

Automati e Linguaggi Formali

Proprieta' dei linguaggi regolari – Il Pumping lemma –

A.A. 2014-2015
Enrico Mezzetti
emezzett@math.unipd.it



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Proprieta' dei linguaggi regolari

- **Pumping Lemma**
 - Ogni linguaggio regolare soddisfa il pumping lemma
 - Se L non e' regolare il pumping lemma mostrera' una contraddizione
- **Proprieta' di chiusura**
 - Costruire automi da componenti usando delle operazioni algebriche sui linguaggi
 - E.g., dati L e M possiamo costruire un automa per $L \cap M$
- **Proprieta' di decisione**
 - Analisi computazionale degli automi
 - Quanto costa controllare proprieta sugli automi
- **Tecniche di minimizzazione**
 - Risparmiare costruendo automi piu piccoli
 - Processo di semplificazione



Automati e Linguaggi Formali – A.A. 2014-2015
Docente: Enrico Mezzetti

2 of 10

Pumping lemma - Informale

- Supponiamo che $L_{01} = \{0^n 1^n : n \geq 1\}$ sia **regolare**
 - 01, 0011, 000111, ...
- Se L_{01} e' regolare \Rightarrow deve essere accettato da un DFA \mathcal{A}
 - $Q_{\mathcal{A}}$ e' un insieme *finito* di stati k
- Supponiamo che \mathcal{A} legga 0^k attraverso le transizioni
$$\begin{array}{ccccccc} p_0 & \xrightarrow{0} & p_1 & \xrightarrow{0} & p_2 & \cdots & \xrightarrow{0} & p_k \\ \epsilon & & 0 & & 00 & & & 0^k \end{array}$$
 - Utilizzando $k + 1$ stati
 - *Pigeonhole principle* $\Rightarrow \exists i < j : p_i = p_j$
- Quindi $\exists q$ che accomoda due sottosequenze $0^i \neq 0^j$ per $i < j$



Automati e Linguaggi Formali – A.A. 2014-2015
Docente: Enrico Mezzetti

3 of 10

Pumping lemma - Informale - 2

- Quindi $\exists q$ che accomoda due sottosequenze $0^i \neq 0^j$ per $i < j$
- Supponiamo di essere in q
 - Abbiamo letto i o j occorrenze di 0
$$\begin{array}{ccccccc} p_0 & \xrightarrow{0} & p_1 & \xrightarrow{0} & p_2 & \cdots & \xrightarrow{0} & p_i \\ \epsilon & & 0 & & 00 & & & 0^i \vee 0^j \end{array}$$
- \mathcal{A} puo' essere idotto a prendere una decisione errata:
 - Se $\hat{\delta}(q, 1^i) \in F$ l'automata accetta $0^i 1^i \rightarrow$ Errore!
 - Se $\hat{\delta}(q, 1^i) \notin F$ l'automata non accetta $0^i 1^i \rightarrow$ Errore!
- L_{01} non puo' essere rappresnetato da un DFA \mathcal{A}
 $\Rightarrow L_{01}$ **non e' regolare**



Automati e Linguaggi Formali – A.A. 2014-2015
Docente: Enrico Mezzetti

4 of 10

Pumping lemma per LR

Th. 4.1 Pumping lemma

Sia L un linguaggio regolare allora $\exists n$ costante (per L) tale che $\forall w \in L$ per cui $|w| \geq n$ possiamo scomporre in tre sotto-stringhe $w = xyz$ tale che:

- 1 $y \neq \epsilon$
- 2 $|xy| \leq n$
- 3 $\forall k \geq 0, xy^kz \in L$

👁️ Possiamo trovare una sotto-stringa non vuota (non troppo lontano dall'inizio) che puo' essere replicata un numero indefnito di volte senza uscire da L



Pumping lemma per LR (Prova)

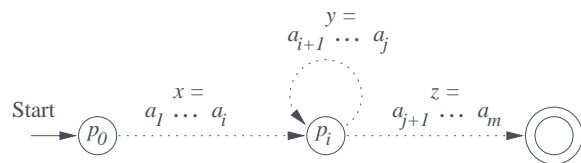
- Supponiamo che L sia regolare
 - $L = L(\mathcal{A})$: L e' riconosciuto da un DFA \mathcal{A}
 - \mathcal{A} ha un numero finito di n stati
- Sia $w = a_1a_2 \dots a_m \in L$ $m \geq n$ e $a_i \in \Sigma_{\mathcal{A}}$
- Sia $p_i = \hat{\delta}(q_0, a_1a_2 \dots a_i)$
 - p_i per $i \in \{0, 1, \dots, n\}$ non possono essere distinti
 - Pigeonhole principle $\Rightarrow \exists i < j : p_i = p_j$



Pumping lemma per LR (Prova) - 2

- Scomponiamo $w = xyz$ come

- 1 $x = a_1a_2 \dots a_i$
- 2 $y = a_{i+1}a_{i+2} \dots a_j$
- 3 $z = a_{j+1}a_{j+2} \dots a_m$



- \mathcal{A} riceve in input xy^kz per $k \geq 0$
 - $K=0 \rightarrow \mathcal{A}$ accetta xz poiche' $p_i = p_j$
 - $K>0 \rightarrow \mathcal{A}$ accetta xz poiche' cicla su p_i k volte
- Quindi anche $xy^kz \in L$, per ogni $k \geq 0$

👁️ x, y potenzialmente ϵ per $i=0$ e $j=n$ rispettivamente



Pumping lemma per LR - Esempio

- Sia L_{eq} il linguaggio delle stringhe con ugual numero di 0 e 1
 - Nessun vincolo su ordine
- Supponiamo che L_{eq} sia regolare
 - Scegliamo $w = 0^n1^n \in L_{eq}$ ($|w| \geq n$)
- Per il pumping lemma $w = xyz$ e:
 - $|xy| \leq n$
 - $y \neq \epsilon$
 - $xy^kz \in L_{eq}$

- Quindi xy e' composta di soli 0

$$w = \underbrace{000 \dots 000}_x \underbrace{0111 \dots 11}_y \underbrace{\dots 11}_z$$

- Scegliamo $k = 0$
- Per PL: se L_{eq} regolare allora anche $xz \in L_{eq}$
 - xz contiene almeno uno 0 in meno del dovuto! (mancano quelli di y)
- $\Rightarrow L_{eq}$ **non e' regolare**



Pumping lemma per LR - Esempio 2

- Sia L_{pr} il linguaggio delle stringhe composte da un numero p di 1, dove p e' primo
 - $L_{pr} = \{1^p : p \text{ e' primo}\}$
- Supponiamo sia regolare:
 - Esiste n che soddisfa le condizioni del PL
 - Scegliamo $w = 1^p$ dove $p \geq n + 2$

$$w = \underbrace{111 \dots 1}_x \underbrace{\dots 1}_y \underbrace{1111 \dots 11}_z$$

$|y|=m$

- Supponiamo $|y| = m \neq \epsilon \rightarrow |xz| = p - m$
- Ora scegliamo $k = p - m$
- Se L_{pr} e' regolare allora anche $xy^{p-m}z \in L_{pr}$



Pumping lemma per LR - Esempio 2

- Sia L_{pr} il linguaggio delle stringhe composte da un numero p di 1, dove p e' primo

$$w = \underbrace{111 \dots 1}_x \underbrace{\dots 1}_y \underbrace{1111 \dots 11}_z$$

$|y|=m$

- Abbiamo scelto $k = p - m$

- $xy^{p-m}z \in L_{pr}?$
- $|xy^{p-m}z| = |xz| + (p - m)|y|$
 $= p - m + (p - m)m$
 $= (1 + m)(p - m)$

(che non e' primo a meno che uno dei due fattori non sia 1)

a) $y \neq \epsilon \Rightarrow 1 + m > 1$

b) $m = |y| \leq |xy| \leq n, p \geq n + 2 \Rightarrow p - m \geq n + 2 - n = 2$

$\Rightarrow L_{pr}$ non e' regolare

