Introduction to Computational Linguistics

Frank Richter

fr@sfs.uni-tuebingen.de.

Seminar für Sprachwissenschaft

Eberhard-Karls-Universität Tübingen

Germany

Incremental Linguistic Analysis

- tokenization
- morphological analysis (lemmatization)
- part-of-speech tagging
- named-entity recognition
- partial chunk parsing
- full syntactic parsing
- semantic and discourse processing

Finite State Technology

Regular languages and finite state automata

- deterministic finite state automata,

Finite State Technology

Regular languages and finite state automata

- deterministic finite state automata,
- nondeterministic finite state automata,

Finite State Technology

Regular languages and finite state automata

- deterministic finite state automata,
- nondeterministic finite state automata,
- finite state automata, and

Finite State Technology

Regular languages and finite state automata

- deterministic finite state automata,
- nondeterministic finite state automata,
- finite state automata, and
- regular expressions

Finite State Technology

Regular languages and finite state automata

- deterministic finite state automata,
- nondeterministic finite state automata,
- finite state automata, and
- regular expressions

characterize the same class of languages, *viz.* Type 3 languages

Regular Expressions

Given an alphabet Σ of symbols the following are all and only the regular expressions over the alphabet $\Sigma \cup \{Ø, 0, |, *, [,]\}$:

Ø	empty set	
0	the empty string	(ϵ, [])
σ	for all $\sigma \in \Sigma$	
[α | β]	union (for α, β reg.ex.)	($\alpha \cup \beta$, $\alpha + \beta$)
[α β]	concatenation (for α, β reg.ex.)	
[α*]	Kleene star (for α reg.ex.)	

Meaning of Regular Expressions

L(Ø) = $\emptyset$	the empty language
L(0) = $\{0\}$	the empty-string language
L(σ) = $\{\sigma\}$	
L([α | β]) = L(α) $\cup$ L(β)	
L([α β]) = L(α) $\circ$ L(β)	
L([α^*]) = (L(α))*	

Σ^* is called the universal language. Note that the universal language is given relative to a particular alphabet.

Remarks on Regular Expressions

- $\emptyset^* =_{def} \{0\}$
- The empty string, i.e., the string containing no character, is denoted by 0. The empty string is the neutral element for the concatenation operation. That is:

  $$\texttt{for any string } w \in \Sigma^* : \quad w0 = 0w = w$$

- Square brackets, [], are used for grouping expressions. Thus [A] is equivalent to A while (A) is not.
  We leave out brackets for readability if no confusion can arise.

Regular Expressions: Syntax

- ( ) is (sometimes) used for optionality; e.g. (A) ; definable in terms of union with the empty string.
- ? denotes any symbol; L(?) = Σ (our ? corresponds to # in the textbook by Kozen)
- A^+ denotes iteration; one or more concatenations of A. Equivalent to A (A*).
- Note the following simple expressions:
  - $[\,]$ denotes the empty-string language
  - ?* denotes the universal language (corresponds to @ in Kozen)

Deterministic Finite-State Automata

Definition 1 (DFA) A deterministic FSA (DFA) is a quintuple $(\Sigma, Q, i, F, \delta)$ where

- Σ is a finite set called *the alphabet*,
- Q is a finite set of *states*,
- $i \in Q$ is the *initial state*,
- $F \subseteq Q$ the set of *final states*, and
- δ is the transition function from $Q \times \Sigma$ to Q.

Generalizing Finite-State Automata

Definition 2 (rNFA) A restricted nondeterministic finite-state automaton is a quintuple $(\Sigma, Q, i, F, \Delta)$ where

Σ is a finite set called *the alphabet*,

Q is a finite set of *states*,

$i \in Q$ is the *initial state*,

$F \subseteq Q$ the set of *final states*, and

$\Delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ is the set of edges (the transition *relation*).

Nondeterministic Finite-State Automata

Definition 3 (NFA) A nondeterministic finite-state automaton is a quintuple $(\Sigma, Q, S, F, \Delta)$ where

Σ is a finite set called *the alphabet*,

Q is a finite set of *states*,

$S \subseteq Q$ is the set of *initial states*,

$F \subseteq Q$ the set of *final states*, and

$\Delta \subseteq Q \times \Sigma^* \times Q$ is the set of edges (the transition *relation*).

Some Important Properties of FSAs (1)

- Determinization: For every nondeterministic finite-state automaton there exists an equivalent deterministic automaton.
- Minimization: For every nondeterministic finite-state automaton there exists an equivalent deterministic automaton with a minimal number of states.

What is in a State

Definition 4

Given a DFA M = $(\Sigma, Q, i, F, \delta)$,

a *state of M* is triple (x, q, y)

where $q \in Q$ and $x, y \in \Sigma^*$

The *directly derives* relation

Definition 5 (directly derives)

Given a DFA $(\Sigma, Q, i, F, \delta)$,

a state (x, q, y) *directly derives* state (x', q', y'):

$(x, q, y) \vdash (x', q', y')$ iff

1. there is $\sigma \in \Sigma$ such that y = σy' and x'= xσ (i.e. the reading head moves right one symbol σ)
2. $\delta(q, \sigma) = q'$

The *derives* relation

Definition 6 (derives)

Given a DFA $(\Sigma, Q, i, F, \delta)$,

a state A *derives* state B:

$(x, q, y) \vdash^{*} (x', q', y')$ iff

there is a sequence $S_0 \vdash S_1 \vdash \cdots \vdash S_k$

such that A = S_0 and B = S_k

Acceptance

Definition 7 (Acceptance)

Given a DFA $M = (\Sigma, Q, i, F, \delta)$ and a string $x \in \Sigma^*$,

M *accepts* x iff

there is a $q \in F$ such that $(0, i, x) \vdash {}^*(x, q, 0)$.

Language accepted by M

Definition 8 (Language accepted by M)

Given a DFA $M = (\Sigma, Q, i, F, \delta)$, the language $L(M)$ accepted by M is the set of all strings accepted by M.

Example of String Acceptance

Let M = $(\{a, b\}, \{q_0, q_1, q_2\}, q_0, \{q_1\}, \{((q_0, a), q_1), ((q_0, b), q_1), ((q_1, a), q_2), ((q_1, b), q_2), ((q_2, a), q_2), ((q_2, b), q_2), \})$.

Example of String Acceptance

Let M = $(\{a,b\},\{q_0,q_1,q_2\},q_0,\{q_1\},\{((q_0,a),q_1),((q_0,b),q_1),$
$((q_1,a),q_2),((q_1,b),q_2),((q_2,a),q_2),((q_2,b),q_2),\})$.

M accepts a and b and nothing else, i.e. $L(M) = \{a,b\}$, since

$(0,q_0,a) \vdash (a,q_1,0)$ and
$(0,q_0,b) \vdash (b,q_1,0)$

are the only derivations from a start state to a final state for M.