

# Computational Linguistics

## Lecture 2 – Finite State Automata

Clayton Greenberg  
Stefan Thater

FR 4.7 Allgemeine Linguistik (Computerlinguistik)  
Universität des Saarlandes  
Summer 2015

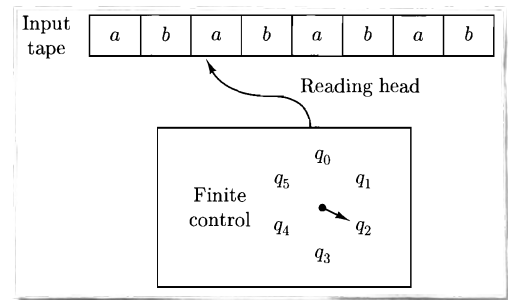
### Some basic definitions

- **An alphabet  $\Sigma$**  is a finite set of symbols
- **A string over  $\Sigma$**  is a sequence of symbols from  $\Sigma$ 
  - $\epsilon$  stands for the empty string
- **The length  $|w|$**  is the number of symbols in  $w$
- **$\Sigma^*$**  denotes this set of all strings over  $\Sigma$
- **A (formal) language** is a subset of  $\Sigma^*$  for some alphabet  $\Sigma$

# Deterministic Finite Automata

- $M = \langle K, \Sigma, \delta, s, F \rangle$

- $K$  is a finite set of states
- $\Sigma$  is an input alphabet
- $\delta$  is a transition function
- $s \in K$  is the start state
- $F \subseteq K$  is the set of final (accepting) states



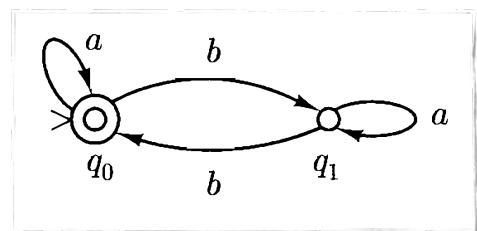
- **Transition function**

- $\delta(q, a) = q'$
- when  $M$  is in state  $q$  and reads input  $a$ , it goes into state  $q'$

3

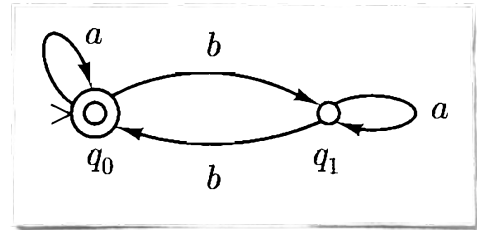
# Automata as Graphs

- Nodes = states
- Edges = transition function
  - an edge from state  $q$  to state  $q'$  labeled by  $a \Leftrightarrow \delta(q, a) = q'$
- $\circ$  marks the start state
- Double circles = final states



4

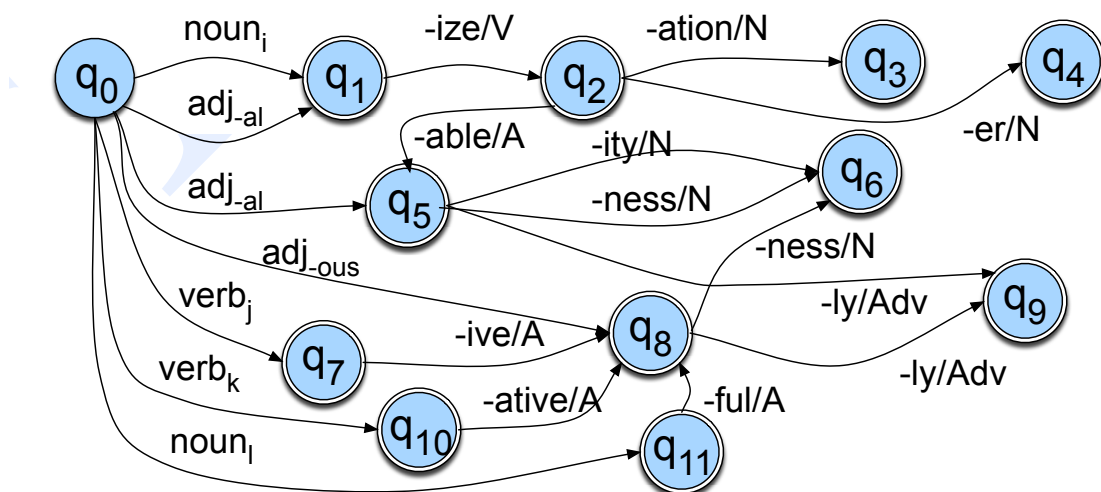
# Automata as Graphs



- $M = \langle K, \Sigma, \delta, s, F \rangle$

- $K = \{q_0, q_1\}$
- $\Sigma = \{a, b\}$
- $s = q_0$
- $F = \{q_0\}$
- $\delta(q_0, a) = q_0$
- $\delta(q_0, b) = q_1$
- $\delta(q_1, a) = q_0$
- $\delta(q_1, b) = q_1$

# Another Example (from J&M)



## More definitions

- **A configuration** is a pair  $\langle q, w \rangle \in K \times \Sigma^*$ 
  - $q$  = the current state
  - $w$  = the unread part of the string being processed
- **Yields in one step**
  - $\langle q, w \rangle \vdash_M \langle q', w' \rangle$
  - iff  $w = aw'$  for some  $a \in \Sigma$ ,  $w' \in \Sigma^*$  and  $\delta(q, a) = q'$
- **Yields**
  - $\vdash_M^*$  is the reflexive, transitive closure of  $\vdash_M$
- **The language accepted** by a DFA  $M = \langle K, \Sigma, \delta, s, F \rangle$ 
  - $L(M) = \{ w \mid \langle s, w \rangle \vdash_M^* \langle q, \epsilon \rangle \text{ for some } q \in F \}$

7

## An Example

$\langle q_0, ababa \rangle \vdash_M \langle q_1, baba \rangle$

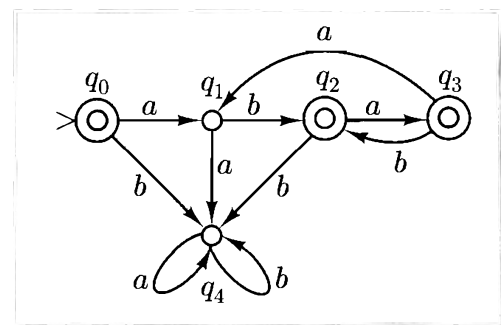
$\vdash_M \langle q_2, aba \rangle$

$\vdash_M \langle q_3, ba \rangle$

$\vdash_M \langle q_2, a \rangle$

$\vdash_M \langle q_3, \epsilon \rangle$

$\Rightarrow ababa \in L(M)$



8

# An Example

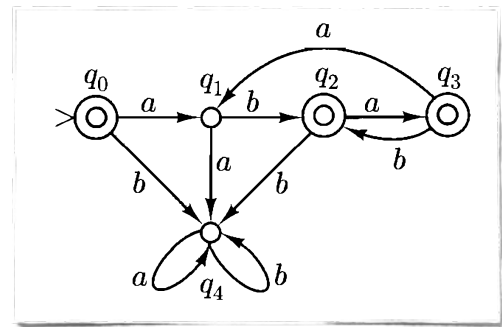
$\langle q_0, abaa \rangle \vdash_M \langle q_1, baa \rangle$

$\vdash_M \langle q_2, ba \rangle$

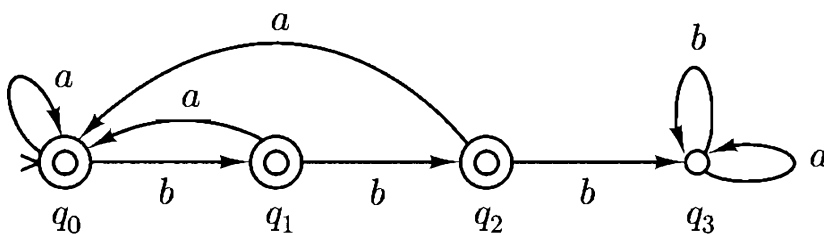
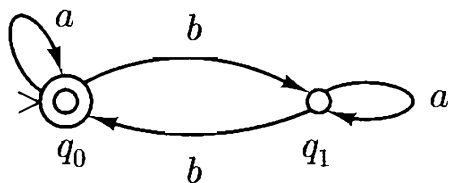
$\vdash_M \langle q_3, a \rangle$

$\vdash_M \langle q_1, \epsilon \rangle$

$\Rightarrow abaa \notin L(M)$



# Exercise: $L(M) = ?$



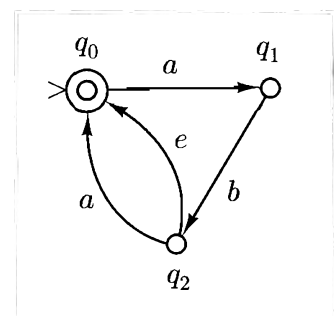
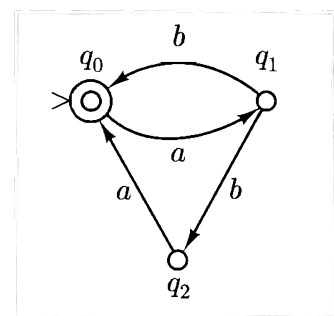
# Recognition Algorithm

```
function RECOGNIZE(DFA M, STRING input)
  index  $\leftarrow$  0
  state  $\leftarrow$  start state of M
  while index < length(input) do
    state  $\leftarrow$  trans[state, input[index]]
    index  $\leftarrow$  index + 1
  end
  if state is a final state of M
  then return accept
  else return reject
end
```

11

# Nondeterministic Automata

- Nondeterministic finite automata:
  - several symbols can be read at once, or none at all
  - several possible next states



12

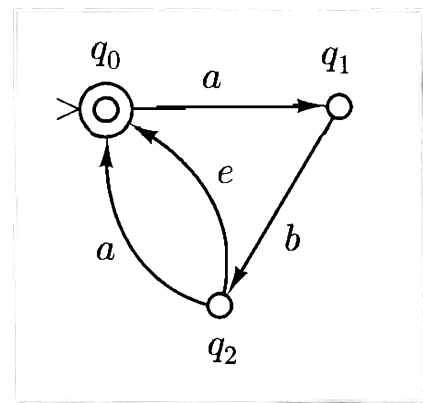
# Nondeterministic Automata

- $M = \langle K, \Sigma, \Delta, s, F \rangle$ 
  - $K$  is a finite set of states
  - $\Sigma$  is an input alphabet
  - $\Delta \subseteq K \times \Sigma^* \times K$  is a finite transition relation
  - $s \in K$  is the start state
  - $F \subseteq K$  is the set of final (accepting) states
- **Transition relation**  $\Delta \subseteq K \times \Sigma^* \times K$ 
  - $\langle q, w, q' \rangle \in \Delta =$  when the automaton is in state  $q$  and reads input  $w$ , it can go into state  $q'$
  - Note: here we restrict ourself to NFA where  $|w| \leq 1$

13

## An Example

- $M = \langle K, \Sigma, \Delta, s, F \rangle$ 
  - $K = \{q_0, q_1, q_2\}$
  - $\Sigma = \{a, b\}$
  - $s = q_0$
  - $F = \{q_0\}$
  - $\Delta = \{\langle q_0, a, q_1 \rangle, \langle q_1, b, q_2 \rangle, \langle q_2, a, q_0 \rangle, \langle q_2, \epsilon, q_0 \rangle\}$



14

# Configurations

- **Configurations**

- are elements from  $K \times \Sigma^*$  (as before)

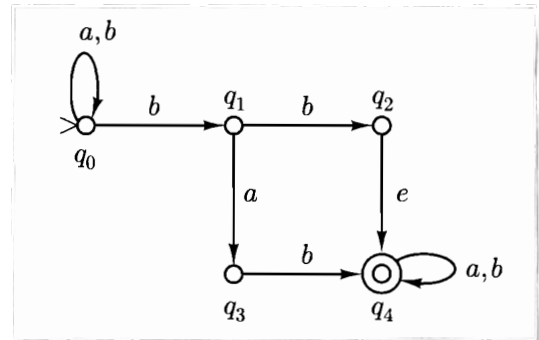
- **Yields in one step**

- $\langle q, w \rangle \vdash_M \langle q', w' \rangle$
- iff  $w = uw'$  for some  $u, w \in \Sigma^*$  and  $\langle q, u, q' \rangle \in \Delta$

- **The language accepted** by an NFA

- $L(M) = \{ w \mid \langle s, w \rangle \vdash_M^* \langle q, \varepsilon \rangle \text{ for some } q \in F \}$

## An Example



$\langle q_0, babba \rangle$

$\vdash_M \langle q_0, abba \rangle$

$\vdash_M \langle q_0, bba \rangle$

$\vdash_M \langle q_0, ba \rangle$

$\vdash_M \langle q_1, a \rangle$

$\vdash_M \langle q_3, \varepsilon \rangle$

$\langle q_0, babba \rangle$

$\vdash_M \langle q_1, abba \rangle$

$\vdash_M \langle q_3, bba \rangle$

$\vdash_M \langle q_4, ba \rangle$

$\vdash_M \langle q_4, a \rangle$

$\vdash_M \langle q_4, \varepsilon \rangle$

$\langle q_0, babba \rangle$

$\vdash_M \langle q_0, abba \rangle$

$\vdash_M \langle q_0, bba \rangle$

$\vdash_M \langle q_1, ba \rangle$

$\vdash_M \langle q_2, a \rangle$

$\vdash_M \langle q_4, a \rangle$

$\vdash_M \langle q_4, \varepsilon \rangle$

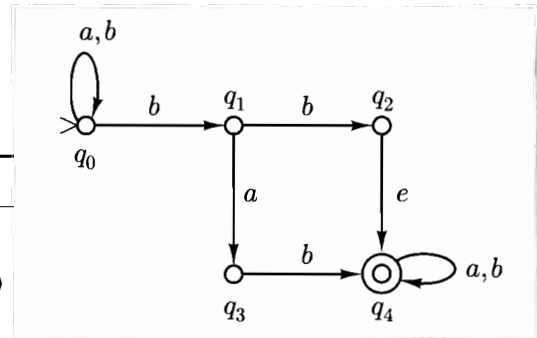
# Recognition Algorithm

```

function RECOGNIZE(NFA M, STRING input)
  agenda = list of configurations, initially containing only
           the configuration (start state of M, input)
  while agenda is not empty do
    conf  $\leftarrow$  POP(agenda)
    if conf is an accepting configuration then
      return accept
    else
      for all conf' such that conf  $\vdash$  conf' do
        PUSH(agenda, conf')
      end
    end
  return reject
end
  
```

17

## An Example



conf	agenda
-	$\langle q_0, babba \rangle$
$\langle q_0, babba \rangle$	<b><math>\langle q_0, abba \rangle</math></b> , $\langle q_1, abba \rangle$
$\langle q_0, abba \rangle$	<b><math>\langle q_0, bba \rangle</math></b> , $\langle q_1, abba \rangle$
$\langle q_0, bba \rangle$	<b><math>\langle q_0, ba \rangle</math></b> , <b><math>\langle q_1, ba \rangle</math></b> , $\langle q_1, abba \rangle$
$\langle q_0, ba \rangle$	<b><math>\langle q_0, a \rangle</math></b> , <b><math>\langle q_1, a \rangle</math></b> , $\langle q_1, ba \rangle$ , $\langle q_1, abba \rangle$
$\langle q_0, a \rangle$	<b><math>\langle q_0, \epsilon \rangle</math></b> , $\langle q_1, a \rangle$ , $\langle q_1, ba \rangle$ , $\langle q_1, abba \rangle$
$\langle q_0, \epsilon \rangle$	$\langle q_1, a \rangle$ , $\langle q_1, ba \rangle$ , $\langle q_1, abba \rangle$
$\langle q_1, a \rangle$	<b><math>\langle q_3, \epsilon \rangle</math></b> , $\langle q_1, ba \rangle$ , $\langle q_1, abba \rangle$
$\langle q_3, \epsilon \rangle$	$\langle q_1, ba \rangle$ , $\langle q_1, abba \rangle$
$\langle q_1, ba \rangle$	<b><math>\langle q_2, a \rangle</math></b> , $\langle q_1, abba \rangle$
$\langle q_2, a \rangle$	<b><math>\langle q_4, a \rangle</math></b> , $\langle q_1, abba \rangle$
$\langle q_4, a \rangle$	<b><math>\langle q_4, \epsilon \rangle</math></b> , $\langle q_1, abba \rangle$
$\langle q_4, \epsilon \rangle$	$\langle q_1, abba \rangle$

18

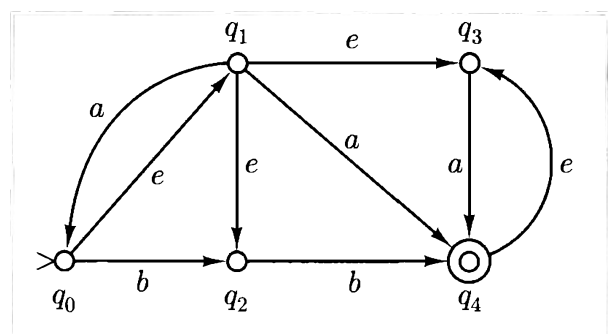
# NFA = DFA (preliminary)

- **Theorem:** for every NFA  $M = \langle K, \Sigma, \Delta, s, F \rangle$  there is an equivalent DFA  $M'$  such that  $L(M) = L(M')$
- Let us first consider the special case where for all elements  $\langle q, w, q' \rangle \in \Delta$ ,  $w$  is a string of length 1
- We construct the DFA  $M' = \langle K', \Sigma, \delta, s', F' \rangle$  as follows:
  - $K' = 2^K$
  - $s' = \{s\}$
  - $\delta(Q, a) = \{ k \in K \mid \langle q, a, k \rangle \in \Delta \text{ for some } q \in Q \}$ 
    - for all  $Q \subseteq K, a \in \Sigma$
  - $F' = \{ Q \subseteq K \mid Q \cap F \neq \emptyset \}$

19

# $\epsilon$ -Closure

- **$\epsilon$ -Closure**
  - $E(q) = \{ k \mid \langle q, \epsilon \rangle \vdash^* \langle k, \epsilon \rangle \}$
- Examples:
  - $E(q_0) = \{ q_0, q_1, q_2, q_3 \}$
  - $E(q_1) = \{ q_1, q_2, q_3 \}$
  - $E(q_2) = \{ q_2 \}$
- Note:
  - For all  $q, q \in E(q)$



20

## NFA = DFA

- **Theorem:** for every NFA  $M = \langle K, \Sigma, \Delta, s, F \rangle$  there is an equivalent DFA  $M'$  such that  $L(M) = L(M')$
- We construct the DFA  $M' = \langle K', \Sigma, \delta', s', F' \rangle$  as follows:
  - $K' = 2^K$
  - $s' = E(s)$
  - $\delta'(Q, a) = \cup \{ E(k) \in K \mid \langle q, a, k \rangle \in \Delta \text{ for some } q \in Q \}$ ,
    - for all  $Q \subseteq K$
  - $F' = \{ Q \subseteq K \mid Q \cap F \neq \emptyset \}$

21

## NFA = DFA

- **Lemma:** For any string  $w \in \Sigma^*$  and any states  $p, q \in K'$ :
  - $\langle q, w \rangle \vdash_M^* \langle p, \varepsilon \rangle$  iff  $\langle E(q), w \rangle \vdash_{M'}^* \langle P, \varepsilon \rangle$   
for some  $P$  containing  $p$
- Using this lemma, it is easy to show that  $L(M) = L(M')$ 
  - $w \in L(M)$
  - iff  $\langle s, w \rangle \vdash_M^* \langle f, \varepsilon \rangle$ , for some  $f \in F$
  - iff  $\langle E(s), w \rangle \vdash_{M'}^* \langle Q, \varepsilon \rangle$ , for some  $Q$  containing  $f$
  - iff  $\langle E(s), w \rangle \vdash_{M'}^* \langle Q, \varepsilon \rangle$ , for some  $Q \in F'$
  - iff  $w \in L(M')$

22

# Subset construction algorithm

- **$\epsilon$ -closure( s )**  
returns the set of NFA states reachable from state s using  $\epsilon$ -transitions
- **$\epsilon$ -closure( T )**  
returns the set of NFA states reachable from some s in T using  $\epsilon$ -transition
- **move( T, a )**  
returns the set of NFA states to which there is transition for input  $a \in \Sigma$  from some state  $s \in T$

23

# Subset construction algorithm

```
function DFA(K,  $\Sigma$ ,  $\Delta$ , s, F)
  K'  $\leftarrow$  list that contains only  $\epsilon$ -closure(s), unmarked
  while there is an unmarked state T in K' do
    mark T
    for each symbol  $a \in \Sigma$  do
      U  $\leftarrow$   $\epsilon$ -closure(move(T, a))
      if U  $\notin$  K' then
        add U as an unmarked state to K'
       $\delta[T, a] \leftarrow$  U
    end
  end
  return <the corresponding DFA>
end
```

24

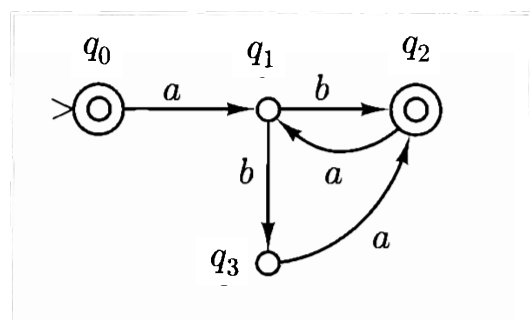
# Literature

- Jurafsky and Martin (2009). Chapter 2.
- Lewis and Papadimitriou (1981). Elements of the theory of computation. Chapter 2.

25

# Exercise 1

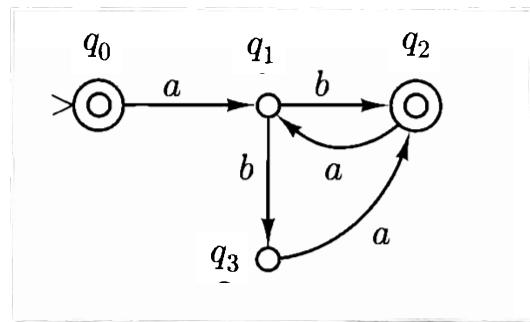
- Apply (using pen and paper) the recognition algorithm on slide 16 for the nondeterministic automaton shown below to the input string “abaaba”
- There is a problem with this algorithm. Which one? How can the algorithm be improved?



26

## Exercise 2

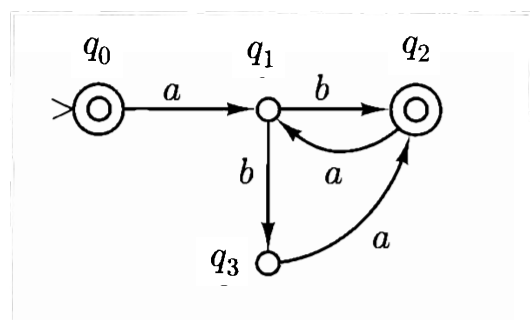
- Construct a deterministic automaton for the nondeterministic automaton shown below, using the subset construction algorithm on slide 23.



27

## Exercise 3

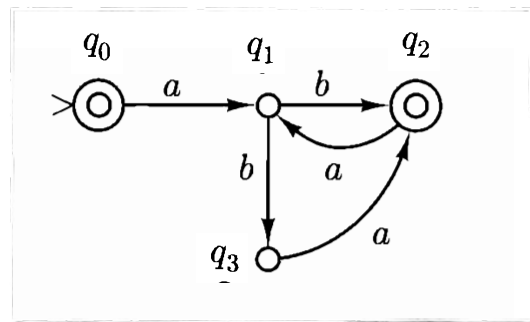
- Implement the recognition algorithm for NFA on slide 16.
- Your submission should use the automaton shown below and the following inputs as test case.
  - $ab \in L(M)$
  - $aba \in L(M)$
  - $abaab \in L(M)$
  - $abba \notin L(M)$
  - $aabab \notin L(M)$
  - More test cases are welcome!



28

## Exercise 4

- Implement the subset construction algorithm on slide 23.
- Your submission should use the automaton show below as a test case.



29

## Exercises – Remarks

- Submit the source code by email to me (stth@...)
- The source code should contain a comment that tells **how to run your code.**

30