# PCP, ambiguità di una CFG e teorema di Rice

Nicola Gulmini

**Def. 1** (PCP)**.** Siano $A$ e $B$ due liste di stringhe tali che $|A| = |B|$.

$$A = w_1, \dots, w_k$$
$$B = x_1, \dots, x_k$$

e $w_i, x_i \in \Sigma^+$ con $|\Sigma| \geq 2$. Il problema di corrispondenza di Post è il seguente: dire se esitono $m \geq 1$ indici $i_1, \dots, i_m$ tali che

$$w_{i1} \cdot w_{i2} \cdots w_{im} = x_{i1} \cdot x_{i2} \cdots x_{im}.$$

In pratica ci stiamo chiedendo se esiste un modo di scegliere le stringhe di $A$ in modo tale che la loro concatenazione corrisponda a una analoga concatenazione in $B$.

**Esempio 1.** Istanza PCP con $\Sigma = \{0, 1\}$.

| | $A$ | $B$ |
|---|---|---|
| $i$ | $w_i$ | $x_i$ |
| 1 | 1 | 111 |
| 2 | 10111 | 10 |
| 3 | 10 | 0 |

Una soluzione è rappresentata dalla scelta $m = 4$: $i_1 = 2$, $i_2 = 1$, $i_3 = 1$, $i_4 = 3$. Si noti prima di tutto che le liste sono ordinate e gli indici scelti sono unici, inoltre possono essere effettuate scelte ripetute,

$$w_2 w_1 w_1 w_3 = x_2 x_1 x_1 x_3 = 101111110.$$

Sono soluzioni anche $w_2 w_1 w_1 w_3 w_2 w_1 w_1 w_3$ e tutti i vari multipli.

**Def. 2** (MPCP)**.** Il problema di Post modificato è uguale al PCP, con l'unica restrizione che la prima coppia di stringhe confrontate è vincolata a essere $w_1$, $x_1$. In pratica si cercano degli indici $i_1, \dots, i_m$ tali che

$$w_1 w_{i1} \dots w_{im} = x_1 x_{i1} \dots x_{im}.$$

Qui però è ammesso il caso $m = 0$ che prima non era possibile.

**Teo. 1.** $MPCP \leq PCP$.

*Proof.* No dim. □

Ora invece dimostriamo (informalmente) che PCP è un problema indecidibile, fornendo una riduzione strategica.

**Teo. 2.** $L_u \leq MPCP$.

Dato che $L_u \leq MPCP$ e $MPCP \leq PCP$ allora vale $L_u \leq PCP$, così abbiamo dimostrato che il problema di Post non è $REC$, anzi, sappiamo che è almeno strettamente $RE$.

*Proof.* No dim. □

## 0.1 Ambiguità di una CFG

Indichiamo con $enc(G)$ la codifica della grammatica libera dal contesto $G$. Il problema

$$L_{amb} = \{enc(A) |\ A \text{ è ambigua.}\}$$

è indecidibile. Ci chiediamo allora se $L_{amb} \in RE$.

Trasformiamo istanze di $PCP$ in istanze di $L_{amb}$ eseguendo di fatto una riduzione. Sia $(A, B)$ una istanza di PCP su $\Sigma$ con $A = w_1 \dots w_n$ e $B = x_1 \dots x_n$. Sua $G_A$ una grammatica così definita:

- simboli nonterminali $\{A\}$
- alfabeto $\Sigma \cup \{a_i |\ i \in [1, k]\}$ dove $a_i$ è un alias per la coppia $w_i, x_i$
- insieme di produzioni $A \to w_1Aa_1 | \dots | w_kAa_k | w_1a_1 | \dots | w_ka_k$

Le stringhe derivate da $G_A$ hanno la forma $w_{i1} \dots w_{im}a_{im} \dots a_{i1}$ con $m \geq 1$.

Facciamo una cosa analoga con $G_B$ e le $x_i$. Osserviamo quindi che $G_A$ e $G_B$ non sono abimbue. Definiamo $L_A = L(G_A)$ e analogamente $L_B$. $G_{AB}$ è la CFG che genera $L_A \cup L_B$: i nonterminali sono $\{S, A, B\}$, l'alfabeto è lo stesso e le produzioni sono $S \to A|B$ dove $A$ e $B$ sono rispettivamente tutte le produzioni di $G_A$ e $G_B$.

**Teo. 3.** $PCP \leq L_{amb}$.

*Proof.* Per la precedente trasformazione si ha $enc(G)$ ambigua se e solo se $(A, B)$ ha soluzione. □

Intuitivamente produco una stringa con ciascuna grammatica, applico PCP e se trovo una corrispondenza allora ho trovato due derivazioni da $G_{AB}$ diverse per la medesima stringa.

## 0.2 Altri problemi indecidibili

Siano $G_1, G_2$ CFG e $R$ espressione regolare, allora i seguenti problemi sono indecidibili:

- $L(G_1) \cap L(G_2) = \varnothing$;
- $L(G_1) = L(G_2)$;
- $L(G_1) = L(R)$;
- $L(G_1) = T^*$ per un alfabeto $T$ assegnato;
- $L(G_1) \subseteq L(G_2)$;
- $L(R) \subseteq L(G_2)$.

# 1 Considerazioni sul teorema di Rice

**Def. 3** (Proprietà)**.** Una proprietà $\mathcal{P}$ dei linguaggi è definita come l'insieme dei linguaggi che la soddisfano:

$$\mathcal{P} = \{L \mid L \text{ soddisfa } \mathcal{P}\}.$$

**Def. 4** (Proprietà banale)**.** Sia $\mathcal{P}$ una proprietà dei linguaggi $RE$:

$$\mathcal{P} = \{L \in RE \mid L \in \mathcal{P}\} = \{L \in RE \cap \mathcal{P}\}$$

$\mathcal{P}$ è banale se è soddisfatta da nessun linguaggio oppure da tutti.

In caso contrario, cioè nel caso esistesse un $L' \in RE$ che la soddisfi e un $L'' \in RE$ che non la soddisfi, allora $\mathcal{P}$ è non banale.

**Prop. 1.** $\mathcal{P}$ è banale se $\mathcal{P} = \varnothing$ oppure $\mathcal{P} = RE$. Si noti che $\mathcal{P} = \varnothing$ se e solo se $|\mathcal{P}| = 0$.

Se $\mathcal{P} = \{\varnothing\}$ si ha $|\mathcal{P}| = 1$, cioè la proprietà è soddisfatta da un solo linguaggio, che è quello vuoto. In questo caso $\mathcal{P}$ è la proprietà di essere il linguaggio vuoto ed è non banale.

**Def. 5.** Definiamo

$$L_{\mathcal{P}} = \{enc(M_i) \mid L(M_i) \in \mathcal{P}\}$$

il linguaggio delle codifiche delle mdT che accettano tutti e soli i linguaggi che soddisfano $\mathcal{P}$.

Il motivo per il quale si passa da questa definizione è che i linguaggi sono tipicamente insiemi di cardinalità infinita, perciò difficili da trattare. Le macchine di Turing che li accettano, però, possono essere rappresentate come stringhe (finite in quanto tali). Esiste perciò un isomorfismo tra linguaggi e stringhe (quindi tra sottoinsiemi di $\Sigma^*$ ed elementi di $\Sigma^*$):

$$L_i \subseteq \Sigma^* \longleftrightarrow M_i \mid L(M_i) = L_i \longleftrightarrow enc(M_i) \in \Sigma^*$$

che è un modo formale per relazionare un problema (inteso come insieme delle istanze del problema stesso) con il programma che lo risolve (cioè che risolve qualsiasi istanza di esso). Si ha $|\mathcal{P}| = |L_{\mathcal{P}}|$ dato l'isomorfismo, solo che gli elementi di $\mathcal{P}$ sono a loro volta insiemi di cardinalità infinita mentre quelli di $L_{\mathcal{P}}$ sono stringhe. Nulla si può dire invece sulla cardinalità di $\mathcal{P}$: può essere finita ma anche infinita, come vedremo in un esempio.

**Teo. 4** (Rice)**.** Ogni proprietà non banale dei linguaggi $RE$ è indecidibile.

*Proof.* No dim. □

In altre parole non esiste una $M$ che si arresta sempre tale che $L(M) = L_{\mathcal{P}}$ se $\mathcal{P}$ è non banale. Bisogna però fare attenzione a non confondersi: data una stringa o un sottoinsieme di $L_P$ si possono esibire delle mdT che si arrestano sempre.

**Esempio 2.** Sia $\mathcal{P}$ la proprietà di un linguaggio $RE$ di essere regolare: $\mathcal{P} = Reg \cap RE = Reg$. Allora il linguaggio $L_{\mathcal{P}}$ è il linguaggio di tutte e sole le mdT che accettano i linguaggi regolari. Si noti che in questo caso $|\mathcal{P}| = |L_{\mathcal{P}}| = |Reg| = +\infty$. É immediato concludere che per Rice una mdT che accetta $L_{\mathcal{P}}$ si arresta solo in caso di accettazione: in pratica un compilatore per i programmi che riconoscono un linguaggio regolare si arresta solo se il programma è corretto.

**Esempio 3.** Sia $\mathcal{P}$ la proprietà di un linguaggio $RE$ di essere finito: $L_{\mathcal{P}}$ è il linguaggio di tutte e sole le mdT che accettano i linguaggi finiti. In questo caso sia gli elementi di $\mathcal{P}$, sia quelli di $L_{\mathcal{P}}$ sono finiti, pur essendo $|\mathcal{P}|$ e $|L_{\mathcal{P}}| = +\infty$. Anche qui Rice permette di concludere che il linguaggio delle mdT che accettano i linguaggi finiti non è ricorsivo (accetta ma non rifiuta).