

Automati e Linguaggi Formali

Parte 4 – Linguaggi non regolari

Davide Bresolin
Ultimo aggiornamento: 23 marzo 2023



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

1 Linguaggi non regolari

- costruite un FA che riconosce il linguaggio

$$L_{01} = \{0^n 1^n : n \geq 0\}$$

- rispondete alla domanda

il linguaggio L_{01} è regolare?

Dimostriamo che L_{01} non è regolare



- Supponiamo che $L_{01} = \{0^n 1^n : n \geq 0\}$ sia regolare

Dimostriamo che L_{01} non è regolare



- Supponiamo che $L_{01} = \{0^n 1^n : n \geq 0\}$ sia regolare
- Allora deve essere accettato da un DFA A con un certo numero k di stati

Dimostriamo che L_{01} non è regolare

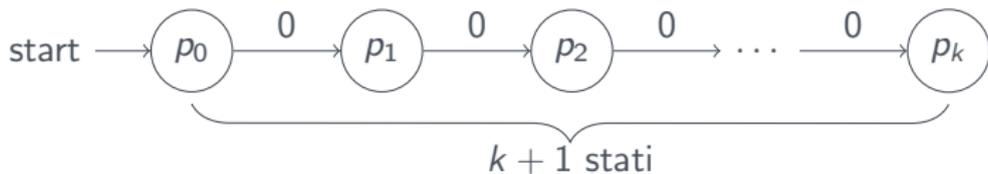


- Supponiamo che $L_{01} = \{0^n 1^n : n \geq 0\}$ sia regolare
- Allora deve essere accettato da un DFA A con un certo numero k di stati
- Cosa succede quando A legge 0^k ?

Dimostriamo che L_{01} non è regolare



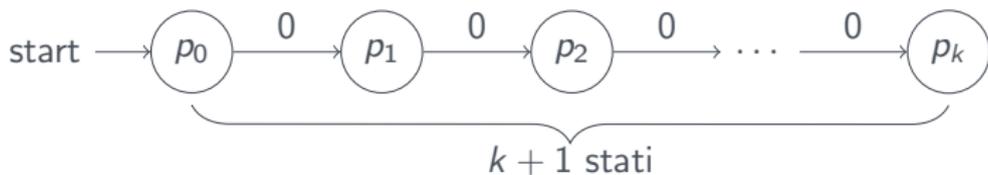
- Supponiamo che $L_{01} = \{0^n 1^n : n \geq 0\}$ sia regolare
- Allora deve essere accettato da un DFA A con un certo numero k di stati
- Cosa succede quando A legge 0^k ?
- Seguirà una qualche sequenza di transizioni:



Dimostriamo che L_{01} non è regolare



- Supponiamo che $L_{01} = \{0^n 1^n : n \geq 0\}$ sia regolare
- Allora deve essere accettato da un DFA A con un certo numero k di stati
- Cosa succede quando A legge 0^k ?
- Seguirà una qualche sequenza di transizioni:

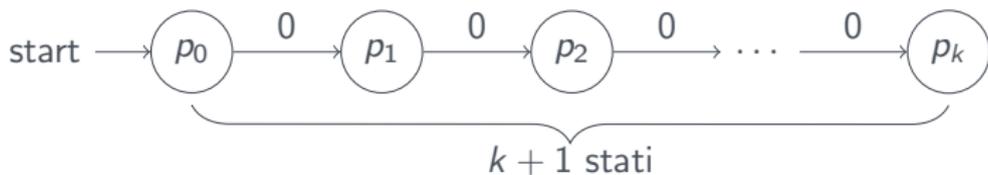


- Siccome ci sono $k + 1$ stati nella sequenza, **esiste uno stato che si ripete**: esistono $i < j$ tali che $p_i = p_j$

Dimostriamo che L_{01} non è regolare



- Supponiamo che $L_{01} = \{0^n 1^n : n \geq 0\}$ sia regolare
- Allora deve essere accettato da un DFA A con un certo numero k di stati
- Cosa succede quando A legge 0^k ?
- Seguirà una qualche sequenza di transizioni:



- Siccome ci sono $k + 1$ stati nella sequenza, **esiste uno stato che si ripete**: esistono $i < j$ tali che $p_i = p_j$
- Chiamiamo q questo stato

Dimostriamo che L_{01} non è regolare



- Cosa succede quando l'automa A legge 1^i partendo da q ?

Dimostriamo che L_{01} non è regolare



- Cosa succede quando l'automa A legge 1^i partendo da q ?
- Se l'automa finisce la lettura in uno stato finale:
 - allora accetta, sbagliando, la parola $0^j 1^i$

Dimostriamo che L_{01} non è regolare



- Cosa succede quando l'automa A legge 1^i **partendo da q** ?
- Se l'automa finisce la lettura in uno stato finale:
 - allora accetta, **sbagliando**, la parola 0^j1^i
- Se l'automa finisce la lettura in uno stato non finale:
 - allora rifiuta, **sbagliando**, la parola 0^j1^i

Dimostriamo che L_{01} non è regolare



- Cosa succede quando l'automa A legge 1^i partendo da q ?
- Se l'automa finisce la lettura in uno stato finale:
 - allora accetta, sbagliando, la parola 0^i1^i
- Se l'automa finisce la lettura in uno stato non finale:
 - allora rifiuta, sbagliando, la parola 0^i1^i
- In entrambi i casi abbiamo ingannato l'automa, quindi L_{01} non può essere regolare



Theorem (Pumping Lemma per Linguaggi Regolari)

Sia L un *linguaggio regolare*. Allora

- *esiste una lunghezza* $k > 0$ *tale che*
- *ogni parola* $w \in L$ *di lunghezza* $|w| \geq k$
- *può essere spezzata in* $w = xyz$ *tale che:*
 - 1 $y \neq \varepsilon$ *(il secondo pezzo è non vuoto)*
 - 2 $|xy| \leq k$ *(i primi due pezzi sono lunghi al max k)*
 - 3 $\forall i \geq 0, xy^i z \in L$ *(possiamo “pompare” y rimanendo in L)*



Dimostrazione:

- Supponiamo che L sia un linguaggio regolare

Dimostrazione:

- Supponiamo che L sia un linguaggio regolare
- Allora è riconosciuto da un DFA con, supponiamo, k stati

Dimostrazione:

- Supponiamo che L sia un linguaggio regolare
- Allora è riconosciuto da un DFA con, supponiamo, k stati
- Consideriamo una parola $w = a_1a_2 \dots a_n \in L$ di lunghezza $n \geq k$

Dimostrazione:

- Supponiamo che L sia un linguaggio regolare
- Allora è riconosciuto da un DFA con, supponiamo, k stati
- Consideriamo una parola $w = a_1a_2 \dots a_n \in L$ di lunghezza $n \geq k$
- Consideriamo gli stati nella computazione di A per w :

$$p_0 p_1 p_2 \dots p_k \dots p_n$$

Dimostrazione:

- Supponiamo che L sia un linguaggio regolare
- Allora è riconosciuto da un DFA con, supponiamo, k stati
- Consideriamo una parola $w = a_1 a_2 \dots a_n \in L$ di lunghezza $n \geq k$
- Consideriamo gli stati nella computazione di A per w :

$$p_0 p_1 p_2 \dots p_k \dots p_n$$

- Siccome in p_0, p_1, \dots, p_k ci sono $k + 1$ stati, ne esiste uno che si ripete:

esistono $l < m$ tali che $p_l = p_m$ e $m \leq k$

- Possiamo spezzare w in tre parti $w = xyz$:

1 $x = a_1 a_2 \dots a_l$

2 $y = a_{l+1} a_{l+2} \dots a_m$

3 $z = a_{m+1} a_{m+2} \dots a_n$

- Possiamo spezzare w in tre parti $w = xyz$:

- 1 $x = a_1 a_2 \dots a_l$

- 2 $y = a_{l+1} a_{l+2} \dots a_m$

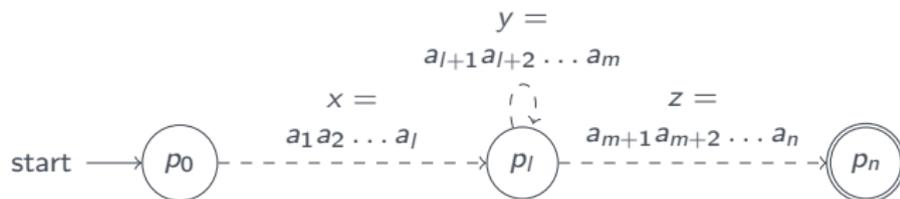
- 3 $z = a_{m+1} a_{m+2} \dots a_n$

- che rispettano le condizioni del Lemma:

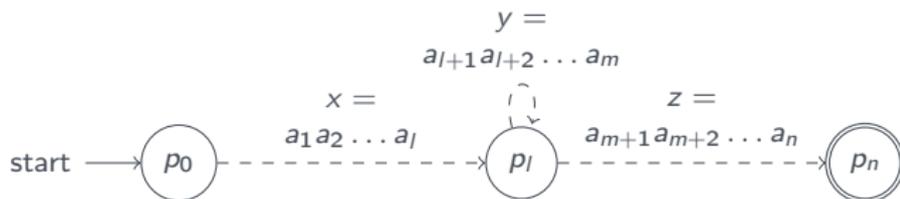
- $y \neq \varepsilon$ perché $l < m$

- $|xy| \leq k$ perché $m \leq k$

- Quindi, nel grafo delle transizioni di A :



- Quindi, nel grafo delle transizioni di A :



- E di conseguenza anche $xy^i z$ viene riconosciuta dall'automa per ogni $i \geq 0$ □

Il Pumping Lemma come Gioco



www.geocities.com/creatorgang



- L'avversario sceglie la lunghezza k
- Noi scegliamo una parola w
- L'avversario spezza w in xyz
- Noi scegliamo i tale che $xy^iz \notin L$
- allora **abbiamo vinto**

In aula: giocate con il vostro vicino di banco:

- Scegliete chi muove per primo e chi per secondo
- Il primo giocatore ha l'obiettivo di mostrare che il linguaggio rispetta il Pumping Lemma
- Il secondo giocatore che non lo rispetta
- Giocate riempiendo gli spazi nel testo sottostante
- Girate il foglio e giocate la seconda partita scambiandovi i ruoli (chi ha mosso per primo ora muove per secondo)

Online:

- Esercizi nel moodle: [4 - Esercizi: gioco del Pumping Lemma](#)

Se il linguaggio rispetta il Pumping Lemma:

- il Giocatore 1 ha una strategia vincente
- per qualsiasi mossa faccia il Giocatore 2, può rispondere in modo da vincere il gioco.

Se il linguaggio non rispetta il Pumping Lemma:

- il Giocatore 2 ha una strategia vincente
- per qualsiasi mossa faccia il Giocatore 1, può rispondere in modo da vincere il gioco.

- Ogni linguaggio regolare soddisfa il Pumping Lemma.
- Un linguaggio che **falsifica** il Pumping Lemma non può essere regolare:
 - per ogni lunghezza $k \geq 0$
 - esiste una parola $w \in L$ di lunghezza $|w| \geq k$ tale che
 - per ogni suddivisione $w = xyz$ tale che:
 - 1 $y \neq \varepsilon$ (il secondo pezzo è non vuoto)
 - 2 $|xy| \leq k$ (i primi due pezzi sono lunghi al max k)
 - esiste un $i \geq 0$ tale che $xy^iz \notin L$ (possiamo “pompare” y ed uscire da L)
- **Attenzione:** esistono linguaggi non regolari che rispettano il Pumping Lemma!

- 1 Sia L_{ab} il linguaggio delle stringhe sull'alfabeto $\{a, b\}$ dove il numero di a è uguale al numero di b . L_{ab} è regolare?

1 Sia L_{ab} il linguaggio delle stringhe sull'alfabeto $\{a, b\}$ dove il numero di a è uguale al numero di b . L_{ab} è regolare?

No, L_{ab} non è regolare:

- supponiamo per assurdo che lo sia
- sia k la lunghezza data dal Pumping Lemma
- consideriamo la parola $w = a^k b^k$
- sia $w = xyz$ una suddivisione di w tale che $y \neq \varepsilon$ e $|xy| \leq k$:
$$w = \underbrace{aaa \dots a}_x \underbrace{a \dots a}_y \underbrace{a \dots ab \dots bb}_z$$
- poiché $|xy| \leq k$, le stringhe x e y sono fatte solo di a
- per il Pumping lemma, anche $xy^2z \in L_{ab}$, ma contiene **più a che b** \Rightarrow assurdo

2 Il linguaggio $L_{rev} = \{ww^R : w \in \{a, b\}^*\}$ è regolare?

2 Il linguaggio $L_{rev} = \{ww^R : w \in \{a, b\}^*\}$ è regolare?

No, L_{rev} non è regolare:

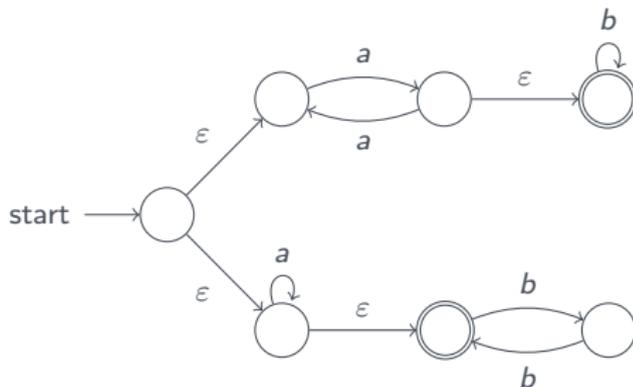
- supponiamo per assurdo che lo sia
- sia k la lunghezza data dal Pumping Lemma
- consideriamo la parola $w = a^k b b a^k$
- sia $w = xyz$ una suddivisione di w tale che $y \neq \varepsilon$ e $|xy| \leq k$:
$$w = \underbrace{aaa \dots a}_x \underbrace{a}_{y} \underbrace{abb aaa \dots aaa}_z$$
- poiché $|xy| \leq k$, le stringhe x e y sono fatte solo di a
- per il Pumping lemma, anche $xy^0z = xz \in L_{rev}$, ma non la posso spezzare in $ww^R \Rightarrow$ **assurdo**

3 Il linguaggio $L_{nm} = \{a^n b^m : n \text{ è dispari oppure } m \text{ è pari}\}$ è regolare?

3 Il linguaggio $L_{nm} = \{a^n b^m : n \text{ è dispari oppure } m \text{ è pari}\}$ è regolare?

Si, L_{nm} è regolare:

- è rappresentato dall'espressione regolare $a(aa)^*b^* + a^*(bb)^*$
- e riconosciuto dall'automa



4 Il linguaggio $L_p = \{1^p : p \text{ è primo}\}$ è regolare?

4 Il linguaggio $L_p = \{1^p : p \text{ è primo}\}$ è regolare?

No, L_p non è regolare:

- supponiamo per assurdo che lo sia
- sia k la lunghezza data dal Pumping Lemma
- consideriamo una parola $w = 1^p$ con p primo e $p > k + 2$
- sia $w = xyz$ una suddivisione di w tale che $y \neq \varepsilon$ e $|xy| \leq k$:

$$w = \underbrace{11 \dots 11}_x \underbrace{11 \dots 1}_y \underbrace{1111 \dots 11}_z$$

■ ...

- ...
- sia $|y| = m$: allora $|xz| = p - m$
- per il Pumping lemma, anche $v = xy^{p-m}z \in L_p$
- allora $|v| = m(p - m) + p - m = (p - m)(m + 1)$ si può scomporre in due fattori
- poiché $y \neq \varepsilon$, allora $|y| = m > 0$ e $m + 1 > 1$
- anche $p - m > 1$ perché abbiamo scelto $p > k + 2$ e $m \leq k$ perché $|xy| \leq k$
- i due fattori sono entrambi maggiori di 1 e quindi $|v|$ non è un numero primo
- $v \notin L_{rev}$, **assurdo**

5 Il linguaggio $L_{3n} = \{1^{3n+2} : n \geq 0\}$ è regolare?

6 Il linguaggio $L_{mn} = \{0^n 1^m 0^n : m + n > 0\}$ è regolare?

7 Il linguaggio $L_{mnp} = \{0^n 1^m 0^p : m + n + p > 0\}$ è regolare?

8 Il linguaggio

$L_{2ab} = \{w \in \{a, b\}^* : \text{numero di } a \text{ è due volte il numero di } b\}$

è regolare?

Considera il linguaggio

$$L = \{a^\ell b^m c^n \mid \ell, m, n \geq 0 \text{ e se } \ell = 1 \text{ allora } m = n\}.$$

- 1** Mostra che L non è regolare.
- 2** Mostra che L si comporta come un linguaggio regolare rispetto al Pumping Lemma. In altre parole, dai una lunghezza del pumping k e dimostra che L soddisfa le condizioni del Pumping Lemma per questo valore di k .
- 3** Spiega perché i punti (1) e (2) non contraddicono il Pumping Lemma.