Pumping Lemma - Slide

Gabriel Rovesti

Testo degli Esercizi

1. $L_{3n} = \{\, 1^{3n+2} \mid n \geq 0\}$: stabilire se è regolare.
2. $L_{mn} = \{\, 0^n\, 1^m\, 0^n \mid m + n > 0\}$: stabilire se è regolare.
3. $L_{mnp} = \{\, 0^n\, 1^m\, 0^p \mid m + n + p > 0\}$: stabilire se è regolare.
4. $L_{2ab} = \{\, w \in \{a, b\}^* \mid \#a(w) = 2 \times \#b(w)\}$: stabilire se è regolare.

Soluzioni

1) $L_{3n} = \{\, 1^{3n+2} \mid n \geq 0\}$

Soluzione

Osserviamo che il linguaggio L_{3n} è:

$$L_{3n} = \{\, 1^2, 1^5, 1^8, 1^{11}, \dots \},$$

cioè tutte le stringhe di 1 la cui lunghezza è congruente a 2 modulo 3.
Dimostrazione di regolarità: Possiamo costruire un automa a stati finiti (anche deterministico) che lavora sui resti mod 3 della lunghezza della stringa letta. In particolare, bastano 3 stati per ricordare il resto. Lo stato finale sarà quello associato al resto "2". Ad esempio:

- Stato q_0: resto $= 0$ (iniziale)
- Stato q_1: resto $= 1$
- Stato q_2: resto $= 2$ (finale)

Quando leggiamo un simbolo "1", passiamo da q_0 a q_1, da q_1 a q_2, da q_2 a q_0; così ciclicamente. Lo stato finale è q_2, giacché accettiamo solo se la lunghezza totale è 2 (mod 3). L'insieme di tali stringhe è quindi regolare.

2) $L_{mn} = \{\, 0^n\, 1^m\, 0^n \mid m + n > 0\}$

Soluzione

Questo linguaggio contiene stringhe della forma "un blocco di 0, poi un blocco di 1, poi un blocco di 0" con la condizione che il numero di 0 nel primo blocco sia uguale a quello del terzo blocco. Un tipico esempio: $0^3\, 1^5\, 0^3$.
Intuizione: riconoscere l'uguaglianza di due blocchi di 0 separati da un certo numero (qualunque) di 1 *non* è possibile con un automa a stati finiti, perché per memorizzare quanti 0 vanno poi "riconfrontati" successivamente servirebbe memoria illimitata. Ciò suggerisce che L_{mn} è **non regolare**.
Dimostrazione (Pumping Lemma):

1. Supponiamo, per assurdo, che L_{mn} sia regolare. Sia p il pumping length.
2. Consideriamo la stringa $s = 0^p\, 1^0\, 0^p = 0^p 0^p$ (due blocchi di 0 lunghi p). Tale stringa è in L_{mn}, ed ha lunghezza $2p \geq p$.
3. Dividiamo s in xyz con $|xy| \leq p$, $|y| > 0$. La parte y conterrà solo 0 dal primo blocco di zero (poiché $|xy| \leq p$).
4. Se "pompiano" verso il basso ($i = 0$) o verso l'alto ($i > 1$), la parte y aggiunta o rimossa sbilancia il conteggio dei 0 nella prima porzione rispetto alla terza. In particolare, riducendo y (cioè rimuovendola) si ottiene un blocco iniziale di 0 più corto di quello finale, uscendo così da L_{mn}. Contraddizione.

Concludiamo che L_{mn} non è regolare.

3) $L_{mnp} = \{\, 0^n\, 1^m\, 0^p \mid m + n + p > 0\}$

Soluzione

Qui la stringa ha la forma: un blocco di 0, poi un blocco di 1, poi un blocco di 0, *senza* alcuna richiesta di uguaglianza o relazioni particolari. L'unica condizione è che non sia la stringa vuota ($m + n + p > 0$ significa che almeno uno dei blocchi è di lunghezza positiva).
Verifica di regolarità: Si può costruire un NFA o un'espressione regolare. Ad esempio, consideriamo:

$$L_{mnp} = \{0^*1^*0^*\} \setminus \{\varepsilon\}.$$

Il linguaggio $0^*1^*0^*$ è sicuramente regolare (si tratta di un blocco di 0, poi un blocco di 1, poi un blocco di 0, inclusa la stringa vuota). $\{\varepsilon\}$ è anch'esso regolare, e i linguaggi regolari sono chiusi rispetto alla differenza. Quindi L_{mnp} è l'intersezione di $0^*1^*0^*$ con il compl. di $\{\varepsilon\}$, oppure semplicemente $0^*1^*0^*$ meno la vuota. Comunque, in qualunque modo lo si veda, **è regolare**.
Costruendo direttamente un automa, bastano 3 fasi: leggi 0 finché vuoi (stato 0), poi leggi 1 finché vuoi (stato 1), poi leggi 0 finché vuoi (stato 2). Per assicurarsi di non accettare la stringa vuota, imponiamo di passare almeno una volta a consumare qualche simbolo. In pratica, l'automa è banale e conferma la regolarità.

4) $L_{2ab} = \left\{ w \in \{a, b\}^* \mid \#a(w) = 2 \cdot \#b(w) \right\}$

Soluzione

Qui abbiamo la condizione che il numero di `a` nella stringa sia *esattamente* doppio del numero di `b`. Un automa a stati finiti non può tenere traccia di quest'uguaglianza (rapporti tra contatori illimitati) in modo finito.
Dimostrazione (Pumping Lemma):

1. Assumiamo che L_{2ab} sia regolare, con pumping length p.
2. Scegliamo una stringa s con $\#a(s) = 2p$ e $\#b(s) = p$ in cui tutti i `a` e `b` siano "gruppati" in modo da poter poi pompare. Per esempio:
$$s = a^p\, b^p\, a^p,$$
che ha in totale $2p + p = 3p$ `a` e `b`. In tale stringa, $\#a = 2p$ e $\#b = p$, quindi $s \in L_{2ab}$.
3. Dividiamo $s = xyz$ con $|xy| \leq p$, $|y| > 0$. Poiché $|xy| \leq p$, la parte y contiene `a` (o comunque si concentra nella sezione iniziale di `a`).
4. Se "pompiano" la sottostringa y (sia rimuovendola che duplicandola), la quantità di `a` cambia, mentre il numero di `b` rimane invariato. Dunque $\#a \neq 2\,\#b$ nella stringa pompata. Contraddizione.

Ne segue che L_{2ab} non è regolare.

Riassunto Finale:

- L_{3n}: **Regolare** (basta un DFA mod 3).
- L_{mn}: **Non regolare** (simile a $0^n 1^m 0^n$, si dimostra col Pumping Lemma).
- L_{mnp}: **Regolare** (forma $0^*1^*0^*$, eccetto la stringa vuota).
- L_{2ab}: **Non regolare** (il a = 2 b non è memorizzabile in modo finito).