

# Automi e Linguaggi Formali

a.a. 2017/2018

LT in Informatica  
12 Marzo 2018

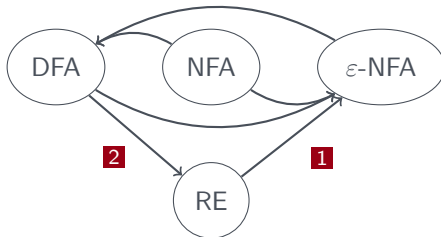


UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Le **espressioni regolari** sono un metodo alternativo per descrivere i linguaggi regolari, costruite utilizzando

- un insieme di **costanti** di base:
  - $\epsilon$  per la stringa vuota
  - $\emptyset$  per il linguaggio vuoto
  - $a, b, \dots$  per i simboli  $a, b, \dots \in \Sigma$
- collegati da **operatori**:
  - $+$  per l'unione
  - $\cdot$  per la concatenazione
  - $*$  per la chiusura di Kleene
- raggruppati usando le **parentesi**:
  - $( \ )$

Sappiamo già che DFA, NFA, e  $\epsilon$ -NFA sono tutti equivalenti.



Gli FA sono equivalenti alle espressioni regolari:

- 1** Per ogni espressione regolare  $R$  esiste un  $\epsilon$ -NFA  $A$ , tale che  $L(A) = L(R)$
- 2** Per ogni FA  $A$  possiamo costruire un'espressione regolare  $R$ , tale che  $L(R) = L(A)$

## Theorem

*Per ogni espressione regolare  $R$  possiamo costruire un  $\varepsilon$ -NFA  $A$  tale che  $L(A) = L(R)$*

## Dimostrazione:

Costruiremo un  $\varepsilon$ -NFA  $A$  con:

- un solo stato finale
- nessuna transizione entrante nello stato iniziale
- nessuna transizione uscente dallo stato finale

La dimostrazione è per induzione strutturale su  $R$

## Caso Base:

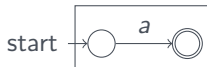
- automa per  $\varepsilon$



- automa per  $\emptyset$

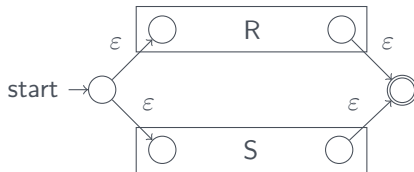


- automa per  $a$



## Caso Induttivo:

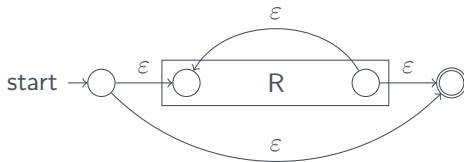
- automa per  $R + S$



- automa per  $RS$



- automa per  $R^*$



- 1 Trasformiamo  $(0 + 1)^*1(0 + 1)$  in  $\varepsilon$ -NFA
- 2 Scrivere un'espressione regolare per rappresentare il linguaggio sull'alfabeto  $\{a, b, c\}$  che contiene
  - tutte le stringhe che iniziano con  $a$  e sono composte solo di  $a$  oppure  $b$ ;
  - la stringa  $c$
- 3 Trasformare l'espressione regolare dell'esercizio 2 in  $\varepsilon$ -NFA

- 4 Scrivere una espressione regolare per tutte stringhe binarie che cominciano e finiscono per 1
- 5 Scrivere una espressione regolare per le stringhe binarie che contengono almeno tre 1 consecutivi
- 6 Scrivere una espressione regolare per le stringhe binarie che contengono almeno tre 1 (anche non consecutivi)
- 7 Scrivere una espressione regolare per stringhe di testo che descriva le date in formato GG/MM/AAAA