Non regolarità

Corso di Fondamenti di Informatica - modulo 1
Corso di Laurea in Informatica
Università di Roma "Tor Vergata"
a.a. 2020-2021

Giorgio Gambosi

Pumping lemma

Ogni stringa sufficientemente lunga appartenente ad un linguaggio regolare presenta delle regolarità: in particolare, contiene una sottostringa che può essere ripetuta quanto si vuole, ottenendo sempre stringhe del linguaggio.

Più precisamente:

sia L un linguaggio regolare : allora $\exists n > 0$ tale che per ogni $\forall z \in L : \mid z \mid \geq n$ possiamo scrivere $z = uvw$, con $\mid uv \mid \leq n$, $\mid v \mid \geq 1$ e ottenere che $\forall i \geq 0$, $uv^i w \in L$.

Pumping lemma: interpretazione come gioco a due

Se L è regolare, Alice vince sempre questo gioco con Bob:

1. Alice fissa un intero $n > 0$ opportuno
2. Bob sceglie una stringa $z \in L$ con $\mid z \mid > n$
3. Alice divide z in tre parti uvw con $\mid uv \mid \leq n$ e $\mid v \mid \geq 1$
4. Bob sceglie un intero $i \geq 0$
5. Alice mostra a Bob che $uv^i w \in L$

Pumping lemma: dimostrazione

Se L è regolare, sia $\mathcal{A}$ l'ASFD che lo decide e che ha il minimo numero n di stati.

Una stringa $z \in L$ di lunghezza $m \geq n$ in input a $\mathcal{A}$ gli fa eseguire m transizioni e quindi attraversare $m + 1 > n$ stati, quindi esiste almeno uno stato che viene attraversato più volte.

Pumping lemma: dimostrazione

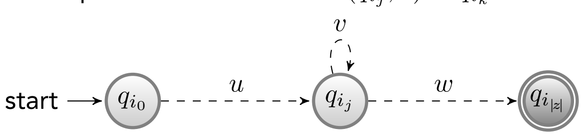
Indichiamo con $q_{i_0}, q_{i_1}, \ldots, q_{i_{m+1}}$ la sequenza di stati, non tutti distinti, attraversati (chiaramente, $q_{i_0} = q_0$ e $q_{i_{m+1}} \in F$) e con q_{i_j} il primo stato della sequenza che ricompare in seguito, ad esempio come q_{i_k}.

Sia u il prefisso (eventualmente nullo) di z tale che $\overline{\delta}(q_0, u) = q_{i_j}$ e sia $z = ux$, per cui $\overline{\delta}(q_{i_j}, x) = q_{i_{m+1}}$.

Pumping lemma: dimostrazione

$\mathcal{A}$ quindi esegue una sotto-computazione $q_{i_j}, q_{i_{j+1}}, \ldots, q_{i_k}$ di una computazione di accettazione (di z) che inizia e termina nello stesso stato. Si noti che la sotto-sequenza deve prevedere almeno due stati, per cui $q_{i_k} - q_{i_j} > 1$. Si noti inoltre che non possono essere stati attraversati più di n stati prima di arrivare a q_{i_k}, perché altrimenti questo non sarebbe il primo stato a comparire di nuovo.

Sia v il prefisso di x tale che $\overline{\delta}(q_{i_j}, v) = q_{i_k}$ e sia $x = vw$: da quanto detto, $\mid uv \mid \leq n$ e $\overline{\delta}(q_{i_j}, w) = q_{i_{|z|}}$



Pumping lemma: dimostrazione

Una computazione in cui questa sotto-sequenza è eseguita più volte è ancora una computazione di accettazione.

Per ogni $i \geq 0$ abbiamo infatti

$$\begin{aligned}\overline{\delta}(q_0, uv^i w) &= \overline{\delta}\Big(\overline{\delta}(q_0, u), v^i w\Big) = \overline{\delta}\left(q_{i_j}, v^i w\right) = \overline{\delta}\Big(\overline{\delta}(q_{i_j}, v), v^{i-1} w\Big) \\ &= \overline{\delta}\left(q_{i_j}, v^{i-1} w\right) = \ldots = \overline{\delta}(q_{i_j}, w) = q_{i_{m+1}} \in F\end{aligned}$$

il che mostra che ogni stringa del tipo $uv^i w$ appartiene ad L.

Pumping lemma

Evidenzia il fatto che gli automi finiti: non possono contare. Il numero di situazioni diverse che possono memorizzare è finito.

Fornisce soltanto una condizione necessaria perché un linguaggio sia regolare: non può essere utilizzato per mostrare la regolarità di un linguaggio, ma solo per dimostrarne la non regolarità.

L regolare $\implies$ pumping lemma verificato

pumping lemma non verificato $\implies$ L non regolare

Pumping lemma

Sia L un linguaggio e supponiamo che $\forall n > 0$ si abbia che $\exists z \in L :\mid z \mid \geq n$ tale che comunque dividiamo z in $z = uvw$, con $\mid uv \mid \leq n$, $\mid v \mid \geq 1$, $\exists i \geq 0$ tale che $uv^i w \notin L$. Allora, L non è regolare.

Pumping lemma: interpretazione come gioco a due

Se L non è regolare, Alice vince sempre questo gioco con Bob:

1. Bob fissa un intero $n > 0$
2. Alice sceglie una stringa opportuna $z \in L$, con $\mid z \mid > n$
3. Bob divide z in tre parti uvw con $\mid uv \mid \leq n$ e $\mid v \mid \geq 1$
4. Alice sceglie un intero $i \geq 0$ e mostra a Bob che $uv^i w \notin L$

Esempio

Consideriamo il linguaggio $L = a^k b^k$, $k > 0$: per mostrare che L non è regolare, interpretiamo il ruolo di Alice nel gioco.

1. Bob fissa un intero $n > 0$
2. Scegliamo la stringa $z = a^n b^n$
3. Bob divide z in tre parti uvw con $\mid uv \mid \leq n$ e $\mid v \mid \geq 1$: per la struttura di z, necessariamente $uv = a^h$, con $0 < h \leq n$. Quindi, $v = a^l$, per $0 < l < h$, e corrispondentemente $u = a^{h-l}$; inoltre, $w = a^{n-h} b^n$.
4. Scegliamo l'intero 2 e mostriamo a Bob che

$$uv^2 w = a^{h-l} a^l a^l a^{n-h} b^n = a^{n+l} b^n \notin L$$

Esempio

Si consideri il linguaggio $L = \{w\tilde{w} \mid w \in \{a, b\}^*\}$, ove si è indicata con $\tilde{w}$ la stringa ottenuta invertendo i caratteri presenti in w.

Dimostrare, utilizzando il pumping lemma, che tale linguaggio non è regolare.

Esempio

Interpretiamo il ruolo di Alice nel gioco.

1. Bob fissa un intero $n > 0$
2. Scegliamo la stringa $z = a^n bba^n$
3. Bob divide z in tre parti uvw con $\mid uv \mid \leq n$ e $\mid v \mid \geq 1$: per la struttura di z, necessariamente $uv = a^h$, con $0 < h \leq n$. Quindi, $v = a^l$, per $0 < l < h$, e corrispondentemente $u = a^{h-l}$; inoltre, $w = a^{n-h} bba^n$.
4. Scegliamo l'intero 2 e mostriamo a Bob che

$$uv^2 w = a^{h-l} a^l a^l a^{n-h} bba^n = a^{n+l} bba^n \notin L$$