Linguaggi Regolari e Linguaggi Liberi

Potenza espressiva degli automi

Potenza espressiva delle grammatiche

Linguaggi Regolari

- Tutti i linguaggi che possono essere accettati da automi a stati finiti non deterministici sono detti **Linguaggi Regolari**
- La definizione include linguaggi su tutti i possibili alfabeti Σ **finiti**

Determinismo vs Non determinismo

- Abbiamo visto che, tramite la costruzione dei sottoinsiemi, un **qualsiasi** automa non deterministico può essere trasformato in uno deterministico **equivalente**
- Inoltre un automa deterministico può essere visto come un particolare automa non deterministico
- **Pertanto** in realtà è superfluo dire, nella definizione di linguaggi regolari, che gli automi debbano essere non deterministici
- La classe di linguaggi accettati da automi deterministici **è uguale** a quella dei linguaggi accettati da automi non deterministici

Potere espressivo

- In questo caso si dice che i due formalismi, gli automi deterministici e non deterministici, hanno lo stesso potere espressivo
- Oltre a quello degli automi deterministici, anche il formalismo delle espressioni regolari ha lo stesso potere espressivo degli automi non deterministici
- Una costruzione per costruire un NFA a partire da una qualsiasi espressione regolare si vedrà nel corso di Compilatori

Linguaggi Regolari

- La classe dei linguaggi regolari può essere quindi specificata equivalentemente con tre tipi diversi di formalismi:
  - Automi non deterministici (NFA)
  - Automi deterministici (DFA)
  - Espressioni Regolari
- Questi tre formalismi hanno tutti lo stesso potere espressivo

Potere espressivo superiore

- Ma tutti i linguaggi sono regolari?
- La risposta è NO
- Ci sono dei linguaggi che non possono essere specificati con NFA (o DFA o espressioni regolari)
- I linguaggi di questo tipo sono detti non regolari

Un esempio classico

- L’esempio classico di linguaggio non regolare è il seguente

$$L = \{ a^n b^n \mid n \geq 0 \}$$

- È impossibile scrivere un automa o un’espressione regolare che accetti/denoti questo linguaggio
- La dimostrazione di questo enunciato è interessante e può essere trovata sulla dispensa alternativa di sintassi

Limiti degli automi

- Intuitivamente il motivo è che gli automi non possono “contare”, cioè non possono implementare un contatore che possa assumere un **qualsiasi** valore intero (la n) in modo da accettare stringhe in cui ci sia una corrispondenza fra il numero di certi elementi e il numero di altri elementi

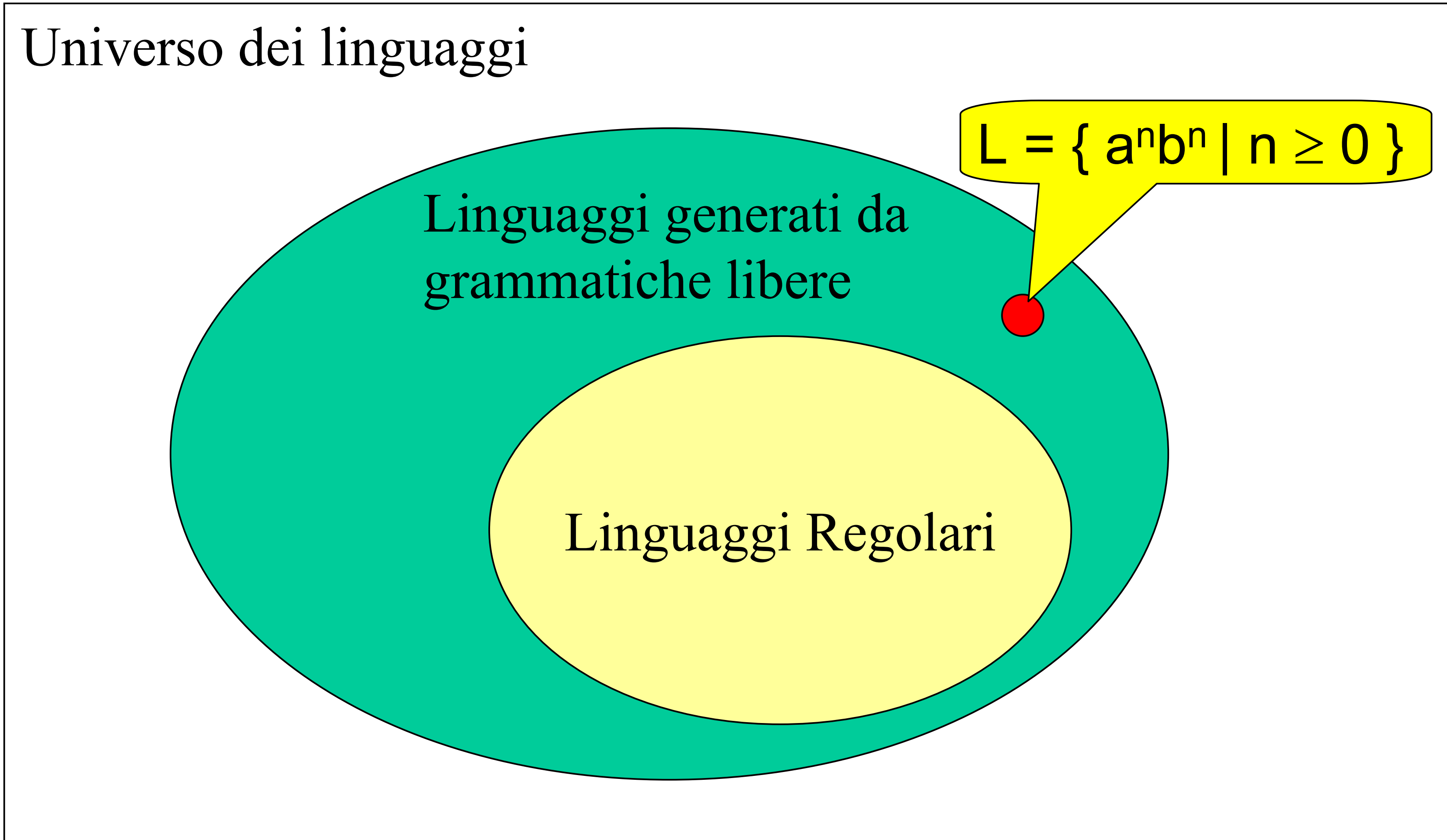
Potere espressivo delle grammatiche

- È facile scrivere una grammatica libera dal contesto per il linguaggio visto:

$$S \rightarrow aSb \mid \varepsilon$$

- In effetti si ha che le grammatiche libere dal contesto hanno un potere espressivo maggiore degli automi/espressioni regolari
- Ciò significa più precisamente che:
  - Ogni linguaggio regolare può essere generato con una grammatica
  - Esiste almeno un linguaggio non regolare che è accettato da una grammatica

Potere Espressivo delle grammatiche



Potere espressivo delle grammatiche

- Per mostrare la seconda parte basta usare il linguaggio $\{ a^n b^n \mid n \geq 0 \}$
- Per la prima parte definiamo un **algoritmo** che, dato un qualsiasi automa (deterministico o no), costruisce una grammatica equivalente

Algoritmo: dagli automi alle grammatiche

- Input: Automa <S, Σ, $\mathbf{s}_0$, **move**, F>
- Output: Grammatica <V, Σ, S, P> equivalente

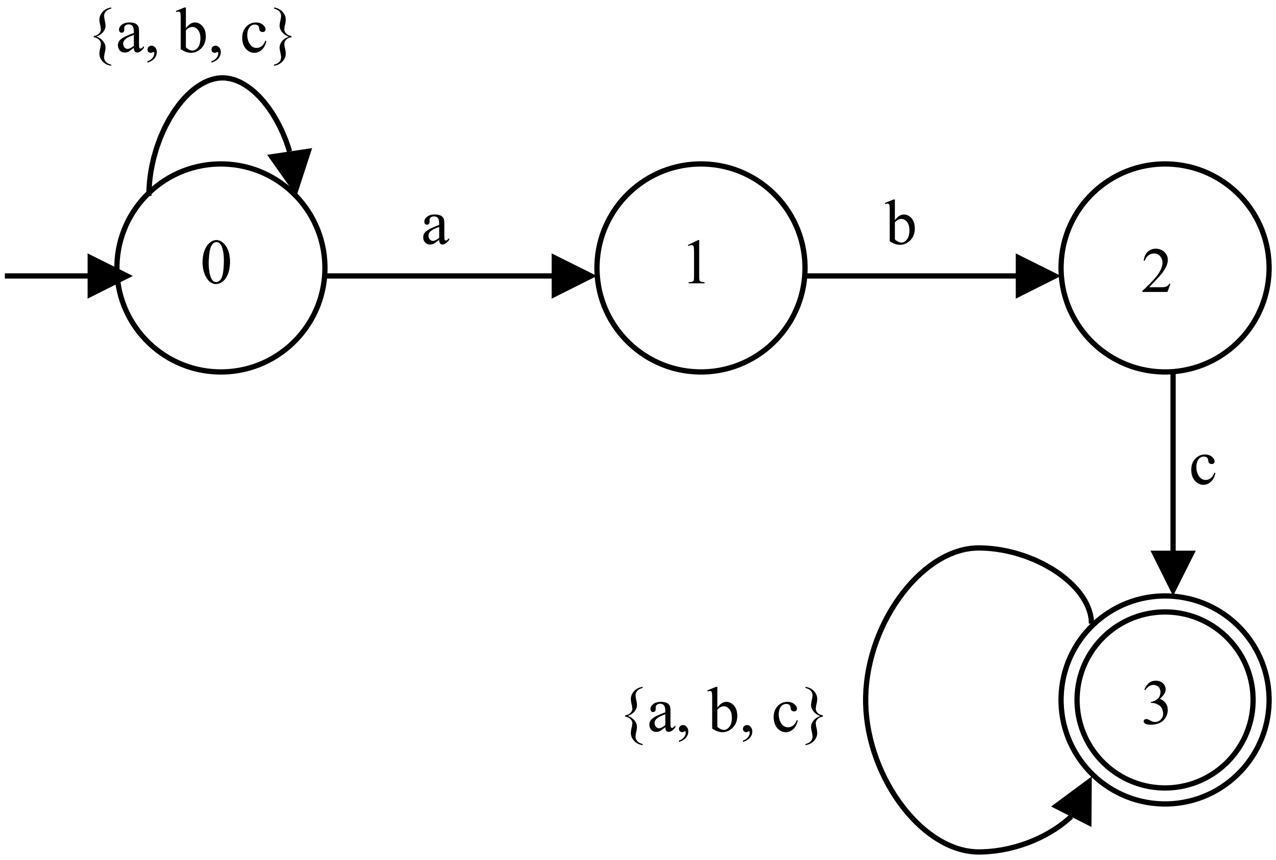
- Costruzione dei simboli non terminali di V e dello stato iniziale:
  - Per ogni stato s di S costruiamo un simbolo non terminale <s> in V
  - Poniamo <s_0> come simbolo iniziale S della grammatica

Algoritmo: dagli automi alle grammatiche

Costruzione delle produzioni P:

- Per ogni s $\in$ S e per ogni $a \in \Sigma$:
  - Se c'è nell'automa una transizione etichettata con a che dallo stato s va nello stato s' allora inseriamo la produzione <s> $\rightarrow$ a <s'> in P
  - Se c'è nell'automa una transizione etichettata con a che dallo stato s va nello stato s' **e** s' $\in$ F allora inseriamo una produzione <s> $\rightarrow$ a in P
- Se $s_0 \in$ F allora inseriamo la produzione <s_0> $\rightarrow$ ε in P

Esempio



Esempio

- Seguendo le indicazioni dell'algoritmo otteniamo la seguente grammatica:

<0> → a <0> | b <0> | c <0> | a <1>

<1> → b <2>

<2> → c <3> | c

<3> → a <3> | b <3> | c <3> | a | b | c

Grammatiche regolari

- Le grammatiche in cui le produzioni sono tutte della forma

  $A \rightarrow aB$ oppure

  $A \rightarrow a$ oppure

  $A \rightarrow \varepsilon$

- Sono chiamate **Grammatiche Regolari** e hanno lo stesso potere espressivo degli automi/espressioni regolari
- L’algoritmo visto genera grammatiche regolari

Linguaggi liberi

- Tutti i linguaggi che possono essere specificati attraverso una grammatica libera dal contesto sono detti **Linguaggi Liberi**
- I linguaggi liberi, sappiamo dal potere espressivo, includono i linguaggi regolari più altri linguaggi non regolari come $\{ a^n b^n \mid n \geq 0 \}$
- Ma i linguaggi non liberi sono tutti i possibili linguaggi?
- La risposta è ancora NO

Linguaggi non liberi

- Esistono dei linguaggi che non possono essere generati da nessuna grammatica libera dal contesto:

  $L = \{w\ c\ w^R \mid w \in \Sigma^*, w^R \text{ è } w \text{ rovesciata}\}$

  $L = \{a^m\ b^m\ c^m \mid m \geq 0\}$

  ...
- Esistono infiniti linguaggi non liberi. Esistono infiniti linguaggi non regolari, ma liberi.
- Esistono gerarchie di formalismi più potenti delle grammatiche libere dal contesto

Potere Espressivo delle grammatiche