Indice

1 Ambiguità

Prendiamo il linguaggio

$$\mathcal{L} = \{a^p b^q c^r \mid p = q \vee q = r\}$$

sappiamo gli automi pre $a^n b^n c^m$ e per $a^m b^n c^n$, quindi nondeterministicamente all'inizio scegliamo quale strada prendere nell'automa. Lo stesso si può fare con una grammatica

$$S \to S'C \mid AS''$$

dove S' genera $a^n b^n$

$$S' \to aS'b \mid \varepsilon$$

e S'' genera $b^n c^n$

$$S'' \to bS''c \mid \varepsilon$$

e

$$C \to cC \mid \varepsilon$$
$$A \to aA \mid \varepsilon$$

vediamo il caso particolare e questo corrisponde alla stringa $a^2b^2c^2$. ma anche questo corrisponde alla stringa $a^2b^2c^2$. Entrambi gli alberi rappresentano derivazioni leftmost che generano la stessa stringa.

Una grammatica viene detta *ambigua* se esiste almeno una stringa che può essere generata in due modi diversi. Chiamiamo il *grado di ambiguità* di una stringa $w \in \Sigma^*$ in G è uguale al numero di alberi di derivazione di w in G. Ed il grado di ambiguità di G è il massimo grado di ambiguità generato dalle stringhe della grammatica.

Non sempre si può trovare una grammatica non ambigua per un linguaggio. Esistono linguaggi deti *inerentemente ambigui*, cioè per cui ogni grammatica di tipo 2 che li genera sarà ambigua.

Utilizzando il lemma di Ogden mostreremo che il linguaggio L è interentemente ambiguo.

Mostriamo prima una versione alternativa del lemma

Lemma 1 (Lemma di Ogden)**.** *Sia G una grammatica CF e sia $L = L(G)$. Esiste una costante N tale che per ogni $z \in L$ con almeno N posizioni marcate possiamo decomporre z in 5 parti*

$$z = uvwxy$$

tali che

1. *vwx contiene al più N posizioni marcate*
2. *vx contiene almeno una posizione marcata*
3. *esiste una variabile $A \in V$ tale che $S \overset{*}{\Rightarrow} uAy$, e $A \overset{*}{\Rightarrow} vAx$ e $A \overset{*}{\Rightarrow} w$. E dunque per ogni $i \geq 0$ $uv^iwx^iy \in L$.*

Noi il teorema di Ogden lo abbiamo dimostrato utilizzando grammatiche in CNF, si può fare anche per grammatiche generiche, ma è più tedioso.

Dimostrazione. Noi vogliamo dimostrare che

$$\mathcal{L} = \{a^p b^q c^r \mid p = q \vee q = r\}$$

è inerentemente ambiguo. Sia G una qualunque grammatica CF per $\mathcal{L}$.

Sia N la costante del lemma di Ogden e $m = \max(N, 3)$ il massimo tra N e 3. Prendiamo

$$z = a^m b^m c^{m+m!} \in \mathcal{L}$$

e marchiamo tutte le a.

Prendiamo la stringaα ottenuta ponendo $i = 2$, cioè la stringa $uv^2wx^2y \in \mathcal{L}$. Contiamo le b in α

$$\begin{aligned}
\#_b(\alpha) &= \underbrace{\#_b(z)}_{m} + \underbrace{\#_b(vx)}_{\leq m} \\
&\leq 2m \\
&< m + m! && \text{(Visto che } m \text{ è almeno 3)} \\
&\leq \#_c(\alpha) && \text{(Le } c \text{ rimaste sono almeno quelle che c'erano prima)}
\end{aligned}$$

e visto che $uv^2w^{x^2}y \in \mathcal{L}$ allora $\#_b(\alpha) = \#_a(\alpha)$ e per la proprietà 2 del lemma di Ogden $\#_a(vx) = \#_b(vx) \geq 1$. Inoltre affinché $uv^2wx^2y \in \mathcal{L}$ sia ancora in $\mathcal{L}$ deve valere che

$$\begin{aligned}
v &= a^l \\
x &= b^l
\end{aligned}$$

con $1 \leq l \leq m$. Altrimenti ripetendo perderemmo la struttura.

Prendiamo ora

$$uv^iwx^iz = a^{m+(i-1)l}b^{m+(i-1)l}c^{m+m!} \in \mathcal{L}$$

scegliendo $i = 1 + \frac{m!}{l}$ otteniamo la stringa

$$\beta = a^{m+m!}b^{m+m!}c^{m+m!}$$

che appartiene al nostro linguaggio. Nella stringa β la maggior parte delle b è ottenuta replicando il secondo sottoalbero.

Partiamo invece da una stringa

$$z' = a^{m+m!}b^mc^m$$

e marco tutte le c. Scomponiamola in

$$u'v'w'x'y'$$

e utilizzando i ragionamenti di prima mostriamo che $v' = b^j$ e $x' = c^j$ e ripetendo i volte v' e x' otteniamo

$$a^{m+m!}b^{m+(i-1)j}c^{m+(i-1)j}$$

e scegliendo $i = 1 + \frac{m!}{j}$ come prima otteniamo β. Come prima possiamo immaginare alberi di forma Qui ancora la maggior parte delle b è generata dal secondo albero. Ma visto che l'albero di A genera anche a e l'albero di A' genera anche c i due sono per forza diversi. ■

Possiamo dare una definizione analoga di ambiguità per gli automi a pila.

Definizione 1. *Un PDA è ambiguo se esiste una stringa con due computazioni accettanti differenti.*

È facile vedere attraverso la trasformazione da grammatica ad automa che il numero di computazioni diverse è uguale al numero di alberi leftmost diversi, visto che si simulava la derivazione leftmost. Dal lato opposto è più difficile da mostrare e soprattutto non preserva il grado di ambiguità. Quindi parlare di ambiguità negli automi a pila e nelle grammatiche di tipo 2 è equivalente. E di conseguenza un linguaggio inerentemente ambiguo avrà sia grammatica che automa a pila ambigui.

Perché un automa a pila ammetta ambiguità questo deve per forza essere nondeterministico. Quindi

$$\mathcal{L} = \{a^p b^q c^r \mid p = q \vee q = r\}$$

necessita il nondeterminismo.

Richiamiamo la definizione di automa a pila deterministico

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$$

è deterministico se ad ogni passo possiamo fare una singola scelta, cioè sse

- $\forall q \in Q, A \in \Gamma \mid \delta(q, \varepsilon, A) \neq \varnothing \Rightarrow \forall a \in \Sigma \delta(q, a, A) = \varnothing$
- $\forall q \in Q, A \in \Gamma, a \in \Sigma \cup \{\varepsilon\} \mid |\delta(q, a, A)| = 1$

mostriamo ora che i due criteri di accettazione sono diversi per automi deterministici, specificamente la accettazione per pila vuota è più debole. Supponiamo infatti di avere un automa a pila che accetta per pila vuota il linguaggio regolare $(aa)^+$. Siccome l'automa deve accettare aa dopo questa avrà la pila vuota. Ma da una pila vuota non può più continuare, quindi non può accettare $aaaa$ ad esempio. Quindi i linguaggi che accetta sono tutti quelli le cui stringhe non hanno come prefissi altre stringhe del linguaggio, perché nel riconoscere il prefisso svuoterà la pila e non potrà più andare avanti. Questi in teoria dei codici sono detti codici prefissi e contengono alcuni regolari e alcuni context free.

Di solito si ovvia a questo utilizzando un simbolo finale non nel linguaggio, ad esempio $(aa)^+\#$, in modo tale che la pila non si svuoti completamente fino ad arrivare al marcatore.

Quindi da ora in poi quando parliamo di DCFL, cioè i linguaggi CF deterministici, intendiamo i linguaggi riconosciuti per stati finali. Questi contengono per forza i linguaggi regolari, perché semplicemente possiamo non utilizzare la pila. Vale che

$$\text{Reg} \subset \text{DCFL} \subset \text{CFL}$$

perché $\{a^p b^q c^r \mid p = q \vee q = r\} \in \text{CFL}$ ma $\notin \text{DCFL}$.

Abbiamo visto che l'ambiguità necessità del nondeterminismo. Possiamo chiederci l'inverso, cioè se un linguaggio necessita del nondeterminismo allora questo è per forza ambiguo. Un linguaggio che necessita del nondeterminismo

	CFL	DCFL
Unione	Sì	No
Intersezione	No	No
Intersezione con un regolare	Sì	Sì
Complemento	No	Sì

Tabella 1: Chisura delle operazioni

è quello dei palindromi (non ancora dimostrato), o – per semplicità – dei palindromi pari

$$L = \{ww^R \mid w \in \{a, b\}^*\}$$

L'automa deve "scommettere" su quando è arrivato a metà , quindi il nondeterminismo è necessario, ma la metà è una sola, qiundi non è ambiguo. È anche facile vederlo dalla grammatica che è

$$S \to aSa \mid bSb \mid \varepsilon$$

Quindi il nondeterminismo non implica la ambiguità.

Il determinismo per le grammatiche è più strano.

2 Operazioni e chiusura con i linguaggi CF

Unione Date due grammatiche $G' = \langle V', \Sigma, P', S' \rangle$ e $G'' = \langle V'', \Sigma, P'', S'' \rangle$ con $V' \cap V'' = \varnothing$, definiamo la unione delle due come

$$G = \langle V' \cup V'' \cup \{S\}, \Sigma, P' \cup P'' \cup \{S \to S', S \to S''\}, S \rangle$$

e nel caso degli automi possiamo fare la scelta all'inizio. La unione è chiusa per i CFL e non chiusa per i CDFL, infatti dati $L' = \{a^n b^n c^m\} \in$ DCFL e $L'' = \{a^m b^n c^n\} \in$ DCFL la loro unione $L' \cup L'' = \{a^p b^q c^r \mid p = q \vee q = r\}$ è ambigua, quindi nondeterministica.

Intersezione La intersezione non è chiusa, infatti $L' \cap L'' = \{a^n b^n c^n\}$ che abbiamo dimostrato che non è CF. Inoltre non possiamo utilizzare il metodo dei FSA di far andare in parallelo i due automi, qui i due due automi interferirebbero nel loro utilizzo della pila.

Intersezione con un regolare Possiamo usare il metodo degli FSA di far andare in parallelo i due automi, quindi incorporo nel controllo del PDA l'FSA.

Complemento Se il complemento fosse chiuso, potremmo ottenere una intersezione chiusa attraverso l'unione e De Morgan, quindi il complemento deve per forza non essere chiuso. Il complemento per i deterministici invece è chiuso.