queste notazioni sono immmediatamente generalizzabili a qualsiasi altro alfabeto Σ*, ottenendo in particolare* $\Sigma^{\mathbb{N}}, \Sigma^{<\mathbb{N}}, w\Sigma^{\mathbb{N}}$.

L'insieme $\{0, 1\}^{\mathbb{N}}$ costituisce lo *spazio di Cantor* quando è associato alla topologia che ha come base la collezione degli insiemi del tipo $w\{0, 1\}^{\mathbb{N}}$.[2]

L'usuale concetto di continuità topologica si applica allo spazio di Cantor nel seguente modo: una funzione $F : \{0, 1\}^{\mathbb{N}} \to \{0, 1\}^{\mathbb{N}}$ è continua quando per ogni $p \in \{0, 1\}^{\mathbb{N}}$ e per ogni $v \in \{0, 1\}^{<\mathbb{N}}$ tale che $v \sqsubseteq F(p)$, esiste un $w \in \{0, 1\}^{<\mathbb{N}}$ per cui $w \sqsubseteq p$ e $F(w\{0, 1\}^{\mathbb{N}}) \subseteq v\{0, 1\}^{\mathbb{N}}$.

Intuitivamente, la successione $F(p)$ è univocamente determinata dai segmenti iniziali di p.

Proposizione 4.9 *Sia data una funzione* $f : X \to Y$ *tra spazi topologici. Allora le seguenti condizioni sono equivalenti:*

[2] In poche parole, lo spazio di Cantor possiede la *topologia prodotto* rispetto alla topologia discreta su $\{0, 1\}$.

- *f è continua;*
- *per ogni aperto $O \subseteq Y$, $f^{-1}(O)$ è aperto in X.*

DIM.: Esercizio. □

Il concetto di continuità della Definizione 4.7, e quindi anche il Teorema 4.9, si estendono al caso delle funzioni parziali $f :\subseteq X \to Y$ mediante la topologia relativa su $\mathrm{dom}(f)$.

Definizione 4.10 (Topologia relativa) *Dato uno spazio topologico X ed un insieme $X' \subseteq X$, la* topologia relativa su X' indotta da X *è data dall'insieme:*

$$\tau_{X'} := \{\, O \cap X' \mid O \text{ è aperto in } X \,\}$$

Esercizio 4.11 *Definisci la topologia relativa su $[0,1]$ indotta da $\mathbb{R}$.*

Pertanto una funzione parziale $f :\subseteq X \to Y$ sarà continua se per ogni $O \subseteq Y$ si ha che $f^{-1}(O) \cap \mathrm{dom}(f)$ è aperto in $\mathrm{dom}(f)$, dove ovviamente $f^{-1}(O) = \{\, x \in X \mid f(x)\downarrow \;\&\; f(x) \in O \,\}$.

4.2 Rappresentazioni e spazi rappresentati

Diamo ora la definizione basilare della teoria, quella di *spazio rappresentato*, che intuitivamente consiste in un insieme e in una codifica degli oggetti di questo insieme mediante successioni infinite di simboli da un determinato alfabeto, in modo che ciascuna successione infinita codifichi al massimo un oggetto dell'insieme. Come alfabeto sceglieremo per comodità il *minimo alfabeto utile* $\{0,1\}$. Pertanto, in particolare, considereremo gli sviluppi binari dei numeri reali[3].

Definizione 4.12 (Spazi rappresentati) *Uno* spazio rappresentato *è una coppia (X, δ_X) costituita da un insieme X ed una funzione parziale suriettiva $\delta :\subseteq \{0,1\}^{\mathbb{N}} \to X$.*

Analogamente al caso degli spazi topologici, spesso indicheremo uno spazio rappresentato (X, δ_X) con il solo insieme X, lasciando per implicita la rappresentazione δ_X.

Una funzione $F :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ è *computabile* se vi è una macchina TTE in grado di associare ad ogni $p \in \mathrm{dom}(F) \subseteq \{0,1\}^{\mathbb{N}}$ l'elemento $F(p)$.

È facile vedere che ogni funzione computabile è anche continua: la macchina produrrà un segmento iniziale v arbitrariamente lungo di $F(p)$ leggendo un certo segmento inziale (sufficientemente lungo) w di p[4]. Per fare ciò, la macchina può solo utilizzare l'informazione finita contenuta nel suo programma finito.

[3] Osserviamo che ad essere rigorosi per codificare i numeri secondo l'usuale sistema binario avremmo bisogno di cinque simboli: 0, 1, +, − e la virgola. Ma possiamo ridurci senza problemi ai solo 0 e 1. Ad esempio, possiamo supporre che ciascuna codifica cominci con $i0^{n+1}1$ per $i \in \{0,1\}$ e per un certo n, seguito poi da una qualsiasi successione binaria; $i = 0$ indica segno positivo, $i = 1$ indica segno negativo, e il numero di occorrenze di 0 in 0^n1 indica quante cifre dopo l'1 terminale bisogna contare per posizionare la virgola. Per semplicità, possiamo comunque concederci l'utilizzo dei segni + e − e della virgola basandoci sul fatto che una traduzione automatica in una successione puramente binaria è sempre possibile.

[4] Notare che nessun w potrebbe bastare alla produzione del solo v, magari ciò che la macchina scrive è un'estensione di v, ma questo non cambia nulla, perché se $F(w\{0,1\}^{\mathbb{N}}) \subseteq v'\{0,1\}^{\mathbb{N}}$ per $v' \in \{0,1\}^{<\mathbb{N}}$ estensione di v, allora comunque $F(w\{0,1\}^{\mathbb{N}}) \subseteq v\{0,1\}^{\mathbb{N}}$.

L'inverso vale solo se consideriamo la possibilità di accedere ad informazione ausiliaria detta *oracolo*. La funzione $F :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ è quindi continua sse è computabile da una qualche macchina TTE mediante un oracolo fissato $r \in \{0,1\}^{\mathbb{N}}$. Questo significa che c'è una macchina TTE capace di trasformare p in $F(p)$ ricevendo in input $\langle r, p\rangle$. Attenzione!: le successioni p variano sull'intero $\mathrm{dom}(F)$, ma r è fissato ed è lo stesso per tutti gli input di F! Un modo semplice per vedere la questione è la seguente: sia $(w_0, v_0)(w_1, v_1), (w_2, v_2)$.... un'enumerazione effettiva di tutte le coppie di parole in $\{0,1\}^{<\mathbb{N}}$. Allora un oracolo r per F può essere costituito da una successione $0^{n_0}10^{n_1}10^{n_2}1$.... tale che per ogni $q \in \mathrm{cod}(F)$ e per ogni $n \in \mathbb{N}$ esiste $v \in \{0,1\}$ tale che $|v| \geq n$ e $v \sqsubseteq q$, ed esiste $j \in \mathbb{N}$ tale che $v = v_{i_j}$ e $F(w_{i_j}\{0,1\}^{\mathbb{N}}) \subseteq v_{i_j}\{0,1\}^{\mathbb{N}} = v\{0,1\}^{\mathbb{N}}$. Se F è computabile, allora la macchina può ovviamente ignorare l'oracolo.

Ovviamente anche le macchine TTE (con oracolo) sono effettivamente enumerabili sulla base dei loro programmi finiti.

Esercizio 4.13 *Dimostra che:*

- *La funzione* $id : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}, p \mapsto p$ *è computabile, e quindi continua.*
- *La funzione* $H : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ *che tale che*

$$F(p)(n) = \begin{cases} 1 & se\ \varphi_k(k)\downarrow\ per\ k = \sum_{i=0}^{n} p(i) \\ 0 & se\ \varphi_k(k)\uparrow\ per\ k = \sum_{i=0}^{n} p(i) \end{cases}$$

  è continua ma non computabile.

Analogamente al caso degli spazi topologici, una funzione $f :\subseteq (X, \delta_X) \to (Y, \delta_Y)$ tra spazi rappresentati sarà una funzione $f :\subseteq X \to Y$ sugli insiemi corrispondenti.

Il concetto di computabilità viene trasferito dallo spazio di Cantor agli spazi rappresentati mediante i realizzatori:

Definizione 4.14 (Realizzatori) *Data la multifunzione* $f :\subseteq X \rightrightarrows Y$ *per* X, Y *spazi rappresentati, la funzione* $F :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ *è detta* realizzatore *di* f *(in simboli* $F \vdash f$*) sse* $\delta_Y(F(p)) = f(\delta_X(p))$ *per ogni* $p \in \mathrm{dom}(f \circ \delta_X)$.

Una funzione f tra spazi rappresentati sarà quindi *computabile* se ha un realizzatore computabile.

Possiamo trasferire anche la continuità dei realizzatori alle funzioni da esse realizzate: diciamo che una funzione $f : X \to Y$ è *realizer-continua* se possiede un realizzatore continuo.

Definizione 4.15 *Dati due spazi rappresentati* X *ed* Y*, lo spazio delle funzioni realizer-continue da* X *in* Y *è denotato come* $\mathcal{C}(X, Y)$ *ed è rappresentato mediante la rappresentazione* $[\delta_X \to \delta_Y]$ *tale che* $[\delta_X \to \delta_Y](0^n 1p) := f$ *se l'*n*-sima macchina TTE calcola* f *mediante l'oracolo* $p \in \{0,1\}$.

Definizione 4.16 (Spazio prodotto) *Dati due spazi rappresentati* X *ed* Y*, lo spazio prodotto* $X \times Y := \{ (x, y) \mid x \in X\ \&\ y \in Y \}$ *è rappresentato mediante la rappresentazione* $\delta_{X\times Y}$ *tale che* $\mathrm{dom}(\delta_{X\times Y}) = \{ \langle p, q\rangle \mid p \in \mathrm{dom}(\delta_X)\ \&\ q \in \mathrm{dom}(\delta_Y) \}$ *e* $\delta_{X\times Y}(\langle p, q\rangle) := (\delta_X(p), \delta_Y(q))$.

Proposizione 4.17 *Siano dati gli spazi rappresentati X, Y, Z. Le seguenti funzioni sono computabili:*

1. $eval : \mathcal{C}(X, Y) \times X \to Y$, $eval(f, x) = f(x)$
2. $curry : \mathcal{C}(X \times Y, Z) \to \mathcal{C}(X, \mathcal{C}(Y, Z))$, $curry(f) = x \mapsto (y \mapsto f(x, y))$
3. $uncurry : \mathcal{C}(X, \mathcal{C}(Y, Z)) \to \mathcal{C}(X \times Y, Z), uncurry(f) = (x, y) \mapsto (f(x))(y)$
4. $\circ : \mathcal{C}(Y, Z) \times \mathcal{C}(X, Y) \to C(X, Z), (g, f) \mapsto g \circ f$
5. $\times : \mathcal{C}(X, Y) \times \mathcal{C}(U, Z) \to \mathcal{C}(X \times U, Y \times Z), f \times g : (x, y) \mapsto (f(x), g(y))$
6. $const : Y \to \mathcal{C}(X, Y), const(y) = x \mapsto y$

DIM.:

1. Sia $[\delta_X \to \delta_Y](01^n p) = f$ e $\delta_X(q) = x$. Per definizione di $[\delta_X \to \delta_Y]$, l'n-sima macchina TTE è in grado di trasformare q in $F(q)$ per un qualche realizzatore $F \vdash f$ (cioè $\delta_Y(F(q)) = f(x) \in Y$) a partire dall'input $\langle p, q\rangle$ per un qualche p fissato per tutti gli elementi di $\mathrm{dom}(f \circ \delta_X)$, Utilizziamo quindi una *macchina universale TTE* che prendendo in input $\langle 0^n 1p, q\rangle$ simuli l'n-esima macchina TTE sull'input $\langle p, q\rangle$. Essa produrrà a sua volta $F(q)$.

2. Sia $[\delta_{X\times Y} \to \delta_Z])(0^n 1r) = f \in \mathcal{C}(X \times Y, Z)$. Ciò significa che l'$n$-esima macchina TTE M_n calcola un realizzatore $F \vdash f$ mediante l'oracolo fissato r, cioè se $\langle r, \langle p, q\rangle\rangle$ è dato in input ad M_n con $r \in \mathrm{dom}([\delta_{X\times Y} \to \delta_Z])$ e $\langle p, q\rangle \in \mathrm{dom}(\delta_{X\times Y})$, questa calcola $F(\langle p, q\rangle)$ che è un nome di $f(x, y)$. Si può modificare il programma di M_n ottenendo il programma di una macchina TTE $M_{n'}$ in grado di produrre per q grazie all'oracolo $\langle r, p\rangle$ lo stesso output che M_n forniva per $\langle p, q\rangle$ grazie all'oracolo r, cioè lo stesso nome $F(\langle p, q\rangle)$ di $f(x, y)$: è sufficiente chiedere ad $M_{n'}$ di cercare nella "seconda parte" dell'oracolo $\langle r, p\rangle$ quell'informazione che M_n trovava nella "prima parte' dell'input $\langle p, q\rangle$ di F, e di utilizzare quest'informazione esattamente come la usava M_n. Consideriamo ora una macchina TTE M che sia programmata per trasformare $\langle 0^n 1r, \langle p, q\rangle\rangle$ in $0^{n'}1\langle p, r\rangle$ per ogni $n \in \mathbb{N}$ ed $r, p, q \in \{0, 1\}^{\mathbb{N}}$ (cosa produca M per combinazioni infelici di tali elementi non ci interessa). Questa macchina calcola un realizzatore per *curry*.

3. Esercizio.

4. Esercizio.

5. La stessa (pseudo)coppia di oracoli $\langle r, s\rangle$ che viene usata per calcolare in parallelo due funzioni $f : X \to U$ e $g : Y \to Z$ può essere utilizzata per calcolare la funzione $(f \times g)(x, y) = (f(x), g(y))$.

6. Usiamo una macchina TTE che dato $p \in \mathrm{dom}(\delta_Y)$ con $\delta_Y(p) = y \in Y$ scriva in output $0^n 1p$, dove n è tale che l'ennesima macchina M_n restituisce come output p' per ogni input $\langle p', q\rangle$.

$\square$

Osservazione 4.18 *La funzione eval è molto importante perché ci assicura che le macchine TTE possono effettivamente usare i codici delle funzioni realizer-continue per calcolare le funzioni corrispondenti.*

La funzione curry è estremamente importante, perché riduce le dimostrazioni sulla possibilità di computare un codice per una funzione $g : Y \to Z$ (sulla base di certa informazione da X) all'esposizione di un metodo computabile uniforme per trasformare ciascun $y \in Y$ in $g(y) \in Z$.

4.3 Insiemi realizer-aperti e realizer-chiusi

Consideriamo ora i due seguenti spazi rappresentati:

- $(\mathbb{N}, \delta_{\mathbb{N}})$, per $\delta_{\mathbb{N}}(0^n 1 0^{\mathbb{N}}) := n$ per ogni $n \in \mathbb{N}$
- $(\mathbb{S}, \delta_{\mathbb{S}})$, per $\mathbb{S} := \{0, 1\}$ e $\delta_{\mathbb{S}}(0^{\mathbb{N}}) := 0$ e $\delta_{\mathbb{S}}(p) := 1$ se $p \neq 0^{\mathbb{N}}$ (**Spazio di Sierpinski**).

È facile vedere che le funzioni computabili parziali sullo spazio rappresentato $\mathbb{N}$ coincidono con le funzioni parziali ricorsive.

Lo spazio di Sierpinski ci permette di formalizzare in termini di spazi rappresentati il concetto di *semidecidibilità*: le funzioni computabili $f : \mathbb{N} \to \mathbb{S}$ determinano esattamente gli insiemi r.e. e co.r.e., rispettivamente, tramite $f^{-1}(\{1\})$ e $f^{-1}(\{0\})$. Questo fatto può essere esteso alle funzioni realizer-continue nel seguente modo:

Definizione 4.19 (Insieme realizer-aperti e realizer chiusi) *Dato lo spazio rappresentato X definiamo gli spazi rappresentati $\mathcal{O}(X)$ e $\mathcal{A}(X)$ degli insiemi* realizer-aperti *e* realizer-chiusi *di X nel seguente modo:*

$$\mathcal{O}(X) := \left\{ O \subseteq X \mid (\exists f \in \mathcal{C}(X, \mathbb{S}))\, O = f^{-1}(\{1\}) \right\}$$

$$\mathcal{A}(X) := \left\{ A \subseteq X \mid (\exists f \in \mathcal{C}(X, \mathbb{S}))\, A = f^{-1}(\{0\}) \right\}$$

$$\delta_{\mathcal{O}(X)}(p) := O \;\Leftrightarrow\; [\delta_X \to \delta_Y](p) = f \in \mathcal{C}(X, \mathbb{S}) \;\&\; O = f^{-1}(\{1\})$$

$$\delta_{\mathcal{A}(X)}(p) := X \setminus \delta_{\mathcal{O}(X)}(p)$$

Per $\mathcal{O}(X) \ni O := f^{-1}\{1\}$ possiamo dire che $f = \chi_O : X \to \mathbb{S}$ (notare che $\mathbb{S}$ sostituisce ora il precedente $\{0, 1\}$ con la topologia discreta!).

La rappresentazione $\delta_{\mathcal{A}(X)}$ definisce gli insiemi realizer-chiusi come il complementare degli insiemi realizer-aperti, in analogia al fatto che gli insiemi topologicamente chiusi sono il complementare degli insiemi topologicamente aperti. Ma attenzione! Per il momento non stiamo ancora assumendo che gli insiemi rappresentati siano dotati di una qualche topologia.

Proposizione 4.20 *Siano X, Y spazi rappresentati. Allora le seguenti funzioni sono ben definite e computabili:*

1. ${}^C : \mathcal{O}(X) \to \mathcal{A}(X), O \mapsto X \setminus O$, ${}^C : \mathcal{A}(X) \to \mathcal{O}(X), A \mapsto X \setminus A$
2. $\cup : \mathcal{O}(X) \times \mathcal{O}(X) \to \mathcal{O}(X),\ (O_1, O_2) \mapsto O_1 \cup O_2$,
   $\cup : \mathcal{A}(X) \times \mathcal{A}(X) \to \mathcal{A}(X),\ (A_1, A_2) \mapsto A_1 \cup A_2$

3. $\cap : \mathcal{O}(X) \times \mathcal{O}(X) \to \mathcal{O}(X),\ (O_1, O_2) \mapsto O_1 \cap O_2$
   $\cap : \mathcal{A}(X) \times \mathcal{A}(X) \to \mathcal{A}(X),\ (A_1, A_2) \mapsto A_1 \cap A_2$
4. $\bigcup : \mathcal{C}(\mathbb{N} \to \mathcal{O}(X)) \to \mathcal{O}(X), (O_n)_{n\in\mathbb{N}} \mapsto \bigcup O_n$
5. $\bigcap : \mathcal{A}(\mathbb{N} \to \mathcal{A}(X)) \to \mathcal{O}(X), (A_n)_{n\in\mathbb{N}} \mapsto \bigcap A_n$
6. *invert* $: \mathcal{C}(X, Y) \to \mathcal{C}(\mathcal{O}(Y) \to \mathcal{O}(X)), f \mapsto f^{-1}$
7. $\in : X \times \mathcal{O}(X) \to \mathbb{S}$ *tale che* $\in (x, O) = 1$ *se* $x \in O$ *e* $\notin (x, O) = 0$ *se* $x \notin O$
8. $\times : \mathcal{O}(X) \times \mathcal{O}(Y) \to \mathcal{O}(X \times Y)$, $\times : \mathcal{A}(X) \times \mathcal{A}(Y) \to \mathcal{A}(X \times Y)$

DIM.:

1. Esercizio.
2. Segue dalla Proposizione 4.17.4,5,6 (usando $const(1)$ per $1 \in \mathbb{S}$).
3. Esercizio.
4. Esercizio.
5. Esercizio.
6. Sia (dato un nome di) $f : X \to Y$ e $O \in \mathcal{O}(Y)$, e cioè (il nome di) $\chi_O : Y \to \mathbb{S}$. È chiaro che per ogni $x \in X$ si ha $x \in f^{-1}(O)$ sse $f(x) \in O$ sse $\chi_O \circ f(x) = 1$. Poniamo quindi $\chi_{f^{-1}(O)} := \chi_O \circ f : X \to \mathbb{S}$, che è computabile grazie alla computabilità della composizione (Proposizione 4.17.4)
7. È un caso particolare di eval (vedasi Proposizione 4.17.1).
8. Esercizio.

□

Si noti che gli insiemi realizer-aperti e realizer-chiusi si comportano esattamente come gli insiemi aperti e chiusi rispettivamente per quanto riguarda i punti 3,4,5 della proposizione precedente (si noti anche che non è in generale possibile scambiare $\bigcup$ con $\bigcap$ e viceversa nei punti 4 e 5).

Osservazione 4.21 *IMPORTANTE: si osservi che sulla base del punto 7 della Proposizione 4.20, un insieme O è realizer-aperto quando si può stabilire computabilmente (mediante un suo nome), e quindi in un numero finito di passi, quando un un oggetto x appartiene ad O.*

Dualmente, un insieme A è realizer-chiuso quando si può stabilire computabilmente (mediante un suo nome), e quindi in un numero finito di passi, quando un un oggetto x non appartiene ad A.

Finora non abbiamo supposto che gli spazi rappresentati siano dotati di una topologia, in quanto non era necessario per l'introduzione delle definizioni e per la dimostrazione dei primi risultati. A partire da questo momento forniremo agli spazi rappresentati una *topologia canonica*.

Definizione 4.22 (Topologia (canonica) rappresentata) *Dato uno spazio rappresentato* (X, δ_X)*, la* topologia canonica *indotta da* δ_X *è definita come:*

$$\tau_X := \left\{ O \subseteq X \mid \delta^{-1}(O) \text{ è aperto in } \mathrm{dom}(\delta_X) \right\}$$

Ovviamente $\mathrm{dom}(\delta_X)$ *è considerato con la topologia relativa indotta da* $\{0,1\}^{\mathbb{N}}$*.*

In pratica, la topologia rappresentata è la *topologia finale* (detta anche *topologia quoziente* o *topologia forte*) indotta da $\delta_X :\subseteq \{0,1\}^{\mathbb{N}} \to X$, ovvero la topologia più ricca tra quelle che rendono continua δ_X.

Si può verificare che

Teorema 4.23 *In una topologia canonica rappresentata X gli insiemi realizer-aperti coincidono esattamente con quelli aperti.*

Dim.: Sia O un insieme realizer-aperto di X, cioè $O \in \mathcal{O}(X)$. Allora la funzione $\chi_O : X \to \mathbb{S}$ ha un realizzatore continuo $F :\subseteq \{0,1\} \to \{0,1\}^{\mathbb{N}}$. Consideriamo l'insieme di tutti i δ_X-nomi che vengono proiettati da F su un nome di $1 \in \mathbb{S}$:

$$F^{-1}\Big(\{0,1\}^{\mathbb{N}} \setminus \{0^{\mathbb{N}}\}\Big) \cap \mathrm{dom}(\delta_X) := U.$$

Questo insieme U è aperto in $\mathrm{dom}(\delta_X)$ in quanto pre-immagine di un insieme aperto (in $\{0,1\}^{\mathbb{N}}$), ed inoltre $U = \delta_X^{-1}(O)$. Quindi O è aperto nella topologia finale di δ_X.

Viceversa, sia $O \subseteq X$ aperto, allora $\delta^{-1}(O)$ è aperto in $\mathrm{dom}(\delta_X) \subseteq \{0,1\}^{\mathbb{N}}$. Considerando un'enumerazione effettiva (iniettiva) $w_0, w_1, w_2, ...$ di $\{0,1\}^{<\mathbb{N}}$ avremo che

$$\delta_X^{-1}(O) := \Big(\bigcup_{j \in \mathbb{N}} w_{i_j} \{0,1\}^{\mathbb{N}} \Big) \cap \mathrm{dom}(\delta_X).$$

per un'opportuna scelta di $w_{i_0}, w_{i_1}, w_{i_2} ...$ (eventualmente con ripetizioni). Allora $0^{i_0} 10^{i_1} 10^{i_2} 10^{i_3} 1.....$ può essere preso come $[\delta_X \to \delta_{\mathbb{S}}]$-nome di $\chi_O : X \to \mathbb{S}$.
□

Questo teorema insieme alle Proposizioni 4.9 e 4.20.6 (computabilità, e quindi, buona definizione di invert) ci assicura che ***le funzioni realizer-continue sono anche continue***.

Ma attenzione! L'inverso non vale!!! E neppure la "versione uniforme" di questo fatto, ovvero la Proposizione 4.20.6 non è in generale... invertibile!!!

Nelle righe seguenti diamo le nozioni necessarie alla costruzione di un controesempio.

Consideriamo la seguente rappresentazione di $\mathbb{R}$. Consideriamo un'enumerazione $I_0, I_1, ..., I_n, ...$ effettiva (ed iniettiva) di tutti gli intervalli aperti razionali $]a,b[\in \mathbb{Q}^2$. Allora $\delta_{\mathbb{R}}(0^{i_0} 10^{i_1} 10^{i_2} ...) = x$ sse $I_{i_0}, I_{i_1}, I_{i_2}, ...$ è un'enumerazione di tutti gli intervalli razionali $]a,b[$ tali che $x \in]a,b[$. È facile mostrare che questa rappresentazione induce la topologia ordinaria generata dagli intervalli aperti (razionali) di $\mathbb{R}$. Da un lato sia $O \subseteq \mathbb{R}$ aperto. È sufficiente considerare che O sia un elemento della base, ovvero un intervallo razionale I_j. Allora $\delta_{\mathbb{R}}^{-1}(O)$ è dato da tutte le successioni in $\mathrm{dom}(\delta_{\mathbb{R}})$ che cominciano con $0^j 1$ oppure contengono la stringa $10^j 1$. Tale insieme è quindi aperto in $\mathrm{dom}(\delta_{\mathbb{R}}) \subseteq \{0,1\}^{\mathbb{N}}$. Viceversa sia $\delta_{\mathbb{R}}^{-1}(O)$ aperto in $\mathrm{dom}(\delta_{\mathbb{R}})$. Allora tale aperto è dato da(ll'intersezione con $\mathrm{dom}(\delta_{\mathbb{R}})$ di) un insieme del tipo $\bigcup_{j \in J} w_{i_j} \{0,1\}^{\mathbb{N}}$. Senza perdita di generalità possiamo supporre che $w_{i_j}\{0,1\}^{\mathbb{N}} \cap \mathrm{dom}(\delta_{\mathbb{R}}) \neq \emptyset$ per qualsiasi $j \in J$ (altrimenti tale w_{i_j} può essere eliminato). Possiamo inoltre supporre che ciascun w_{i_j} termini con 1 (altrimenti w_{i_j} può essere sosituito con tutte le sue possibili estensioni w' finite terminanti con 1 e tali che $w'\{0,1\}^{\mathbb{N}} \cap \mathrm{dom}(\delta_{\mathbb{R}}) \neq \emptyset$). In questo modo ciascun w_{i_j} elenca in modo compiuto una successione finita di intervalli aperti razionali, la cui intersezione sarà l'intervallo rzionale che contiene esattamente tutti i reali con un nome cominciante per w_{i_j}. Allora $\delta_{\mathbb{R}}(\bigcup_{j\in J} w_{i_j}\{0,1\}^{\mathbb{N}}) = \bigcup_{j\in J} \delta_{\mathbb{R}}(w_{i_j}\{0,1\}^{\mathbb{N}})$ è quindi un unione di intervalli aperti, vale a dire che è un insieme aperto nella topologia standard di $\mathbb{R}$, e per di più coincide proprio con O.

Consideriamo ora invece l'insieme dei numeri reali $\mathbb{R}$ rappresentato con la rappresentazione binaria δ_b (indichiamo per brevità lo spazio così rappresentato come $\mathbb{R}_b$). Abbiamo già osservato che utilizzando delle convenzioni notazionali possiamo considerare che un nome binario di un numero reale sia della forma "sw,p" per $s \in \{+,-\}$, $w = w(|w|-1)...w(0) \in \{0,1\}^{<\mathbb{N}}$ e $p = p(1)p(2)p(3)... \in \{0,1\}^{\mathbb{N}}$. Questa successione codifica il numero

$$s\Big(\sum_{i=0}^{|w|-1} w(i)\cdot 2^i + \sum_{i=1}^{\infty} p(i)\cdot 2^{-i}\Big) = \sum_{i=0}^{|w|-1} sw(i)\cdot 2^i + \sum_{i=1}^{\infty} sp(i)\cdot 2^{-i}.$$

Ora mostriamo che ciascun insieme aperto nella topologia finale di δ_b è l'unione di intervalli aperti in $\mathbb{R}$. Sia quindi $O \subseteq \mathbb{R}$ aperto nella topologia indotta da δ_b. Allora $\delta_b^{-}(O)$ è aperto (nella topologia relativa di $\mathrm{dom}(\delta_b)$ rispetto a $\{0,1\}^{\mathbb{N}}$). Sia $x \in O$ e sia p uno sviluppo binario di x. Esiste quindi $w \in \{0,1\}^{<\mathbb{N}}$, che senza perdita di generalità è della forma sw_1, w_2 (con $w_2 = w_2(1)...w_2(|w_2|)$), tale che $w \sqsubseteq p$ e $w\{0,1\}^{\mathbb{N}} \subseteq \delta_b^{-}(O)$. Per $p \neq sw_1, w_2 0^{\mathbb{N}}$ e $p \neq sw_1, sw_2 1^{\mathbb{N}}$ osserviamo che se $s = +$ allora $x \in]\delta_b(sw_1, w_2 0^{\mathbb{N}}), \delta_b(sw_1, w_2 1^{\mathbb{N}})[\subseteq O$, e se $s = -$ allora $x \in]\delta_b(sw_1, w_2 1^{\mathbb{N}}), \delta_b(sw_1, w_2 0^{\mathbb{N}})[\subseteq O$ (infatti se $s = +$ allora $\delta_b(sw_1, w_2 0^{\mathbb{N}}) < x$ in quanto in p occorrono degli 1 a destra di w_2, e analogamente $x < \delta_b(sw_1, w_2 1^{\mathbb{N}})$ in quanto in p occorrono degli 0 a destra di w_2; per $s = -$ si dà il contrario). Sia ora invece $p = sw_1, w_2 1^{\mathbb{N}}$. Osserviamo che $\sum_{j>|w_2|} 1\cdot 2^{-j} = \sum_{j>|w_2|} 2^{-j} = 2^{-|w_2|}$. Pertanto un altro nome binario di x è ottenibile sostituendo a destra di w_2 tutti gli 1 con 0 e compensando la conseguente variazione di quantità modificando la sola stringa sw_1, w_2 in modo che $s2^{-|w|}$ venga nuovamente riaggiunto. In questo modo otterremo la stringa sw_1', w_2': se l'ultima cifra di w_2 è 0, allora questa nuova stringa è immediatamente ottenuta da sw_1, w_2 cambiando tale 0 in 1 e lasciando il resto inalterato (infatti avremo immediatamente riaggiunto $s1\cdot 2^{-|w_2|}$ nella somma che definisce x; altrimenti sarà necessario eseguire l'operazione con le regole dei riporti). Siccome $x \in O$ e $\delta_b(sw_1', w_2' 0^{\mathbb{N}}) = x$, allora $sw_1', w_2' 0^{\mathbb{N}} \in \delta_b^{-1}(O)$ e sappiamo che questo insieme è aperto. Pertanto esiste un n tale che $sw_1', w_2' 0^n\{0,1\}^{\mathbb{N}} \in \delta_b^{-1}(O)$. Quindi: (i) se $s = +$ allora $x \in]\delta_b(sw_1, w_2 0^{\mathbb{N}}), \delta_b(sw_1', w_2' 0^n 1^{\mathbb{N}})[\subseteq O$; (ii) se $s = -$ allora $x \in]\delta_b(sw_1', w_2' 0^n 1^{\mathbb{N}}), \delta_b(sw_1, w_2 0^{\mathbb{N}})[\subseteq O$. La dimostrazione per il caso $p = sw_1, w_2 0^{\mathbb{N}}$ è analoga[5].

Difatto, abbiamo dimostrato non solo che gli aperti di $\mathbb{R}_b$ sono aperti di $\mathbb{R}$ (dove la topologia di quest'ultimo è quella definita mediante gli intervalli aperti), ma anche che $id : \mathcal{O}(\mathbb{R}_b) \to \mathcal{O}(\mathbb{R})$ è computabile: infatti senza perdita

[5] L'unico caso particolare a tal rispetto è lo 0, per il quale vale $0 = \delta_b(+0, 0^{\mathbb{N}}) = \delta_b(-0, 0^{\mathbb{N}})$. In questo caso otterremo intervalli del tipo $]\delta_b(-0, 0^n 1^{\mathbb{N}}), \delta_b(0, 0^m 1^{\mathbb{N}})[$.

di generalità, un $\delta_{\mathcal{O}(\mathbb{R}_b)}$- nome di un aperto O in $\mathbb{R}_b$ è dato da una successione di parole del tipo $s_i w_1^i, w_2^i$ per $i \in \mathbb{N}$ con $s_i \in \{+, -\}$, $w_1^i, w_2^i \in \{0,1\}^{<\mathbb{N}}$ tale che $x \in O$ sse per ogni δ_b-nome p di x esiste $i \in \mathbb{N}$ tale che $s_i w_1^i, w_2^i \sqsubseteq p$ (esercizio). La dimostrazione di identità data può poi facilmente essere trasformata in una procedura di automatica uniforme di traduzione di qualsiasi $O \in \mathcal{O}(\mathbb{R}_b)$ nello stesso $O \in \mathcal{O}(\mathbb{R})$.

Ma la funzione $id : \mathcal{O}(\mathbb{R}^b) \to \mathcal{O}(\mathbb{R})$ appena considerata è l'(estensione dell')inversa (agli insiemi aperti) della funzione $id : \mathbb{R} \to \mathbb{R}_b, x \mapsto x$, e pertanto quest'ultima funzione è continua. Tuttavia non è realizer-continua. Supponiamo infatti che lo sia. Allora vi è una macchina di Turing in grado di calcolarla (eventualmente mediante un oracolo). Consideriamo allora $x \in]-1, 1[$. Leggendo un $\delta_{\mathbb{R}}$-nome p di x la macchina di Turing deve prima o poi scrivere l'inizio dello sviluppo binario, che dovrà essere o $-0^n, w_0$ o $+0^m, w_1$ per $w_0, w_1 \in \{0,1\}^{<\mathbb{N}}$ ed $m, n \geq 1$, a seconda del reale valore di x. Sia w l'intervallo iniziale di p letto al momento in cui la macchina comincia a scrivere l'output, e supponiamo che senza perdita di generalità termini con 1. Siano pertanto $I_{i_0}, ..., I_{i_n}$ gli intervalli compiutamente enumerati in w. Supponiamo ancora che $I = \bigcap_{j=0}^{n} I_{i_j} =]a, b[\subseteq -]1, 1[$ per $a < 0$ e $b > 0$ (ciò è plausibile, perché p potrebbe effettivamente denotare 0), e che la macchina scelga di scrivere $-0^n, w_0$. Al passo seguente la macchina potrebbe accorgersi che p elenca un intervallo del tipo $]c, d[$ per $a < 0 < c < d$. A questo punto è ovvio che p denota un numero positivo, pertanto la macchina ha commesso un errore. La dimostrazione è analoga se la macchina sceglie $+0^m, w_1$. Pertanto id non è realizer-continua.

Più in generale possiamo definire una *rappresentazione standard* δ_X per qualsiasi spazio T_0 secondo numerabile X in analogia al caso di $(\mathbb{R}, \delta_{\mathbb{R}})$.

Uno spazio topologico X è T_0 quando per ogni coppia di punti distinti $x, y \in X$ esiste un insieme aperto $O \subseteq X$ tale che $x \in O \Leftrightarrow y \notin O$.

Uno spazio topologico X è *secondo numerabile* quando possiede una (sotto)base numerabile. Sia quindi $B_0, B_1, B_2, ...$ un'enumerazione di una sua sottobase numerabile (eventualmente con ripetizioni). Allora $\delta_X(0^{i_0}10^{i1}10^{i2}1...) = x$ sse $B_{i_0}, B_{i_1}, B_{i_2}, ...$ è un'enumerazione di tutti gli aperti B della sottobase tali che $x \in B$. Intuitivamente, in uno spazio T_0 secondo numerabile X possiamo descrivere in modo univoco qualsiasi $x \in X$ elencando l'insieme di tutte le *proprietà atomiche* da esso godute, in quanto ciascun x è univocamente determinato dalla classe delle proprietà atomiche da esso godute.

La rappresentazione $\delta_{\mathbb{R}}$ è un caso particolare di questo tipo di rappresentazione, essendo $\mathbb{R}$ uno spazio T_0 secondo numerabile mediante la (sotto)base numerabile degli intervalli razionali. Lo stesso vale per $\mathbb{R}^n$, dove una base numerabile di $\mathbb{R}^n$ è costituita dalla classe di tutte le palle aperte razionali, ovvero insiemi del tipo $B(a, r) := \{\, x \in \mathbb{R}^n \mid d(a, x) < r \,\}$ per $a \in \mathbb{Q}^n$ e $r \in \mathbb{Q}$ con $r \geq 0$.

Definizione 4.24 (Rappresentazione topologica degli aperti e chiusi di $\mathbb{R}^n$) *Sia $B_0, B_1, B_2, ...$ un'enumerazione effettiva di tutte le palle aperte razionali. Definiamo la seguente rappresentazione $\delta_{\mathrm{open}(\mathbb{R}^n)}$ degli insiemi aperti in $\mathbb{R}^n$:*

$$\delta_{\mathrm{open}(\mathbb{R}^n)}(0^{i_0}10^{i_1}10^{i_2}1...) := O \Leftrightarrow O = \bigcup_{j=0}^{\infty} B_{i_j}.$$

Definiamo la seguente rappresentazione $\delta_{\text{closed}(\mathbb{R}^n)}$ *degli insiemi chiusi in* $\mathbb{R}^n$*:*

$$\delta_{\text{closed}(\mathbb{R}^n)}(p) := \mathbb{R}^n \setminus \delta_{\mathcal{O}(\mathbb{R}^n)}(p).$$

La rappresentazione $\delta_{\text{open}(\mathbb{R}^n)}$ è computabilmente equivalente a $\delta_{\mathcal{O}(\mathbb{R}^n)}$ (e analogamente per $\delta_{\text{closed}(\mathbb{R}^n)}$ e $\delta_{\mathcal{A}(\mathbb{R}^n)}$), ovvero la funzione $id : \mathcal{O}(\mathbb{R}^n) \to \text{open}(\mathbb{R}^n), O \mapsto O$, così come la sua inversa, sono computabili (dimostrazione per esercizio).

Si noti che le rappresentazioni $\delta_{\mathbb{R}^n}$ e $\delta_{\text{open}(\mathbb{R}^n)}$ mostrano in modo geometricamente intuitivo la proprietà di appartenenza definita dalla Proposizione 4.20.7.

4.4 Spazi rappresentati ammissibili

Definizione 4.25 (Insiemi compatti) *Sia dato uno spazio topologico X. Un sottoinsieme $K \subseteq X$ è detto* compatto *quando ogni ricoprimento aperto di K possiede un sottoricoprimento finito, ovvero per ogni collezione di insiemi aperti $\{O_i\}_{i\in I}$ tale che $K \subseteq \bigcup_{i\in I} O_i$, esiste $J \subseteq I$ finito tale che $K \subseteq \bigcup_{i\in J} O_i$.*

Come nel caso dei concetti di continuità, apertura e chiusura, abbiamo anche una versione di compattezza relativa agli spazi rappresentati e ai realizzatori. Nel caso più generale si applica agli insiemi saturati.

Definizione 4.26 (Insiemi saturati) *Sia X uno spazio rappresentato. Dato $M \subseteq X$, la* realizer-saturazione *$\uparrow M$ di M è definita come*

$$\uparrow M = \bigcap \{\, O \in \mathcal{O}(X) \mid M \subseteq O \,\}.$$

Un insieme $S \subseteq X$ è poi detto realizer-saturato *se $S = \uparrow S$.*

Le definizioni di saturazione *ed* insieme saturato *si ottengono da quelle precedenti sostituendo lo spazio rappresentato con uno topologico, e gli insiemi realizer-aperti $O \in \mathcal{O}(X)$ con gli insiemi (topologicamente) aperti $U \subseteq X$.*

Notiamo che le definizioni di insieme realizer-saturato ed insieme saturato coincidono quindi per gli spazi rappresentati X con topologia indotta da δ_X.

Definizione 4.27 (Insiemi realizer-compatti) *Sia X uno spazio rappresentato. Un insieme K realizer-saturato è detto* realizer-compatto *se la funzione* $\text{IsContainedIn}_K : \mathcal{O}(X) \to \mathbb{S}$ *tale che*

$$\text{IsContainedIn}_K(O) = 1 \;\Leftrightarrow\; K \subseteq O$$

è realizer-continua.

Sia $\mathcal{K}(X)$ lo spazio degli insiemi realizer-compatti. Definiamo la rappresentazione $\delta_{\mathcal{K}(X)}$ come:

$$\delta_{\mathcal{K}(X)}(p) := K \;\Leftrightarrow\; [\delta_{\mathcal{O}(X)} \to \delta_{\mathbb{S}}](p) = \text{IsContainedIn}_K.$$

In pratica tramite la rappresentazione $\delta_{\mathcal{K}(X)}$ stiamo identificando $\mathcal{K}(X)$ con $\mathcal{O}(\mathcal{O}(X))$.

Si osservi che (per X spazio rapresentato canonico) $\uparrow \{x\} \in \mathcal{K}(X)$. Infatti $\uparrow \{x\} \subseteq O \;\Leftrightarrow\; \{x\} \subseteq O \;\Leftrightarrow\; x \in O$ per ogni aperto $O \subseteq X$. Quindi un $\delta_{\mathcal{K}(X)}$-nome di $\uparrow \{x\}$ è essenzialmente dato da(ll'indice numerico di) un programma che calcoli eval e da una codifica di x: tale programma applica la funzione $\chi_O : X \to \mathbb{S}$ a x e se $\chi_O(x) = 1$, dà risposta positiva a "IsContainedIn$_{\uparrow\{x\}}(O)$?"

Si osservi che questo fatto mostra l'importanza delle saturazioni. Infatti non è detto che $\uparrow \{x\} = \{x\}$. Ad esempio in $\mathbb{S}$ si ha $\uparrow \{0\} = \{0, 1\}$. Pertanto le domande "$\{0\} \subseteq U$?" e "$\{0, 1\} \subseteq U$?" darebbero esattamente le stesse risposte per ogni aperto $U \in \mathbb{S}$. Affinchè quindi le rappresentazioni $\delta_{\mathcal{K}(X)}$ siano quindi ben definite, prendiamo le saturazioni degli insiemi. Tuttavia per gli insiemi T_1 gli insiemi coincidono con le loro saturazioni.

Definizione 4.28 (Spazi T_1) *Uno spazio topologico X è detto T_1 quando per ogni coppia di punti distinti $x_0, x_1 \in X$ esistono due aperti distinti O_0, O_1 in X tali che $x_i \in O_i$ e $x_i \notin O_{1-i}$ per $i \in \{0, 1\}$.*

Si può dimostrare che:

Proposizione 4.29 *Uno spazio topologico X è T_1 sse $\{x\}$ è chiuso in X per ogni $x \in X$.*

Proposizione 4.30 *Sia X uno spazio topologico T_1. Allora $\uparrow M = M$ per ogni sottoinsieme $M \subseteq X$.*

Dim.: È ovvio che $M \subseteq \uparrow M$. Mostriamo ora che $\uparrow M \subseteq M$. Supponiamo che ciò non sia. Allora esiste $x \in X$ tale che $x \in \uparrow M$ ma $x \notin M$. Dal fatto che $x \in \uparrow M$ segue che $x \in O$ per ogni aperto O tale che $M \subseteq O$. Poiché $x \notin M$, si ha $M \subseteq X \setminus \{x\}$, ma $X \setminus \{x\}$ è aperto per la Proposizione 4.29. Pertanto non è vero che $x \in O$ per ogni aperto O con $M \subseteq O$. Contraddizione. □

Esempio 4.31 *Nello spazio T_1 $\mathbb{R}^n$ rappresentato in modo standard (in quanto spazio T_0 secondo numerabile) possiamo definire una rappresentazione per gli insiemi compatti K nel seguente modo. Consideriamo un'enumerazione computabile $R_0, R_1, R_2, ...$ di tutti gli insiemi finiti di palle aperte razionali in $\mathbb{R}^n$. Allora $\delta_{\mathrm{cover}(\mathbb{R}^n)}(0^{i_0}10^{i_1}10^{i_2}1...) := K$ sse $R_{i_0}, R_{i_1}, R_{i_2}...$ è un'enumerazione di tutti i ricoprimenti aperti finiti razionali di K.*

Questa rappresentazione induce la topologia T_0 secondo numerabile che ha come sottobase la collezione di tutti gli insiemi del tipo $\{ K \in \mathcal{K}(\mathbb{R}^n) \mid K \subseteq R_i \}$ per un certo i.

Si può dimostrare che tale rappresentazione $\delta_{\mathrm{cover}(\mathbb{R}^n)}$ è computabilmente equivalente a $\delta_{\mathcal{K}(\mathbb{R}^n)}$.

Proposizione 4.32 *Siano X, Y spazi rappresentati. Allora le seguenti operazioni sono ben definite e computabili:*

1. $k_X : X \to \mathcal{K}(X), x \mapsto \uparrow \{x\}$
2. $\cup : \mathcal{K}(X) \times \mathcal{K}(X) \to \mathcal{K}(X), (K_1, K_2) \mapsto K_1 \cup K_2$
3. $\uparrow \cap : \mathcal{K}(X) \times \mathcal{A}(X) \to \mathcal{K}(X), (K, A) \mapsto \uparrow (K \cap A)$

Dim.:

1. Già dimostrato.
2. $\cup : \mathcal{K}(X) \times \mathcal{K}(X) \to \mathcal{K}(X)$ è realizzata da $\cup : \mathcal{O}(\mathcal{O}(X)) \times \mathcal{O}(\mathcal{O}(X)) \to \mathcal{O}(\mathcal{O}(X))$ per la Proposizione 4.20.2.

3. Segue dalla Proposizione 4.20.1,2 in quanto $K \cap A \subseteq O$ per $O \in \mathcal{O}(X)$ sse $K \subseteq \Big(O \cup (X \setminus A)\Big)$

$\square$

Consideriamo ora meglio la funzione k_X. Tramite l'identificazione di $\mathcal{K}(X)$ con $\mathcal{O}(\mathcal{O}(X))$ possiamo interpretare (e rappresentare!) $k_X(x) =\uparrow \{x\}$ come $k_X(x) = \{\, O \in \mathcal{O}(X) \mid x \in O\,\}$. Quindi un nome di $k_X(x)$ sarà dato da un codice per la funzione $\chi_{k_X(x)} : \mathcal{O}(X) \to \mathbb{S}$ tale che $\chi_{k_X(x)}(O) = 1 \Leftrightarrow x \in O$. Come abbiamo visto, questa funzione k_X è computabile.

Notazione 4.33 *Denotiamo con X_k l'immagine $k_X(X) \subseteq \mathcal{O}(\mathcal{O}(X))$ rappresentata con la rappresentazione $\delta_{\mathcal{O}(\mathcal{O}(X))}$.*

Diamo ora finalmente la definizione cruciale della teoria:

Definizione 4.34 (Spazi ammissibili) *Uno spazio topologico canonicamente rappresentato X è* (computabilmente) ammissibile *sse $k_X : X \to X_k$ ammette un'inversa continua (computabile) $k_X^{-1} : X_k \to X$.*

Esempio 4.35

1. *$\mathbb{S}$ è computabilmente ammissibile: dato $x_k \in \mathbb{S}_k$ possiamo (semi)verificare se $x \in \{1\} \subseteq \mathbb{S}$ (si noti che $\{1\}$ è computabile). Nel frattempo scriviamo sull'output occorrenze di 0. Se il test darà prima o poi risultato positivo, allora prosegueremo la scrittura dell'output aggiungendo $10^{\mathbb{N}}$, altrimenti $0^{\mathbb{N}}$ sarà il risulato finale.*

2. *X T_0 secondo numerabile è ammissibile: sia $O_0, O_1, O_2, ...$ un'enumerazione degli aperti della sottobase di X (eventualmente con ripetizioni), e sia dato $x_k \in X_k$. Testiamo in parallelo x_k su $O_0, O_1, O_2, ...$ Man mano che il test da risutlato positivo per un qualche O_i, proseguiamo la scrittura dell'output con 0^i1 (per essere sicuri che l'output sia infinito ripetizioni possono essere ammesse). Questo ci dà un nome di x secondo la rappresentazione standard.*

   Si noti che anche $\mathbb{S}$ è uno spazio T_0 e la rappresentazione $\delta_{\mathbb{S}}$ è infatti computabilmente equivalente alla rappresentazione standard di $\mathbb{S}$ come spazio T_0 secondo numerabile (esercizio).

   Un'altro caso particolare è dato da $\mathbb{N}$.

3. *Si consideri la* rappresentazione di Cauchy *di $\mathbb{R}$: data un'enumerazione computabile $r_0, r_1, r_2, ...$ dei numeri razionali poniamo $\delta_C(0^{i_0}10^{i_1}10^{i_2}1...) = x \in \mathbb{R}$ sse (i) $|r_{i_j} - r_{i_l}| \leq 2^{-j}$ per $l \geq j$, e (ii) $\lim_{j\to\infty} r_{i_j} = x$. Questa rappresentazione è ammissibile rispetto alla topologia ordinaria su $\mathbb{R}$ ed è infatti computabilmente equivalente alla rappresentazione di $\mathbb{R}$ come spazio T_0 secondo numerabile (esercizio). Si noti che la rappresentazione* naif di Cauchy *per la quale la condizione (i) non vale necessariamente non è invece ammissibile.*

4. *Lo spazio di Cantor $\{0,1\}^{\mathbb{N}}$ rappresentato con $id : p \mapsto p$ è computabilmente ammissibile. Anche lo spazio di Cantor è T_0 secondo numerabile, ed id è una rappresentazione computabilmente equivalente alla sua rappresentazione standard in quanto spazio T_0 secondo numerabile (esercizio).*

5. *La rappresentazione* $\delta_{\mathrm{open}(\mathbb{R}^n)}$ *è ammissibile rispetto all'*upper Fell topology *già studiata in Teoria Descrittiva degli Insiemi generata dalla sottobase data dagli insiemi* $\{ O \in \mathcal{O}(\mathbb{R}^n) \mid O \cap K = \emptyset \}$ *per* K *compatto. Anche questo spazio è* T_0 *secondo numerabile.*

6. *La rappresentazione* $[\delta_{\mathbb{R}^k} \to \delta_{\mathbb{R}^n}]$ *è ammissibile rispetto alla* topologia compact-open *già studiata in Topologia generata dalla sottobase data dali insiemi* $\left\{ f \in \mathcal{C}(\mathbb{R}^k, \mathbb{R}^n) \mid f(K) \subseteq O \right\}$ *per* K *compatto. Anche questo spazio è* T_0 *secondo numerabile.*

Uno spazio non ha tuttavia bisogno di essere necessariamente secondo numerabile per essere ammissibile. Questo lo si può vedere sulla base del prossimo teorema:

Teorema 4.36 *Sia* Y *(computabilmente) ammissibile. Allora* $\mathcal{C}(X,Y)$ *è (computabilmente) ammissibile.*

DIM.: Dato $O \in \mathcal{O}(Y)$ e $x \in X$ possiamo computare $\{ g \in \mathcal{C}(X,Y) \mid g(x) \in O \} \in \mathcal{O}(\mathcal{C}(X,Y))$ mediante eval (Proposizione 4.17.1): presa g in input, calcoliamo $g(x)$ e verifichiamo poi se $g(x) \in O$ mediante Proposizione 4.20.7. Tramite curry (Proposizione 4.17.2) stiamo cioè calcolando con oracolo x una funzione $H_x : O \in \mathcal{O}(Y) \mapsto \{ g \in \mathcal{C}(X,Y) \mid g(x) \in O \} \in \mathcal{O}(\mathcal{C}(X,Y))$.

Consideriamo ora $f_k = \{ U \in \mathcal{O}(\mathcal{C}(X,Y)) \mid f \in U \} \in \mathcal{O}(\mathcal{O}(\mathcal{C}(X,Y)))$ per una precisa $f \in \mathcal{C}(X,Y)$. Allora $H_x(O) \in f_k$ sse $f \in H_x(O)$ sse $f(x) \in O$, dunque $H_x^{-1}(f_k) = \{ O \in \mathcal{O}(Y) \mid f(x) \in O \} = f(x)_k \in \mathcal{O}(\mathcal{O}(Y))$.

Applicando invert (Proposizione 4.20.6), data $H_x : \mathcal{O}(Y) \to \mathcal{O}(\mathcal{C}(X,Y))$ (notare che questa funzione è effettivamente totale!) possiamo computare $H_x^{-1} : \mathcal{O}(\mathcal{O}(\mathcal{C}(X,Y))) \to \mathcal{O}(\mathcal{O}(Y))$. Pertanto dati f_k e H_x (e per avere quest'ultima ci basta, in ultima istanza, x!) possiamo computare $H_x^{-1}(f_k) = f(x)_k$. Poiché Y è ammissibile, questo è sufficiente a farci computare $f(x) \in Y$. In conclusione possiamo calcolare $\mathcal{C}(X,Y)_k \times X \to Y, (f_k, x) \mapsto f(x)$, da cui per curry (Proposizione 4.17.2) possiamo calcolare $k^{-1}_{\mathcal{C}(0,1)} : \mathcal{C}(X,Y)_k \to \mathcal{C}(X,Y), f_k \mapsto f$. $\square$

Corollario 4.37 *Dato uno spazio topologico canonicamente rappresentato, gli spazi* $\mathcal{O}(X), \mathcal{A}(X), \mathcal{K}(X), X_k$ *sono computabilmente ammissibili.*

DIM.: Segue dal Teorema 4.36 e dal fatto che $\mathbb{S}$ è computabilmente ammissibile (Esempio 4.35.1). $\square$

Grazie a questo risultato si può dedurre l'esistenza di spazi ammissibili che non secondo numerabili.

Esempio 4.38 *Lo spazio di Baire* $\mathbb{N}^{\mathbb{N}}$ *è ammissibile (è uno spazio* T_0 *secondo numerabile, con base numerabile data dagli insiemi della forma* $w\mathbb{N}^{\mathbb{N}}$ *per* $w \in \mathbb{N}^{<\mathbb{N}}$*, e cioè è la topologia prodotto della topologia discreta su* $\mathbb{N}$*). Una rappresentazione ammissibile di* $\mathbb{N}^{\mathbb{N}}$ *(equivalente a quella standard come spazio* T_0 *secondo numerabile) è data banalmente da* $\delta_{\mathbb{N}^{\mathbb{N}}}(0^{n_0}10^{n_1}10^{n_2}1...) = n_0 n_1 n_2...$

Tuttavia $\mathcal{O}(\mathbb{N}^{\mathbb{N}})$ *non è secondo numerabile.*

Notazione 4.39 *Dati due spazi rappresentati* X, Y *sciriviamo* $X \simeq Y$ *se* X *ed* Y *sono computabilmente isomorfi, ovvero vi è una funzione biettiva* $f : X \to Y$ *computabile con inversa* $f^{-1} : Y \to X$ *a sua volta computabile.*

$X \simeq X_k$ vale solo se X è ammissibile. Tuttavia il seguente risultato vale in generale:

Corollario 4.40 *Per ogni spazio topologico X canonicamente rappresentato vale $\mathcal{O}(X) \simeq \mathcal{O}(X_k)$, $\mathcal{A}(X) \simeq \mathcal{A}(X_k)$.*

DIM.: La funzione $O \mapsto O_k := \{\, x_k \mid x \in O \,\} = k_X(O)$ per $O \in \mathcal{O}(X)$ è computabile. Sia infatti dato $O \in \mathcal{O}(X)$. Prendiamo un generico x_k e vogliamo determinare se $x_k \in O_k$ tramite l'input $O \in \mathcal{O}(X)$. Sappiamo che $x_k := \{\, O \in \mathcal{O}(X) \mid x \in O \,\} \in \mathcal{O}(\mathcal{O}(X))$. Per cui $x_k \in O_k$ sse $x \in O$ sse $O \in x_k$. Quest'ultima relazione è semidecidibile per l'input O per definizione di x_k. Questo determina O_k come elemento di $\mathcal{O}(X_k)$.

Anche la funzione inversa $O_k \mapsto O := \{\, x \in X \mid x_k \in O_k \,\}) = k_X^{-1}(O_k)$ è computabile: dato O_k, vogliamo sapere se un generico $x \in X$ è in O; usiamo pertanto la computabilità di $x \mapsto x_k$ per verificare (in modo semidecidibile) se $x_k \in O_k$. Questo determina O come elemento di $\mathcal{O}(X)$. $\square$

Teorema 4.41 *Le seguenti proprietà sono equivalenti per uno spazio topologico Y canonicamente rappresentato:*

1. *Y è (computabilmente) ammissibile*
2. *$invert : \mathcal{C}(X,Y) \to \mathcal{C}(\mathcal{O}(Y), \mathcal{O}(X)), f \mapsto f^{-1}$ ha una funzione inversa ben definita e continua (computabile) per ogni spazio topologico X canonicamente rappresentato.*

DIM.: 1.$\Rightarrow$ 2.) Sia data $f^{-1} \in \mathcal{C}(\mathcal{O}(Y), \mathcal{O}(X))$ in input. Vediamo che possiamo computare $f \in \mathcal{C}(X,Y)$, ovvero (per curry), che dato un qualsiasi $x \in X$ possiamo calcolare $f(x) \in Y$. Vediamo che possiamo innanzitutto calcolare $f(x)_k = \{\, O \in \mathcal{O}(Y) \mid f(x) \in O \,\} \in \mathcal{O}(\mathcal{O}(Y))$. Osserviamo che $f(x) \in O$ sse $x \in f^{-1}(O)$ per ogni $O \in \mathcal{O}(Y)$. Pertanto, dato $O \in \mathcal{O}(Y)$ usiamo f^{-1} per calcolare $f^{-1}(O)$, e testiamo se $x \in f^{-1}(O)$. In caso affermativo $O \in f(x)_k \in \mathcal{O}(\mathcal{O}(Y))$. Poichè Y è ammissibile, possiamo poi trasformare computabilmente $f(x)_k$ in $f(x)$.

2. $\Rightarrow$ 1.) Per prima cosa vediamo che $k_Y^{-1} : Y_k \to Y$ è ben definita. Assumiamo per assurdo che k_Y non sia iniettiva (essa è ovviamente suriettiva per definizione), cioè che esistano $y_1, y_2 \in Y$ con $y_1 \neq y_2$ e $k_Y(y_1) = k_Y(y_2)$, il che significa $y_1 \in O$ sse $y_2 \in O$ per ogni $O \in \mathcal{O}(Y)$. Allora le funzioni costanti $f_{y_1}, f_{y_2} : \{*\} \to Y$ con $f_{y_1}(*) := y_1$, $f_{y_2}(*) := y_2$ sono continue e distinte, ma $f_{y_1}^{-1} = f_{y_2}^{-1}$ con $f_{y_i}^{-1} : \mathcal{O}(Y) \to \mathcal{O}(\{*\})$ per $i = 0, 1$ (per ipotesi infatti $y_1 \in O$ sse $y_2 \in O$ per ogni $O \in \mathcal{O}(Y)$). Questo contraddice la condizione 2 rispetto alla buona definizione di $invert^{-1} :\subseteq \mathcal{C}(\mathcal{O}(Y), \mathcal{O}(\{*\})), f^{-1} \mapsto f$ (si noti che nella condizione 2 è quindi implicita una separazione di tipo T_0).

Poniamo ora $X := Y_k$. Per il Corollario 4.40, $k_Y \in \mathcal{C}(\mathcal{O}(Y), \mathcal{O}(Y_k))$ è computabile (si noti che qui si prende l'estensione di k_Y agli insiemi). Poiché k_Y^{-1} è ben definita, si dà $k_Y = (k_Y^{-1})^{-1}$, e quindi per la condizione 2 possiamo ottenere $k_Y^{-1} : Y_k \to Y$. $\square$

La condizione 2. è di fatto l'inversa della Proposizione 4.20.6 che abbiamo invece visto valere per spazi rappresentati in generale.

Vediamo ora che non tutte le topologie finali sono ammissibili. Abbiamo in precedenza visto che $id : \mathcal{O}(\mathbb{R}_b) \to \mathcal{O}(\mathbb{R}), O \mapsto O$ è computabile. Questa funzione è l'inversa (rispetto agli insiemi aperti) di $id : \mathbb{R} \to \mathbb{R}_b, x \mapsto x$. Se $\mathbb{R}_b$ fosse ammissibile, allora anche $id : \mathbb{R} \to \mathbb{R}_b, x \mapsto x$ sarebbe computabile (o per lo meno continua) per il Teorema 4.41. Eppure abbiamo visto che questa funzione non è nemmeno realizer-continua.

Per completezza, vediamo che anche $id : \mathcal{O}(\mathbb{R}) \to \mathcal{O}(\mathbb{R}_b)$ è computabile. Per questo, mostriamo che $id : \mathbb{R}_b \to \mathbb{R}$ è computabile; il resto segue dalla proprietà di invert (Proposizione 4.20.6).

Sia quindi p un nome di $x \in \mathbb{R}_b$, e sia $sw_1, w_2 \sqsubseteq p$ per $w_2 = w_2(1)...w_2(|w_2|)$. Allora $x \in]\delta_b(sw_1, w_2 0^{\mathbb{N}}) - 2^{-|w_2|}, \delta_b(sw_1, w_2 1^{\mathbb{N}}) + 2^{-|w_2|}[$[6].

Pertanto anche $id : \mathcal{O}(\mathbb{R}) \to \mathcal{O}(\mathbb{R}_b), O \mapsto O$ è computabile, e in particolare ben definita, proprio come $id : \mathcal{O}(\mathbb{R}_b) \to \mathcal{O}(\mathbb{R}), O \mapsto O$. Concludiamo che gli aperti di $\mathcal{O}(\mathbb{R}_b)$ e quelli di $\mathcal{O}(\mathbb{R})$ coincidono, ovvero $\mathbb{R}_b$ e $\mathbb{R}_2$ hanno la stessa topologia. Inoltre, poiché $\mathcal{O}(\mathbb{R}_b) \simeq \mathcal{O}(\mathbb{R})$, abbiamo $(\mathbb{R}_b)_k \simeq \mathbb{R}_k \simeq \mathbb{R}$.

4.5 Computabilità sugli insiemi chiusi di $\mathbb{R}$

Definizione 4.42 (Insiemi chiusi r.e.) *Un insieme $A \in \mathcal{A}(\mathbb{R}^n)$ computabile è detto* co-ricorsivamente enumerabile *(*co-r.e.*).*

Vediamo che questo concetto di co-r.e. estende effettivamente il concetto di co-r.e. della Teoria della Computabilità classica.

Proposizione 4.43 *$A \subseteq \mathbb{N}$ è co-r.e. sse $A \subseteq \mathbb{R}$ è co-r.e.*

DIM.: $n \notin A \subseteq \mathbb{N}$ sse $\chi_{\mathbb{R} \setminus A}(n) = 1 \in \mathbb{S}$ per ogni $n \in \mathbb{N}$. □

Proviamo a generalizzare anche il concetto di r.e.

Definizione 4.44 (Insiemi chiusi manifesti ed r.e.) *Un insieme chiuso $A \subseteq \mathbb{R}^n$ è detto* manifesto *se la funzione* $\mathrm{Intersect}_A : \mathcal{O}(X) \to \mathbb{S}$ *tale che*

$$\mathrm{Intersect}_A(O) = 1 \Leftrightarrow A \cap O \neq \emptyset$$

è realizer-continua.

Sia $\mathcal{V}(X)$ lo spazio degli insiemi chiusi manifesti. Definiamo la rappresentazione $\delta_{\mathcal{V}(\mathbb{R}^n)}$ come:

$$\delta_{\mathcal{V}(\mathbb{R}^n)}(p) := A \Leftrightarrow [\delta_{\mathcal{O}(X)} \to \delta_{\mathbb{S}}](p) = \mathrm{Intersect}_A.$$

Un insieme $A \in \mathcal{V}(\mathbb{R}^n)$ computabile è detto ricorsivamente enumerabile (r.e.).

Proposizione 4.45 *$A \subseteq \mathbb{N}$ è r.e. sse $A \subseteq \mathbb{R}$ è r.e.*

DIM.: $n \in A \subseteq \mathbb{N}$ sse $A \cap \left]n - \frac{1}{2}, n + \frac{1}{2}\right[\neq \emptyset$ per ogni $n \in \mathbb{N}$. □

Definizione 4.46 (Insiemi chiusi ricorsivi) *Un insieme chiuso $A \subseteq \mathbb{R}^n$ computabile sia rispetto a $\delta_{\mathcal{A}(\mathbb{R}^n)}$ che rispetto a $\delta_{\mathcal{V}(\mathbb{R}^n)}$ è detto ricorsivo.*

Corollario 4.47 *$A \subseteq \mathbb{N}$ è ricorsivo sse $A \subseteq \mathbb{R}$ è ricorsivo.*

[6] Gli estremi dell'intervallo sono da considerarsi come numeri razionali a tutti gli effetti, cioè calcolati in quanto oggetti "finiti", e dovranno poi essere accoppiati mediante pseudocoppie.

Notiamo che per $\mathbb{R}^n$ i concetti di ricorsivo e decidibile si separano, in contrasto col Teorema 3.11 (dove ovviamente diciamo che un insieme $M \subseteq \mathbb{R}^n$ è decidibile se $\chi_M : \mathbb{R}^n \to \{0,1\}$ è computabile). Per vederlo, utilizziamo la nozione topologica di connessione:

Definizione 4.48 (Spazi connessi) *Uno spazio topologico X è* connesso *sse non è l'unione di due insiemi aperti disgiunti non vuoti.*

Ebbene, esistono infiniti insiemi chiusi ricorsivi in $\mathbb{R}$, basti pensare (ad esempio) a $\emptyset$, $\mathbb{R}$, e agli intervalli $[a,b]$, $[a,\infty[$, $]-\infty,a]$ per $a,b \in \mathbb{R}$ computabili. Vale invece il seguente:

Teorema 4.49 *Gli unici sottoinsiemi di $\mathbb{R}^n$ decidibili sono $\emptyset$ e $\mathbb{R}^n$.*

DIM.: È sufficiente osservare che qualsiasi funzione continua $f : \mathbb{R}^n \to \{0,1\}$ è necessariamente costante. Supponiamo infatti che qualche f continua non lo sia. Poiché $\{0,1\}$ ha la topologia discreta, $f^{-1}(\{0\})$ e $f^{-1}(\{1\})$ sono aperti in $\mathbb{R}$. Inoltre ovviamente $\mathbb{R} = f^{-1}(\{0\}) \cup f^{-1}(\{1\})$. Infine $f^{-1}(\{0\}) \neq \emptyset \neq f^{-1}(\{1\})$ per assunzione. Ma allora $\mathbb{R}^n$ sarebbe non connesso. Contraddizione, in quanto $\mathbb{R}^n$ è connesso. □

Pertanto avere utilizzato $\mathbb{S}$ nello studio degli spazi rappresentati anziché $\{0,1\}$ è veramente il meglio che si possa fare.

Riferimenti bibliografici

[1] Piergiorgio Odifreddi. *Classical Recursion Theory.* Elsevier. 1989

[2] Arno Pauly. On the topological aspects of the theory of represented spaces. *Computability* 5: 159–180. 2016

[3] Klaus Weihrauch. *Computable Analysis.* Springer. 2000

[4] *Handbook of Computability and Complexity in Analysis.* Springer. Forthcoming.