

Quello che ci proponiamo di fare adesso
è di mostrare una serie di problemi che
sono algoritmica mente iusolubili, cioè che
non ammettono un algoritmo generale
di soluzione.

Li limiteremo ad enunciare i risultati
senza darne le dimostrazioni relative in
questo momento.

Definizione

Il problema delle parole per un processo
di (semi)Thue Π è il problema di determinare
per ogni coppia data di parole u, v
sull'alfabeto di Π se $u \xrightarrow[\Pi]^* v$.

Si dimostra che :

Esiste un processo di (semi)Thue su un
alfabeto di due simboli il cui problema
delle parole è iusolubile. Inoltre
tutte le regole di riscrittura $g \rightarrow h$
di tale processo sono tali che $g, h \neq 0$.

SISTEMI DI CORRISPONDENZA DI POST

Un sistema di corrispondenza di Post è dato da un alfabeto finito A ed un insieme finito di coppie di parole (w_i, \bar{w}_i) su A ($1 \leq i \leq m$).

SI DICE CHE IL SISTEMA AMMETTE UNA SOLUZIONE SE ESISTE UNA SUCCESSIONE FINITA DI INDICI DI COPPIE

i_1, i_2, \dots, i_k , $1 \leq i_j \leq m$ tali che

$$w_{i_1} w_{i_2} \dots w_{i_k} \equiv \bar{w}_{i_1} \bar{w}_{i_2} \dots \bar{w}_{i_k} \quad (*)$$

Una parola come quella che compare in $(*)$ si dice PAROLA SOLUZIONE. DEL SISTEMA DI CORRISPONDENZA DI POST.

Una visualizzazione efficace si può ottenere con una rappresentazione che faccia uso di tessere di domino sulle quali sono stampate le due parole delle coppia.

Quindi (w_1, \bar{w}_1) viene rappresentata come

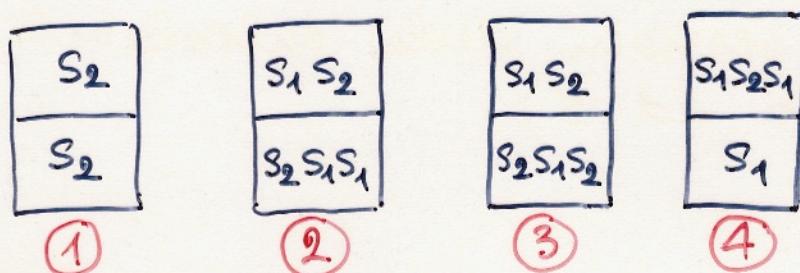
w_1
\bar{w}_1

Consideriamo un esempio.

Sia il seguente sistema di Post

(S_2, S_2) ; $(S_1S_2, S_2S_1S_1)$; $(S_1S_2, S_2S_1S_2)$; $(S_1S_2S_1, S_1)$

Una rappresentazione tipo-domino è
la seguente:



Il problema delle corrispondenze di Post
consiste nel mettere uno accanto all'altro
le tessere (eventualmente con ripetizioni)
in modo tale che le parole che si
legge nella traccia superiore è identica
a quelle che si legge nella traccia inferiore.

LA SEGUENTE SUCCESSIONE DI TESSERE,
FORNISCE UNA SOLUZIONE: 4-2-1-3.

In fatti:

S ₁ S ₂ S ₁	S ₁ S ₂	S ₂	S ₁ S ₂
S ₁	S ₂ S ₁ S ₁	S ₂	S ₂ S ₁ S ₂

In realtà una soluzione si ha già utilizzando
solo tre tessere:

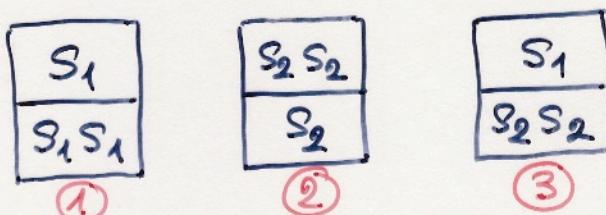
S ₁ S ₂ S ₁	S ₁ S ₂	S ₁ S ₂
S ₁	S ₂ S ₁ S ₁	S ₂ S ₁ S ₂

il che mostra che
le tessere \boxed{a} sono superflue

Consideriamo un altro esempio:
Sia il seguente sistema di corrispondenza di Post:

$$(S_1, S_1 S_1); (S_2 S_2, S_2); (S_1, S_2 S_2)$$

rappresentato da:



È facile convincersi che non esiste alcuna soluzione che fa uso di due sole fessure e neanche di tutte e tre le fessure usate una sola volta (la parola di sopra sarebbe di lunghezza 4, quelle di sotto di lunghezza 5).

Dobbiamo quindi necessariamente usare almeno una fessura più di una volta.

Una soluzione al problema è fornita da:

S_1	S_1	$S_2 S_2$	$S_2 S_2$
$S_1 S_1$	$S_2 S_2$	S_2	S_2
(1)	(3)	(2)	(2)

La ricerca di una soluzione del sistema
di Post si chiama:

PROBLEMA DELLA CORRISPONDENZA DI POST

Post stesso nel 1946 ha dimostrato che
IL PROBLEMA DELLA CORRISPONDENZA È
INDECIDIBILE, CIOÈ NON ESISTE NESSUN
ALGORITMO IN GRADO DI DECIDERE
SE UN ARBITRARIO SISTEMA DI CORRISPONDENZA
DI POST AMMETTE O NO UNA SOLUZIONE.

GRAMMATICHE

- Non esiste nessun algoritmo per determinare, data una grammatica T' , se
 - $L(T') = \emptyset$
 - $L(T')$ è infinito
 - data una parola v , se $v \in L(T')$
- Non esiste nessun algoritmo per determinare, date due grammatiche T'_1 e T'_2 , se
 - $L(T'_1) \subseteq L(T'_2)$
 - $L(T'_1) = L(T'_2)$

LA GERARCHIA DI CHOMSKY

Esamineremo adesso il problema di fornire modelli di calcolo, via via più specifici, sempre nel contesto dei processi di Seuni-Thue.

Questo può essere fatto imponendo delle restrizioni via via più forti ad una grammatica a struttura di frase che verrà indicata in questo contesto come grammatica di tipo 0.

UNA GRAMMATICA T È DETTA DI TIPO 1
O GRAMMATICA CONTESTUALE SE, PER
CIASCUNA PRODUZIONE $g \rightarrow \bar{g}$ DI T SI HA
CHE $|g| \leq |\bar{g}|$.

UNA GRAMMATICA T È DETTA DI TIPO 2
O LIBERA DA CONTESTO SE CIASCUNA
PRODUZIONE È DEL TIPO $X \rightarrow h$, DOVE
 X È UNA VARIABILE ED h UNA QUALSIASI
PAROLA SULL'ALFABETO TOTALE.

(In generale h può essere la parola vuota; se
nessuna produzione di T è tale che $h = \emptyset$ allora
 T si dice GRAMMATICA LIBERA DA CONTESTO POSITIVA)

UNA GRAMMATICA T È DETTA DI TIPO 3
O REGOLARE SE CIASCUNA DELLE SUE
PRODUZIONI È DEL TIPO $U \rightarrow aV$ OPPURE
 $U \rightarrow a$, CON U E V VARIABILI ED a SIMBOLO
TERMINALE

A ciascun tipo di grammatica corrisponde una classe di linguaggi generati.

Si ha che :

Un linguaggio L è regolare SE esiste una grammatica regolare T tale che
(di tipo 3) $L = L(T)$ o $L = L(T) \cup \{0\}$

Un linguaggio L è libero da contesto (tipo 2) SE esiste una grammatica libera da contesto tale che $L = L(T)$ o $L = L(T) \cup \{0\}$

Un linguaggio L è contestuale (tipo 1) SE esiste una grammatica contestuale tale che $L = L(T)$ o $L = L(T) \cup \{0\}$.

Un linguaggio L SE esiste una grammatica a struttura di frase T che lo genera.
e. r. l. (o di tipo 0)

Semplicemente guardando le definizioni date si vede che :

- Ogni linguaggio regolare è libero da contesto
- Ogni linguaggio libero da contesto è contestuale
- Ogni linguaggio contestuale è r. l.

Si ha allora l'inclusione

$$L_3 \subset L_2 \subset L_1 \subset L_0$$

avendo indicato con L_i ciascuna famiglia di linguaggi.

Le inclusioni sono tutte in senso stretto come è, correttamente, indicato, in quanto è possibile postare esempi di linguaggi che appartengono alla classe più vasta ma non a quella immediatamente inferiore nella gerarchia.

Passiamo, molto rapidamente, in rassegna, alcuni risultati riguardanti la decidibilità o l'indecidibilità:

- ESISTE UN ALGORITMO IN GRADO DI DETERMINARE, DATA UNA GRAMMATICA LIBERA DA CONTESTO T' ED UNA PAROLA u , SE u APPARTIENE O NO AD $L(T')$.
- ESISTE UN ALGORITMO PER DECIDERE, DATA UNA GRAMMATICA LIBERA DA CONTESTO T' , SE $L(T')$ È O NON È VUOTO.
- ESISTE UN ALGORITMO PER DECIDERE, DATA UNA GRAMMATICA LIBERA DA CONTESTO T' , SE $L(T')$ È FINITO O INFINTO

La situazione, però, non è del tutto felice neanche qui.

Si ha infatti che:

- NON ESISTE NESSUN ALGORITMO PER DETERMINARE, DATE DUE GRAMMATICHE LIBERE DA CONTESTO T_1 e T_2 , SE

$$L(T_1) \cap L(T_2) = \emptyset$$

- NON ESISTE NESSUN ALGORITMO PER DECIDERE SE UNA DATA GRAMMATICA LIBERA DA CONTESTO È AMBIGUA.

(Una grammatica libera da contesto T' si dice ambigua se esiste almeno una parola appartenente al linguaggio generato da T' che può essere ottenuta mediante due derivazioni di un certo tipo standard)

Per arricchire il quadro dei problemi che è possibile porre nell'ambito di una pura teoria della calcolabilità, presentiamo ora (per il momento senza dimostrazione) altri due teoremi, uno di tipo positivo ed uno di tipo negativo.

I. ESISTE UN NUMERO e TALE CHE LA FUNZIONE DI UNA SOLA VARIABILE CALCOLATA DAL PROGRAMMA NUMERO e SI COMPORTA COME SEGUE:

$$\Xi_e(x) = e$$

CIOÈ, QUALUNQUE SIA LA VARIABILE D'INGRESSO DA' COME USCITA e .

Tale funzione sembrerebbe essere soltanto una funzione costante. La sua novità risiede nel fatto che, poiché nella nostra teoria i numeri sono anche codici di programmi, poniamo interpretare tale codice e sotto lineare il fatto che TALE FUNZIONE "REAGISCE A QUALSIASI STIMOLO ESTERNO" RIPROPONENDO IL SUO STESSO CODICE.

Questo aspetto "autoprodottoivo" è potenzialmente interessante sia in direzione tecnologico (avere una teoria degli automi in grado di produrre copie di se stessi) sia in direzione biologica collegandosi, in particolare ad impostazioni teoriche che assicurano la specificità dei sistemi biologici alla loro tendenza ad automantenersi se stessi.

II.

Sia adesso T' un insieme di funzioni parzialmente calcolabili di una sola variabile.

Chiamiamo insieme di indici di T' l'insieme:

$$R_{T'} = \{t \in N \mid \Phi_t \in T'\}$$

Teorema (Rice)

$R_{T'}$ è ricorsivo solo nel caso in cui T' è l'insieme vuoto oppure è l'insieme di tutte le funzioni parzialmente calcolabili.

ovvero (in modo equivalente)

Sia T' un insieme di funzioni parzialmente calcolabili di una variabile. Se esistono due funzioni parz. calcolabili f e g tali che $f \in T'$ e $g \notin T'$ allora $R_{T'}$ non è ricorsivo.

L'idea alleata ~~alla ricerca di un risultato~~ è quella di fissare una qualsiasi proprietà delle funzioni e poi chiedere una verifica "meccanica" di tale proprietà per una specifica funzione.

Il teorema di Rice ci dice che tale idea, per quanto allietante, non è realizzabile.