Logica e Modelli Computazionali

Limiti dei Linguaggi Regolari

Marco Console

Ingegneria Informatica e Automatica (Sapienza, Università di Roma)

Esercizio?

- Abbiamo definito un modello computazionale molto potente
  - Cosa possiamo farci in pratica?
- **Idea 1**: Usiamoli per riconoscere formule della logica proposizionale ben formate :D
  - Non è un problema banale, ci sono vari simboli da riconoscere ☹
- **Idea 2**: Iniziamo da un sottoproblema. Riconosciamo stringhe di parentesi ben formate
  - È necessario per riconoscere le formule della logica proposizionale

- **Definizione**. Una stringa s sull'alfabeto $\Sigma = \{(,)\}$ è una stringa di parentesi ben formata se
  - $s = (\,)$ oppure
  - $s = (\,p\,)$ e p è una stringa ben formata

- **Definizione**. Il linguaggio $\mathcal{P}$ è l'insieme di tutte le stringhe di parentesi ben formate

- **Esercizio (?)**. Definire un ϵ-ASFND A tale che $\mathcal{P} = L(A)$
  - Utilizzando la chiusura sotto concatenazione e star dei linguaggi regolari, A ci fornisce una buona parte dell'Idea 1!

Linguaggi Regolari

- **Definizione**. Un **linguaggio $\mathcal{L}$ è detto regolare** se esiste un **ASFD A tale che $L(A) = \mathcal{L}$**
  – Un linguaggio è regolare se e solo se esiste un ASFD che lo riconosce
  – Equivalentemente, possiamo richiedere l'esistenza di un ASFND o di un ϵ-ASFND

- **Domanda 1.** Sia $\mathcal{L}$ un qualunque linguaggio su un alfabeto Σ. È vero che esiste un ϵ-ASFND che riconosce $\mathcal{L}$ ovvero $\mathcal{L} = L(A)$?
  – Equivalentemente, è vero che per ogni linguaggio esiste un automa che lo riconosce?

- **Domanda 2.** Come possiamo fornire una prova formale che dimostri la nostra risposta?
  – Se la risposta è sì, dobbiamo costruire un automa che riconosce il linguaggio
  – Se la risposta è no, dobbiamo mostrare che nessun automa riconosce il linguaggo
    • Tutti gli automi non lo riconoscono!!

Linguaggi NON Regolari

- **Definizione**. Un **linguaggio $\mathcal{L}$ è detto regolare** se esiste un **ASFD A tale che $L(A) = \mathcal{L}$**
  - Un linguaggio è regolare se e solo se esiste un ASFD che lo riconosce
  - Equivalentemente, possiamo richiedere l'esistenza di un ASFND o di un ϵ-ASFND

- **Domanda 1**. Sia $\mathcal{L}$ un linguaggio su un alfabeto Σ. È vero che esiste un ϵ-ASFND che riconosce $\mathcal{L}$ ovvero $\mathcal{L} = L(A)$?

- **Teorema 1**. Esiste almeno un linguaggio $\mathcal{L}$ che non è regolare

- **Domanda 2**. Come possiamo fornire una prova formale che dimostri la nostra risposta?

- **Lemma 1**. Esiste un linguaggio $\mathcal{L}$ per cui per ogni ASFD A abbiamo $L(A) \neq \mathcal{L}$
  - Per dimostrare Lemma 1 dobbiamo entrare **ancora più in profondità** nelle proprietà degli ASFD

Linguaggi Finiti

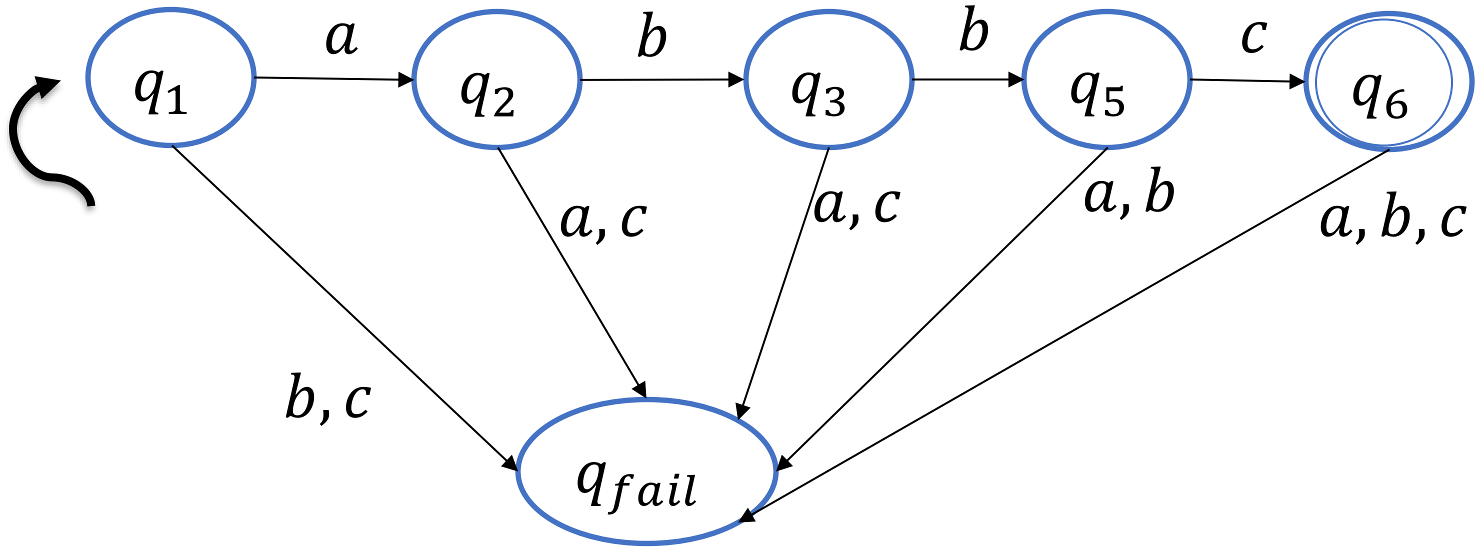
- **Definizione**. Un **linguaggio $\mathcal{L}$ è detto regolare** se esiste un **ASFD A tale che $L(A) = \mathcal{L}$**
  - Un linguaggio è regolare se e solo se esiste un ASFD che lo riconosce
  - Equivalentemente, possiamo richiedere l'esistenza di un ASFND o di un ϵ-ASFND

- Nella nostra ricerca di un linguaggio non regolare, possiamo scartare linguaggi finiti infatti

- **Lemma 2**. Sia $\mathcal{L}$ un linguaggio tale che $|\mathcal{L}| = n$ con $n \in \mathbb{N}$. Il linguaggio $\mathcal{L}$ è regolare
- **Prova**. Data una stringa $s = c_1 c_2 \dots c_k$, costruiamo un ASFD $A = \langle \Sigma, Q, I, F, \ \delta \rangle$ tale che $L(A) = \{s\}$
  - $\Sigma = \{c_1, c_2, \dots, c_k\}$; $Q \ = \{q_1, q_2, \dots, q_k, q_{fin}, q_{fail}\}$; $I = q_1$;
  - $\delta(q_i, c_i) = q_{i+1}$ per ogni $i = 1, \dots, k-1$
  - $\delta(q_i, c_j) = q_{fail}$ per ogni $j = 1, \dots, k-1$ e $j \neq i$
  - $\delta(q_k, c_k) = q_{fin}$
  - $\delta(q_{fin}, c) = q_{fail}$ per ogni $c \in \Sigma$
- Il lemma è ora è una semplice conseguenza del fatto che se $\mathcal{L}_1$ e $\mathcal{L}_2$ sono linguaggi regolari allora anche $\mathcal{L}_1 \cup \mathcal{L}_2$ è un linguaggio regolare

Linguaggi Finiti – Esempio

- L'automa definito dal seguente linguaggio riconosce la stringa $abbc$ sull'alfabeto $\{a, b, c\}$



Il Pumping Lemma

- **Intuizione**: ogni stringa **sufficientemente lunga** di un linguaggio regolare di **cardinalità infinita** esibisce una **struttura ricorrente**, ovvero **contiene una sottostringa che può essere ripetuta a volontà (pumped)** ottenendo altre stringhe dello stesso linguaggio

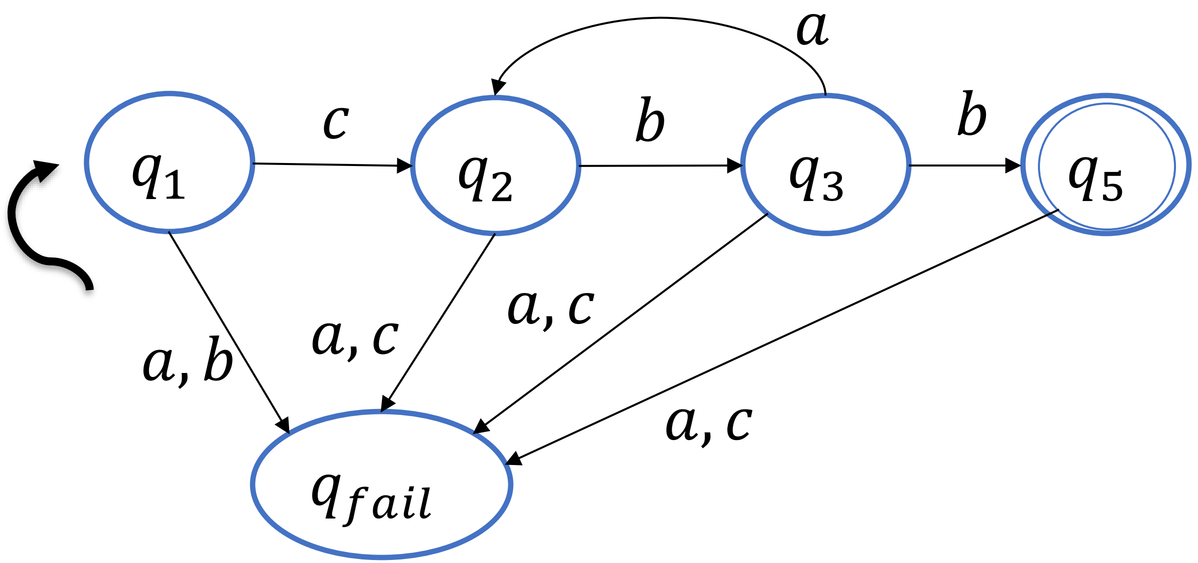
- **Lemma [Pumping Lemma]:** Per ogni **linguaggio regolare $\mathcal{L}$** di **cardinalità infinita**, esiste una **numero intero n** tale che, per ogni $z \in L$ con $|z| \geq n$, esistono tre stringhe u, v e w (non necessariamente in $\mathcal{L}$) tali che le seguenti sono verificate:
  - $z = uvw$
  - $|u| \geq 0, |v| \geq 1, |w| \geq 0$, e $|uv| \leq n$,
  - $uv^iw \in L$ per ogni $i \geq 0$

Il Pumping Lemma – Intuizione

- **Intuizione 1**: Se $\mathcal{L}$ è regolare e di cardinalità infinita, esiste una costante $n \in \mathbb{N}$ tale che, per ogni $z \in L$ con $|z| \geq n$, esiste un prefisso uv di z con $|v| \geq 1$ e $|uv| \leq n$ tale che una sottostringa v del prefisso può essere ripetuta all'infinito

- **Intuizione 2**: Se $\mathcal{L}$ è regolare e di cardinalità infinita, esiste un ciclo nel grafo indotto dall'automa che riconosce $\mathcal{L}$. Se tale ciclo non esiste, il linguaggio definito è finito

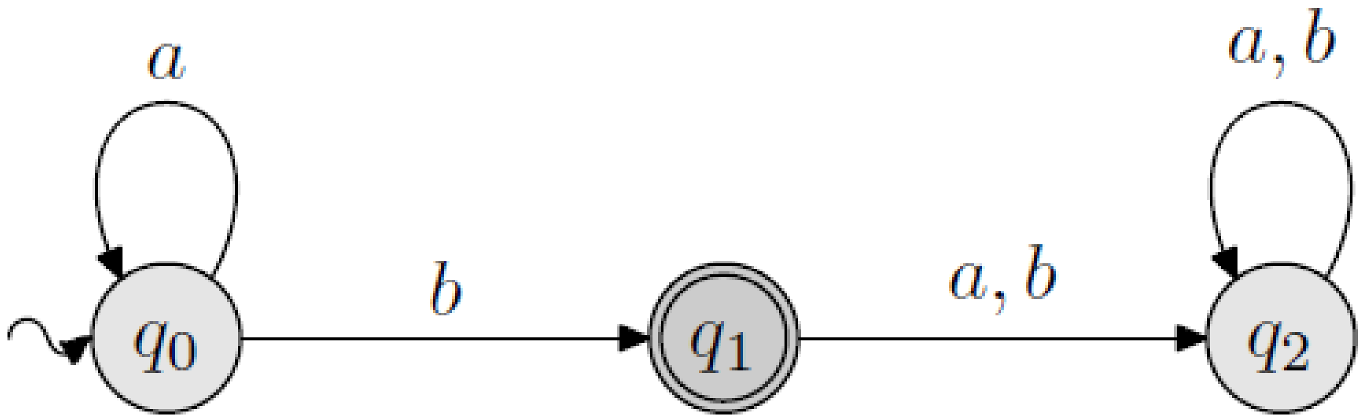
Pumping Lemma – Esempio



$L(A) = \{cb(ab)^n b \mid per\ n \geq 0\}$

- In questo caso, $n = 2$, ovvero il prefisso ricorrente inizia dal secondo simbolo
- **Stringa**: $s = cbabab$,
- $u = c, v = baba, w = b$,
- Ogni stringa $c(baba)^i b$ è in $\mathcal{L}$

Pumping Lemma – Esempio



$L(A) = \{a^n b \mid per\ n \geq 0\}$

- In questo caso, $n = 1$ ovvero il prefisso ricorrente inizia dal primo simbolo
- **Stringa**: $s = aaab$,
- $u = \epsilon, v = a, w = aab$,
- Ogni stringa $a^i aaab$ è in $\mathcal{L}$

Dimostrazione del Pumping Lemma (1/5)

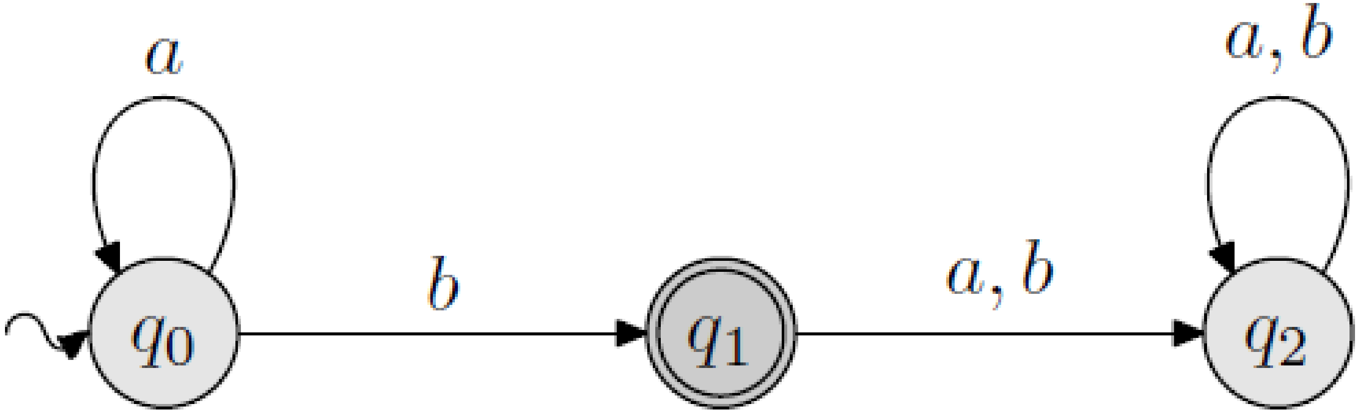
- **Lemma [Pumping Lemma]:** Per ogni **linguaggio regolare $\mathcal{L}$** di **cardinalità infinita**, esiste una **numero intero n** tale che, per ogni se $z \in L$ con $|z| \geq n$, esistono tre stringhe u, v e w (non necessariamente in $\mathcal{L}$) tali che le seguenti sono verificate:
  - $z \ = uvw$
  - $|u| \geq 0, |v| \geq 1, |w| \geq 0,$ e $|uv| \leq n,$
  - $uv^i w \in L$ per ogni $i \geq 0$

- **Dimostrazione**. Sia $A =< \Sigma, Q, \delta, q_0, F >$ **l'ASFD che riconosce $\mathcal{L}$**, ovvero $\mathcal{L} = L(A)$. Procediamo a dimostrare che **il valore n desiderato è uguale a $|Q| + 1$** (cardinalità del numero di stati di A).
  - **Sia $n = |Q| + 1$**
  - Sia $z \in \mathcal{L}$ una stringa di cardinalità maggiore di $|z| \geq n$.
  - Tale $z \in \mathcal{L}$ esiste inquanto $|\mathcal{L}| = \infty$ e $|\Sigma| < \infty$. Quindi, per ogni $k \in \mathbb{N}$ esiste $s \in \mathcal{L}$ tale che $|s| > k$
  - Per definizione, la computazione $A(z) = q_z^0, q_z^1, \dots, q_z^{|z|}$ è **accettante** e quindi dove $q_z^{|z|} \in F$.
  - …

Dimostrazione del Pumping Lemma (2/5)

- Per definizione, la computazione $A(z) = q_z^0, q_z^1, \dots, q_z^{|z|}$ è **accettante** e quindi dove $q_z^{|z|} \in F$.
- Assumiamo che la stringa z sia composta dai caratteri $z = c_1 c_2 \dots c_{|z|}$.
- Dato che $|z| \geq n = |Q| + 1$, esiste almeno uno stato q nella sequenza $A(z)$ che viene ripetuto **due o più volte**
  - **Principio della piccionaia.** Per ogni funzione $f\colon \mathrm{A} \rightarrow B$ con $|A| > |B|$ esistono $a_1, a_2 \in A$ tali che $f(a_1) = f(a_2)$
- Sia ***j* il valore minimo** per cui lo stato q_z^j viene **ripetuto due o più volte** nella sequenza $A(z)$
  - q_z^j è il primo stato di $A(z)$ ad essere ad essere ripetuto
  - q_z^j compare per la prima volta in posizione j in $A(z)$

Dimostrazione del Pumping Lemma (2/5) – Esempio



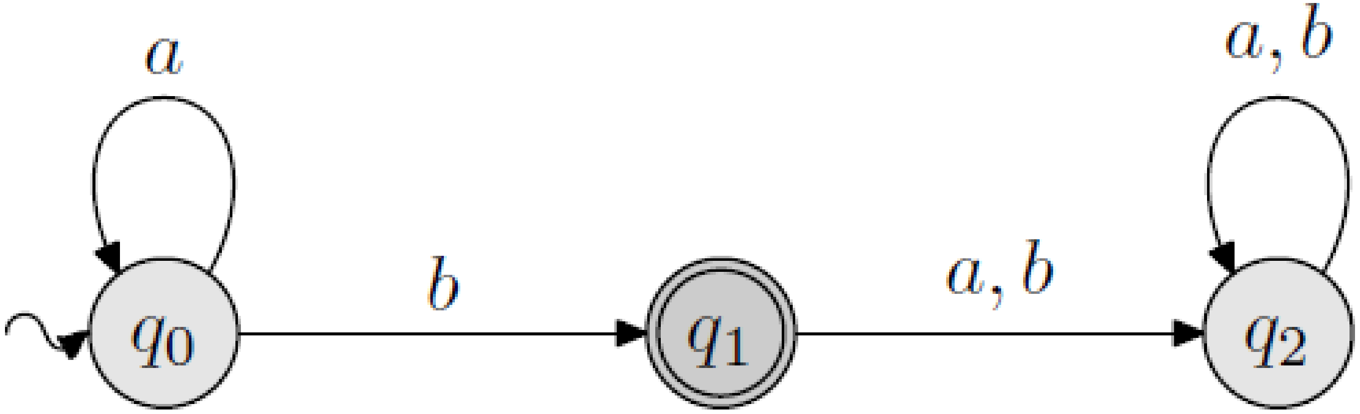
- Supponiamo che $z = aaaab$ e quindi $|z| = 5 > 3 = |Q|$
- La computazione associata è $\mathrm{e} = (q_0, q_0, q_0, q_0, q_0, q_1)$
- Il valore j più piccolo per cui q_z^j in e si ripete è 1 e $q_z^j = q_0$

Dimostrazione del Pumping Lemma (3/5)

- Sia $\boldsymbol{u}$ **la sottostringa** $\boldsymbol{c_1 c_2 \dots c_{j-1}}$ **di** $\boldsymbol{z}$ (se $j-1=0$ allora $u=\epsilon$)
  - I primi $j-1$ simboli di z, cioè il prefisso che precede l'inizio della ripetizione in q_z^j
  - Ovviamente $|u|<n$ essendo q_z^j il primo stato di $A(z)$ che viene ripetuto
    - La sequenze $(q_z^1 \dots q_z^j)$ **non contiene ripetizioni**
    - Possono esserci al più $|Q|$ stati nella sequenza e $n=|Q|+1>|Q|$

- Sia $\boldsymbol{l>j}$ **il valore minimo tale che** $\boldsymbol{q_z^l=q_z^j}$ in $A(z)$
  - q_z^j appare in $A(x)$ per la **prima volta in posizione** $\boldsymbol{j}$ e **per la seconda volta in posizione** $\boldsymbol{l}$
  - Nessun altro stato è ripetuto fra q_z^1 a q_z^l essendo q_z^j il primo stato ad essere ripetuto

Dimostrazione del Pumping Lemma (3/5) – Esempio



- Supponiamo che $z = aaaab$ e quindi $|z| = 5 > 3 = |Q|$
- La computazione associata è $\mathrm{e} = (q_0, q_0, q_0, q_0, q_0, q_1)$
- Il valore j più piccolo per cui q_z^j in e si ripete è 1 e $q_z^j = q_0$
- $u = \epsilon$(stringa vuota) e il più piccolo valore $l > j$ per $q_z^j = q_0$ si ripete è $l = 2$

Dimostrazione del Pumping Lemma (4/5)

- Sia v **la sottostringa** $c_j c_{j+1} \dots c_{l-1}$ **di** z e $z = uvw$
  - v è la sottostringa di z dal carattere j-esimo a quello $(l-1)$-esimo
  - Tale sottostringa è quella riconosciuta utilizzando la sequenza di stati che parte da q_z^j **e arriva a** $q_z^l = q_z^j$

- **Osserviamo** che $|uv| \leq n$
  - Nella sequenza $(q_z^1 \dots q_z^l)$ c'è esattamente una ripetizione e $\mathrm{n} = |Q| + 1$

- **Definiamo** $A(q, s)$ con $q \in Q$ e $s \in \Sigma^*$ la computazione dell'automa $A' = \langle \Sigma, Q, q, F, \delta \rangle$ su s
  - La computazione **dell'automa A ma con stato iniziale q invece di q_0**
  - Chiaramente possiamo comporre le esecuzioni spostando lo stato iniziale
    - $A(q_0, ss') = A(ss')$ e $A(q_0, s) \circ A(q', s') = A(ss')$ se q' è lo stato finale di $A(s) = A(q_0, s)$

Dimostrazione del Pumping Lemma (5/5)

- **Osserviamo** ora quanto segue
  - $A(q_0, u) = (q_z^1, \dots, q_z^j)$
  - $A(q_z^j, v) = (q_z^j, \dots, q_z^l)$
  - $A(q_z^l, w) = (q_z^l, \dots, q_z^{|z|})$ con $q_z^{|z|}$ accettante

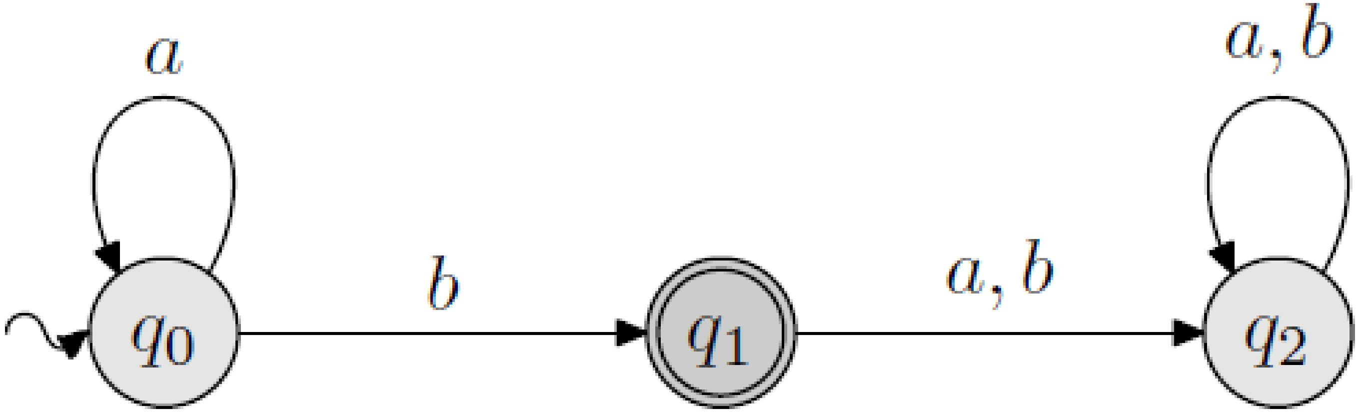
- Essendo $q_z^j = q_z^l = q_{loop}$ per costruzione, per ogni $m > 0$, la seguente identità è verificata

$$A(q_z^j, v^m) = \underbrace{(q_{loop}, \dots, q_{loop}, \dots, q_{loop}, \dots, q_{loop})}_{m\ volte}$$

- **Possiamo concludere che ripetendo la sottostringa v otteniamo ancora stringhe nel linguaggio**
  1. $A(uv^m w) = A(q_0, u) \circ A(q_z^j, v^m) \circ A(q_z^l, w)$
  2. $A(uv^m w)$ è una esecuzione accettante essendo il suo stato finale $q_z^{|z|}$

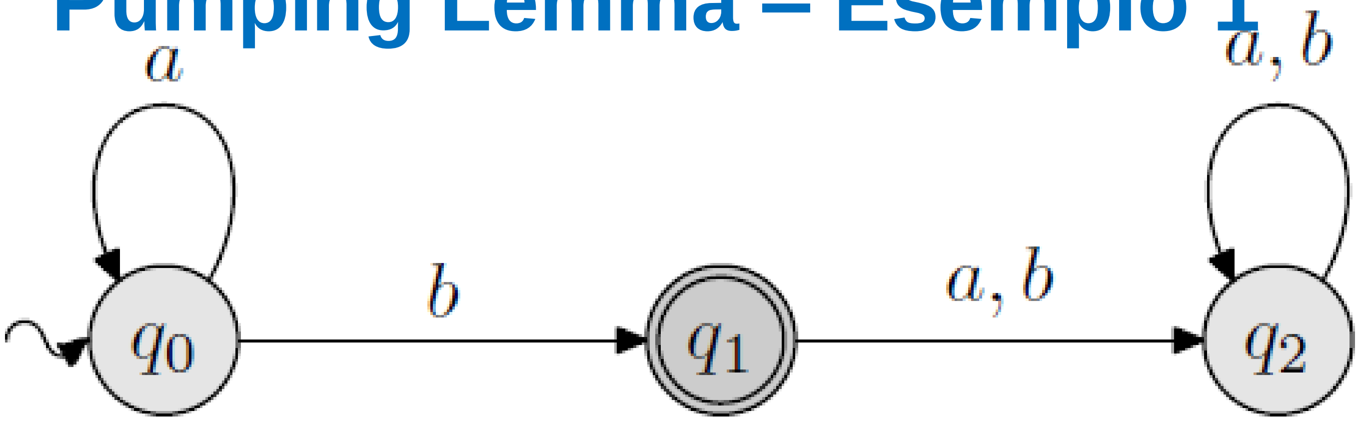
Il Lemma desiderato è dunque dimostrato

Dimostrazione del Pumping Lemma (3/5) – Esempio



- Supponiamo che $z = aaaab$ e quindi $|z| = 5 > 3 = |Q|$
- La computazione associata è $\mathrm{e} = (q_0, q_0, q_0, q_0, q_1)$
- Il valore j più piccolo per cui q_i^j in e si ripete è 1 e $q_i^j = q_0$
- $u = \epsilon$ (stringa vuota) e il più piccolo valore $l > j$ per $q_i^j = q_0$ si ripete nella sequenze è $l = 2$
- $A(q_0, \epsilon) = (q_0)$,
- $A(q_0, a^m) = (q_0, q_0, \dots, q_0)$
- $A(q_0, aaab) = (q_0, q_0, q_0, q_0, q_1)$

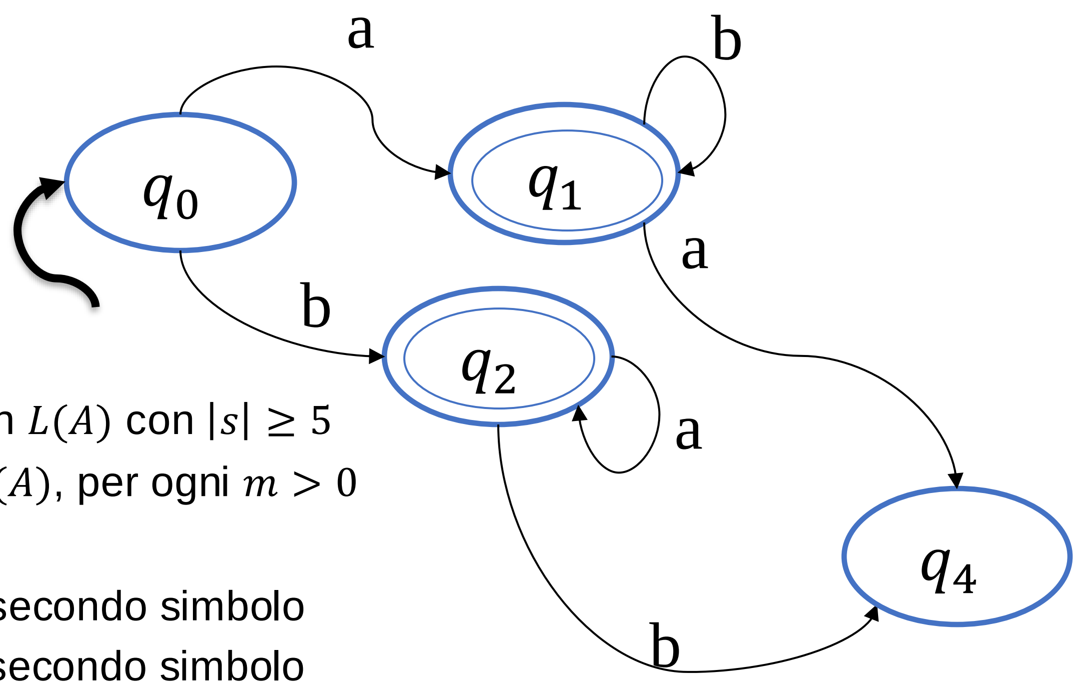
Pumping Lemma – Esempio 1



- Sia $\mathcal{L}$ il linguaggio definito dal seguente ASFD A in figura
- Sia $n = 4$ e s una qualunque stringa in $L(A)$ con $|s| \geq 4$
- **Pumping Lemma**. $s = uvw$ tale che $uv^m w \in L(A)$, per ogni $m > 0$
- In particolare possiamo assumere (**differentemente dal caso precedente**)
  – $w = \mathrm{b}$; $u = \epsilon$; $v =$ sottostringa di s composta da sole a
- **Esempio**. $s = aaaaab$
  – $w = \mathrm{b}$; $u = \epsilon$; $v = aaaaa$
  – $uv^m w = (aaaaa)^m w \in L(A)$ per ogni $m > 0$

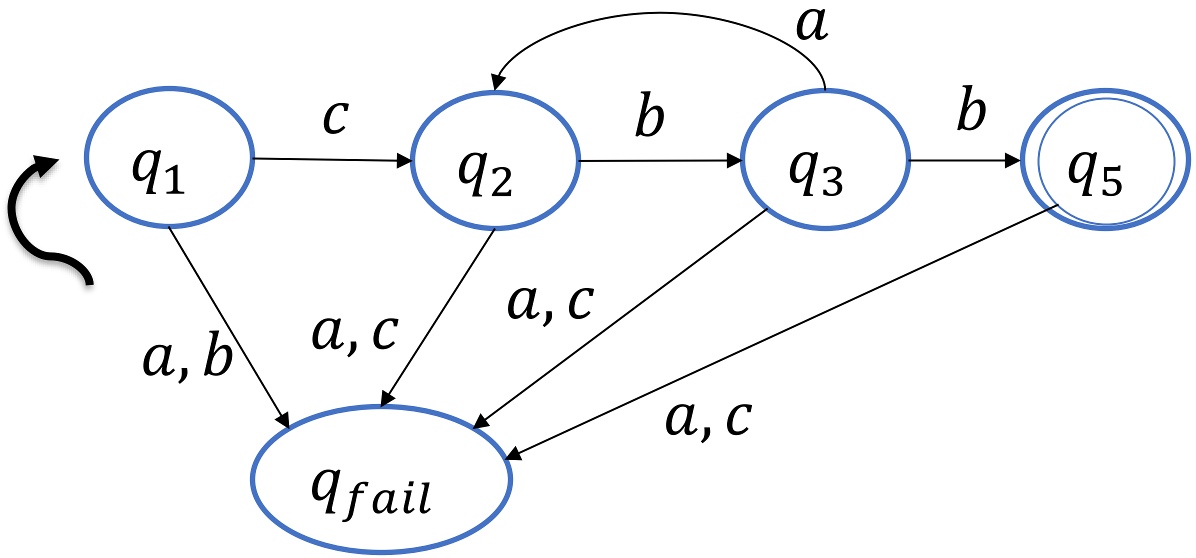
- **Nota**. u, v, w potrebbero essere diverse per stringhe diverse
  – Nessuno ci garantisce che il prefisso sia sempre lo stesso (vedo il prossimo esempio)

Pumping Lemma – Esempio 2



- Sia $\mathcal{L}$ il linguaggio definito dall'ASFD A in figura
- Sia $n = |Q| + 1 = 5$ e s una qualunque stringa in $L(A)$ con $|s| \geq 5$
- **Pumping Lemma**. $s = uvw$ tale che $uv^m w \in L(A)$, per ogni $m > 0$
- In particolare possiamo assumere
  - $w = \epsilon$; $u = a$; $v =$ sottostringa a partire dal secondo simbolo
  - $w = \epsilon$; $u = b$; $v =$ sottostringa a partire dal secondo simbolo
- **Esempio 1**. $s = abbbb$
  - $w = \epsilon$; $u = a$; $v = bbbb$
  - $uv^m w = a(bbbb)^m \in L(A)$ per ogni $m > 0$
- **Esempio 2**. $s = baa$
  - $w = \epsilon$; $u = b$; $v = aa$
  - $uv^m w = b(aa)^m \in L(A)$ per ogni $m > 0$

Pumping Lemma – Esempio



$L(A) = \{cb(ab)^n b \mid per\ n \geq 0\}$

- Sia $\mathcal{L}$ il linguaggio definito dall'ASFD A in figura
- Sia $n = |Q| + 1 = 6$ e s una qualunque stringa in $L(A)$ con $|s| \geq 6$
- **Pumping Lemma**. $s = uvw$ tale che $uv^m w \in L(A)$, per ogni $m > 0$
- **Esempio 1**. $s = cababb$
  - $w = b$; $u = c$; $v = abab$
  - $uv^m w = c(abab)^m c \in L(A)$ per ogni $m > 0$

Applicazioni del Pumping Lemma

Linguaggi Non regolari

- La nostra investigazione era partita dalla ricerca di un **linguaggio non regolare**
  – Sospettiamo che il linguaggio delle parentesi ben formate non lo sia…

- Il **Pumping Lemma** non ci serve per **generare stringhe di un linguaggio**
  – Potrebbe essere utilizzato anche per questo scopo ma ..
  – È uno strumento macchinoso e abbiamo altri strumenti per generare stringhe
  – Per generare una stringa ci basta esplorare l'automa che riconosce il linguaggio

- Il **Pumping Lemma** ci serve per **dimostrare** che **un linguaggio non è regolare**!!
  – Utilizzandolo con la dovuta accortezza ☺

- **Corollario**. Sia $\mathcal{L}$r un linguaggio. Se non esiste $n \in \mathbb{N}$ con le proprietà del Pumping Lemma allora $\mathcal{L}$ non è un linguaggio regolare
  – Non esiste un ϵ-ASFND A tale che $\mathcal{L} = L(A)$

Linguaggi non Regolari – Esempio 1

- **Teorema**. Il linguaggio $\mathcal{L} = \{a^k b^k | k \geq 0\}$ non è un linguaggio regolare

- **Dimostrazione**: Dimostrazione per assurdo. Assumiamo che l'enunciato sia falso e ricaviamo una contraddizione delle ipotesi

1. Si assuma, per assurdo, che **$\mathcal{L}$ sia regolare**.
2. Supponiamo che $n \in \mathbb{N}$ sia il valore tale che per tutte le $s \in \mathcal{L}$ con $|s| \geq n$ esistano u, v, w tale che $s = uvw$, $|v| \geq 1$, $|uv| \leq n$ e $uv^i w \in \mathcal{L}$, per ogni $i > 0$. **(Pumping Lemma)**
3. Sia $z = a^m b^m \in L$, con $m > n$ (chiaramente $|z| = 2m > 2n > n$) allora esistono u, v e w tale che $z = uvw$, $|v| \geq 1$, $|uv| \leq n$ e $uv^i w \in L$ per ogni $i \geq 0$
4. Poiché $m > n$ e poiché $|uv| \leq n$, $u = a^l$ e $v = a^h$ per due interi positivi l e h con $l + h \leq n$
5. Possiamo concludere che L contiene tutte le stringhe $a^l (a^h)^i a^{m-l-h} b^m$, per ogni $i > 0$,
6. Per $i = 2$, Punto 5 contraddice l'ipotesi che il linguaggio L sia $\{a^n b^n | n \geq 0\}$

Linguaggi non Regolari – Esempio 2

- **Definizione**. Una stringa s sull'alfabeto $\Sigma = \{(,)\}$ è una stringa di parentesi ben formata se
  - $s = (\,)$ oppure
  - $s = (\,p\,)$ e p è una stringa ben formata
- **Teorema**. Il linguaggio delle parentesi ben formate non è un linguaggio regolare
- **Dimostrazione**: Dimostrazione per assurdo.

1. Si assuma, per assurdo, che $\mathcal{L}$ **sia regolare**.
2. Supponiamo che $n \in \mathbb{N}$ sia il valore tale che per tutte le $s \in \mathcal{L}$ con $|s| \geq n$ esistano u, v, w tale che $s = uvw$, $|v| \geq 1$, $|uv| \leq n$ e $uv^iw \in \mathcal{L}$, per ogni $i > 0$. **(Pumping Lemma)**
3. Sia $z = (^m)^m \in L$, con $m > n$ (chiaramente $|z| = 2m > 2n > n$) allora esistono u, v e w tale che $z = uvw$, $|v| \geq 1$, $|uv| \leq n$ e $uv^iw \in L$ per ogni $i \geq 0$
4. Poiché $m > n$ e poiché $|uv| \leq n$, $u = (^l$ e $v = (^h$ per due interi positivi l e h con $l + h \leq n$
5. Possiamo concludere che L contiene tutte le stringhe $(^l(^{h \cdot i}(^{m-l-h})^m$, per ogni $i > 0$,
6. Per $i = 2$, Punto 5 contraddice l'ipotesi che il linguaggio L il linguaggio delle parentesi ben formate

Linguaggi non Regolari – Strategia di Prova

- È possibile dimostrare che un linguaggio **non è regolare** usando la seguente strategia di prova
  - Ovviamente ce e sono altre
- Sia $\mathcal{L}$ un linguaggio per cui vogliamo dimostrare che $\mathcal{L}$ non è regolare

1. Si assuma, per assurdo, che **$\mathcal{L}$ sia regolare**.

2. Supponiamo che $n \in \mathbb{N}$ sia il valore tale che per tutte le $s \in \mathcal{L}$ con $|s| \geq n$ esistano u, v, w tale che $s = uvw$, $|v| \geq 1$, $|uv| \leq n$ e $uv^iw \in \mathcal{L}$, per ogni $i > 0$. **(Pumping Lemma)**

3. Sia z una specifica stringa in $\mathcal{L}$ con $|z| > n$, allora esistono u, v e w tale che $z = uvw$, $|v| \geq 1$, $|uv| \leq n$ e $uv^iw \in L$ per ogni $i \geq 0$

4. Dimostriamo che, **comunque prese u e v**, per un qualche $k > 0$ uv^kw non è in $\mathcal{L}$

5. Possiamo concludere che **$\mathcal{L}$ non possiede la proprietà definita dal Pumping Lemma** e quindi **non $\mathcal{L}$ non è regolare**.