


Lecture 3

Regular Expressions and Automata



CS 6320

Outline

- Regular Expressions
- Finite State Automata

The Problem of Information Extraction

Sample text:

Citing high fuel prices, [ORG United Airlines] said [TIME Friday] it has increased fares by [MONEY \$6] per round trip on flights to some cities also served by lower-cost carriers. [ORG American Airlines], a unit of [ORG AMR Corp.], immediately matched the move, spokesman [PERSON Tim Wagner] said. [ORG United Airlines] an unit of [ORG UAL Corp.], said the increase took effect [TIME Thursday] and applies to most routes where it competes against discount carriers, such as [LOC Chicago] to [LOC Dallas] and [LOC Denver] to [LOC San Francisco].

- Identify named entities
- Identify relations

Template Filling

- Example template for “airfare raise”

FARE-RAISE ATTEMPT:	[LEAD AIRLINE:	UNITED AIRLINES]
		AMOUNT:	\$6	
		EFFECTIVE DATE:	2006-10-26	
		FOLLOWER:	AMERICAN AIRLINES]

List of Named Entity Types

Type	Tag	Sample Categories
People	PER	Individuals, fictional characters, small groups
Organization	ORG	Companies, agencies, political parties, religious groups, sports teams
Location	LOC	Physical extents, mountains, lakes, seas
Geo-Political Entity	GPE	Countries, states, provinces, counties
Facility	FAC	Bridges, buildings, airports
Vehicles	VEH	Planes, trains, and automobiles

Examples of Named Entity Types

Type	Example
People	<i>Turing</i> is often considered to be the father of modern computer science.
Organization	The <i>IPCC</i> said it is likely that future tropical cyclones will become more intense.
Location	The <i>Mt. Sanitas</i> loop hike begins at the base of <i>Sunshine Canyon</i> .
Geo-Political Entity	<i>Palo Alto</i> is looking at raising the fees for parking in the University Avenue district.
Facility	Drivers were advised to consider either the <i>Tappan Zee Bridge</i> or the <i>Lincoln Tunnel</i> .
Vehicles	The updated <i>Mini Cooper</i> retains its charm and agility.

Categorical Ambiguities

Name	Possible Categories
<i>Washington</i>	Person, Location, Political Entity, Organization, Facility
<i>Downing St.</i>	Location, Organization
<i>IRA</i>	Person, Organization, Monetary Instrument
<i>Louis Vuitton</i>	Person, Organization, Commercial Product

Categorical Ambiguity

[*PERS* Washington] was born into slavery on the farm of James Burroughs.

[*ORG* Washington] went up 2 games to 1 in the four-game series.

Blair arrived in [*LOC* Washington] for what may well be his last state visit.

In June, [*GPE* Washington] passed a primary seatbelt law.

The [*FAC* Washington] had proved to be a leaky ship, every passage I made...

Regular Expressions

- Regular Expressions (RE)
 - There are a few ways of viewing REs
 - to specify textual search strings
 - to describe finite state automata (FSA)
- RE are widely used
e.g.: Perl, emacs, vi, grep, Word, etc.

Examples of RE 1/2

/the/

/[tT]he/

/\b[tT]he\b/

/colou?r/

/[0-9]/

/[^A-Z]/

/[^\.]/

/beg.n/

/a*/

/aa*/

/[0-9]+/

/^The dog\.\$/

Examples of RE 2/2

- Disjunction, Grouping

/cat|dog/

/gupp(y|ies)/

/((Column [0-9]+ *)*)*/

- Precedence

Parenthesis	()	Highest
Counters	*+?{ }	
Sequences and anchors	The ^my end\$	
Disjunction		Lowest

Advanced Operators

RE	Expansion	Match	Examples
\d	[0-9]	any digit	Party_of_5
\D	[^0-9]	any non-digit	Blue_moon
\w	[a-zA-Z0-9_]	any alphanumeric/underscore	Daiyu
\W	[^\w]	a non-alphanumeric	!!!!
\s	[_\r\t\n\f]	whitespace (space, tab)	
\S	[^\s]	Non-whitespace	in_Concord

Figure 2.6 Aliases for common sets of characters.

RE	Match
*	zero or more occurrences of the previous char or expression
+	one or more occurrences of the previous char or expression
?	exactly zero or one occurrence of the previous char or expression
{n}	n occurrences of the previous char or expression
{n,m}	from n to m occurrences of the previous char or expression
{n, }	at least n occurrences of the previous char or expression

Figure 2.7 Regular expression operators for counting.

RE	Match	Example Patterns Matched
*	an asterisk “*”	“K_A*P*L*A*N”
\.	a period “.”	“Dr. Livingston, I presume”
\?	a question mark	“Why don’t they come and lend a hand?”
\n	a newline	
\t	a tab	

Figure 2.8 Some characters that need to be backslashed.

Finite State Automata 1/4

- REs can describe a FSA machine.
- FSAs are useful for NLP.
- FSA to recognize the "sheep language"
/ baa+! /

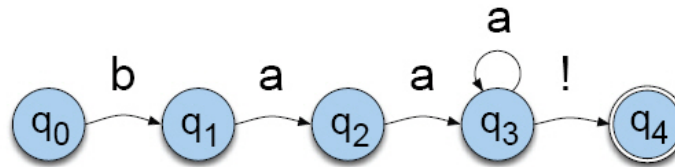


Figure 2.10 A finite state automaton for talking sheep.

- It has five states
- q_0 is the start state
- q_4 is the final state
- It has four transitions.

Finite State Automata 2/4

- FSAs can be encoded as tables.

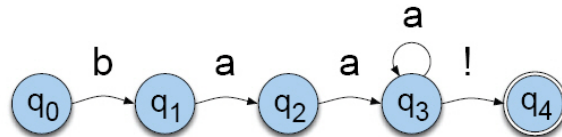


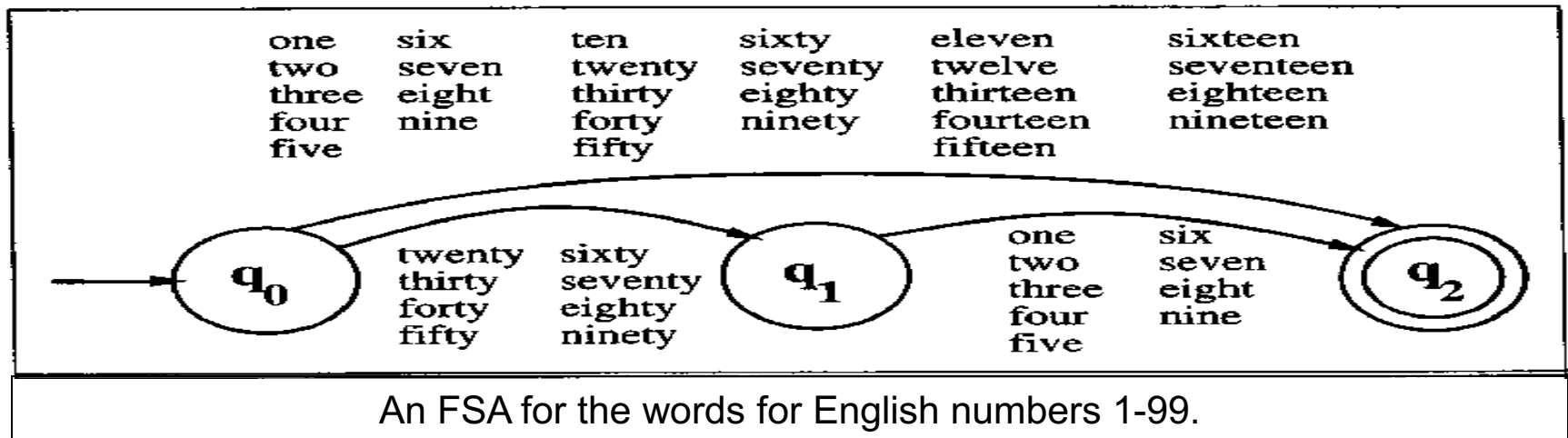
Figure 2.10 A finite-state automaton for talking sheep.

Input			
State	b	a	!
0	1	\emptyset	\emptyset
1	\emptyset	2	\emptyset
2	\emptyset	3	\emptyset
3	\emptyset	3	4
4:	\emptyset	\emptyset	\emptyset

state-transition table

Finite State Automata 3/4

- An example: FSA to recognize amounts of money.
- Ten cents, three dollars, one dollar thirtyfive cents



Finite State Automata 4/4

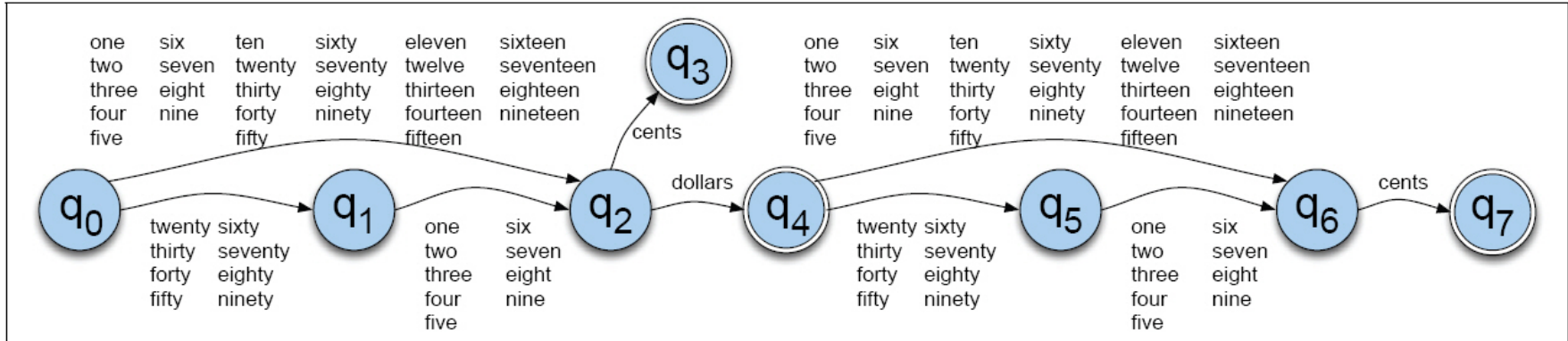


Figure 2.16 FSA for the simple dollars and cents

- FSAs can be formally specified as a 5tuple:
 - The set of states: Q
 - A finite alphabet: Σ
 - A start state
 - A set of final states
 - A transition function that maps $Q \times \Sigma$ to Q .

Recognition

- Recognition is the process of determining whether or not a given input is accepted by a machine.
- In terms of REs, it is the process of determining whether or not a given input matches a particular RE.
- Recognition is viewed as processing an input written on a tape consisting of cells containing elements from the alphabet.

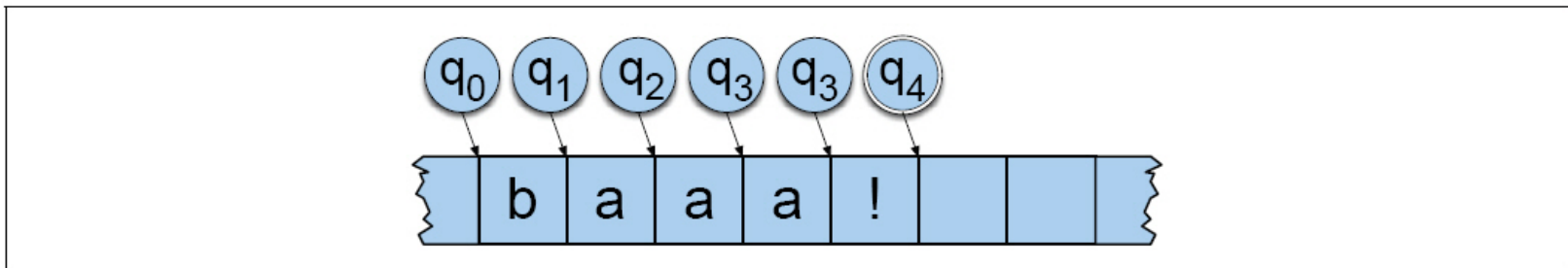


Figure 2.13 Tracing the execution of FSA #1 on some sheeptalk.

Deterministic vs. Non-Deterministic FSAs

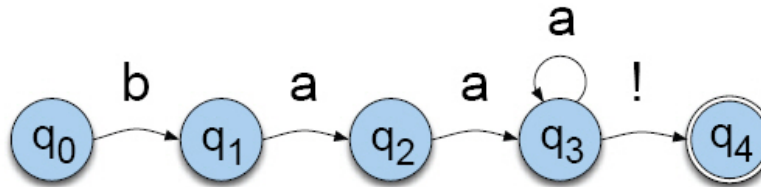


Figure 2.10 A finite-state automaton for talking sheep.

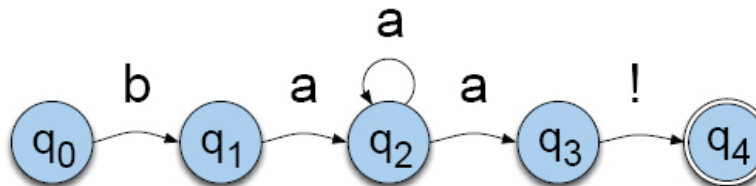


Figure 2.17 A non-deterministic finite-state automaton for talking sheep (NFA #1). Compare with the deterministic automaton in Fig. 2.10

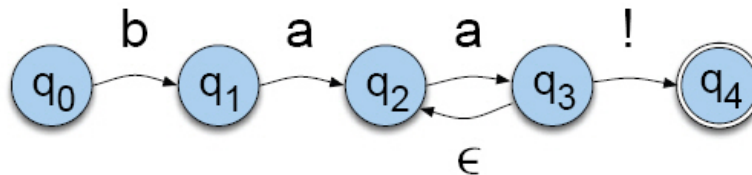


Figure 2.18 Another NFA for the sheep language (NFA #2). It differs from NFA #1 in Fig. 2.17 in having an ϵ -transition.

ND Recognition as Search 1/3

- Idea:
 - A search state is a pairing of a single machine state with a position on the input tape.
 - By keeping track of not yet explored search states, a recognizer can systematically explore all possible paths through a machine given some input.

ND Recognition as Search 2/3

- Depthfirst search or LIFO

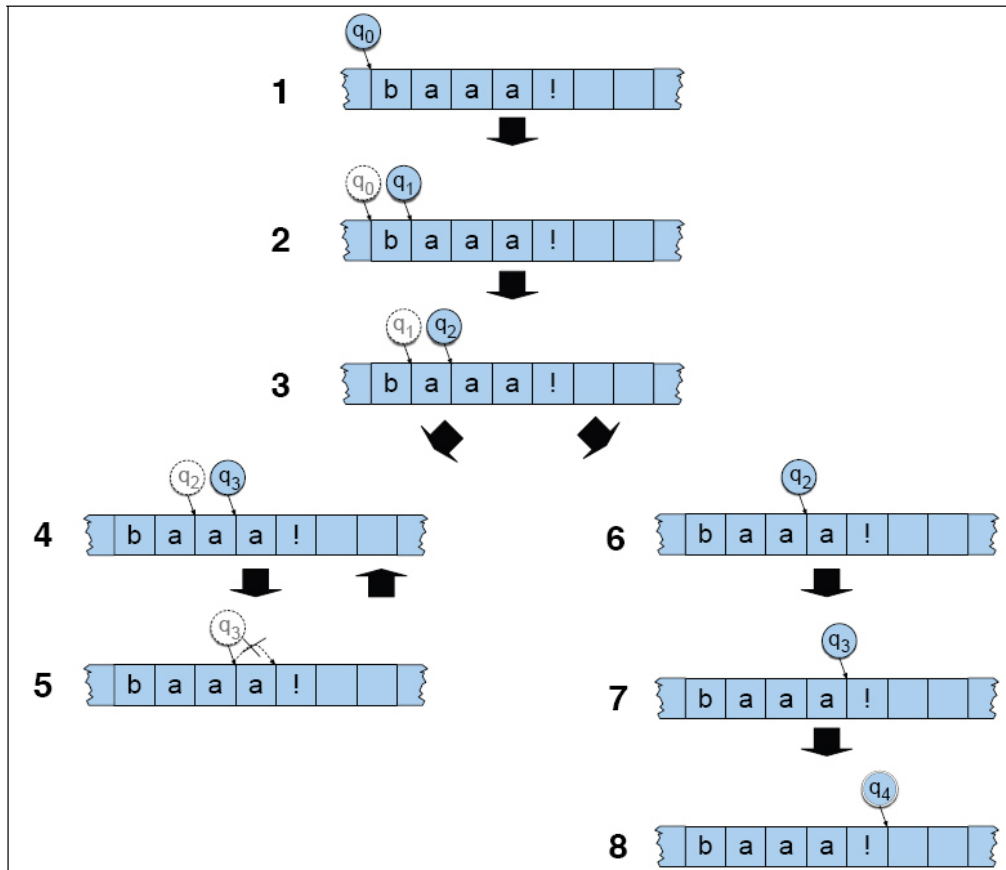


Figure 2.20 Tracing the execution of NFSA #1 (Fig. 2.17) on some sheeptalk.

ND Recognition as Search 3/3

- Breadthfirst search or FIFO

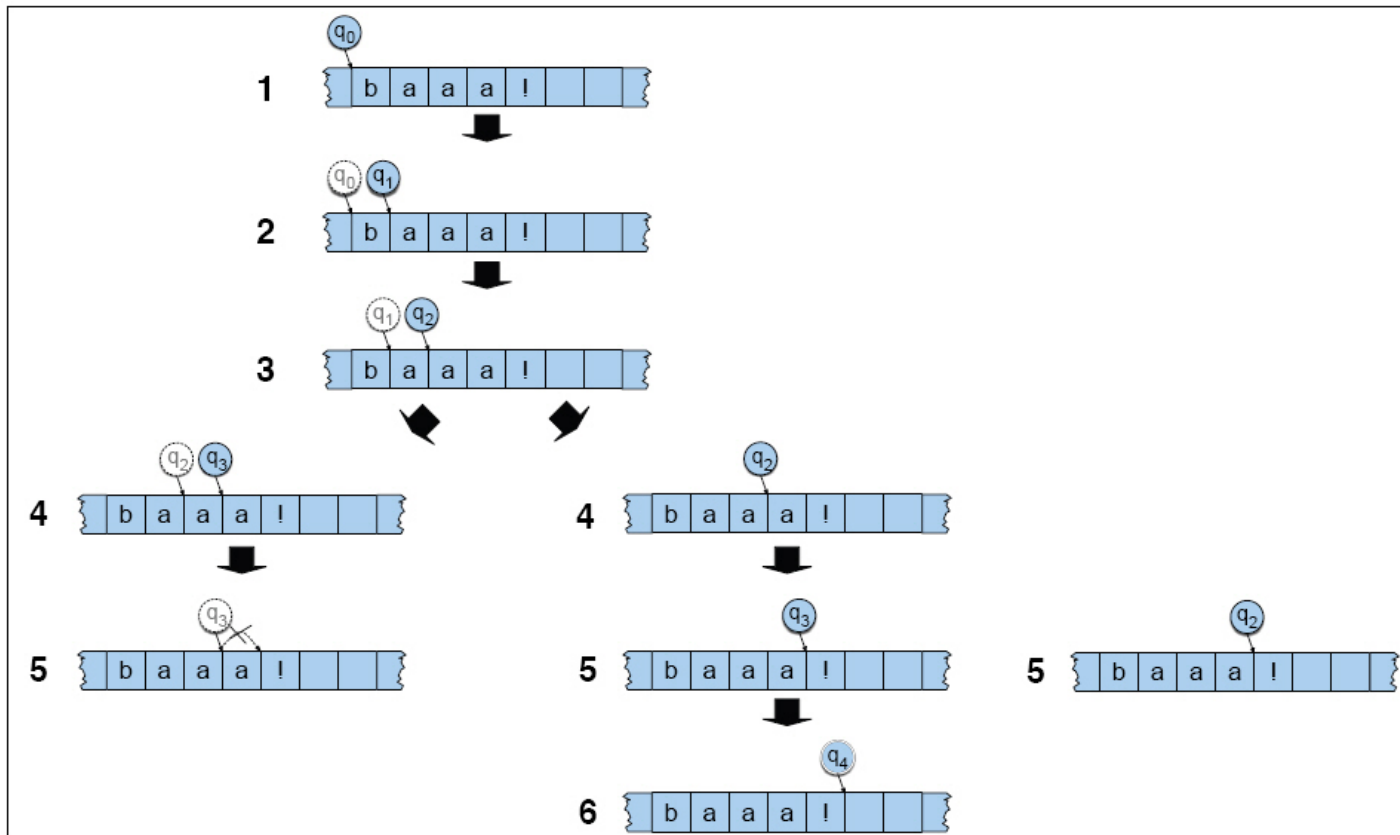


Figure 2.21 A breadth-first trace of FSA #1 on some sheeptalk

Generative Grammars and Formal Languages

- A Formal Language is a set of strings composed of symbols from a finite set of symbols.
- FSAs (and REs) define formal languages without having to explicitly enumerate the set.
- The term generative refers to the idea that FSAs can be viewed as generators of formal languages as well as acceptors.
- To generate you traverse the machine and transition symbols on the tape rather than reading them.