Finite State Automata

Ling 106
September 25, 2003

1. Syntax: Combining Words to Build Sentences

How do speakers of a language put together a finite number of discrete elements (e.g. words) to generate infinite number of sentences?

1.1. Memorization

Lookup table

$Sentence_1$	$Message_1$
$Sentence_2$	$Message_2$
$Sentence_3$	$Message_3$
$Sentence_n$	$Message_n$

Problems

- Novel sentences, infinite number of sentences
- Grammaticality judgments
  A genuinely novel sentence would have the same status as an ungrammatical string

  (1) a. *Tiger the ate basketball the (ungrammatical)
  b. The tiger ate the basketball (novel)

What is the simplest device that could compute the method of combining words necessary for natural language?

We can think of a sentence as a chain of words and a speaker as a device which consists of a finite number of mental predispositions, each associated with a rule that allows the speaker to produce a word and move to a new mental predisposition. That is, a speaker's grammar may be equivalent to a **finite state automaton** (FSA).

2. Example of a finite state automaton: automatic doors

Finite state automaton (FSA) is the simplest model of computation. Many useful devices can be modeled using FSA.

- Behavior of automatic doors:
  - The door can be in one of two states: OPEN or CLOSED

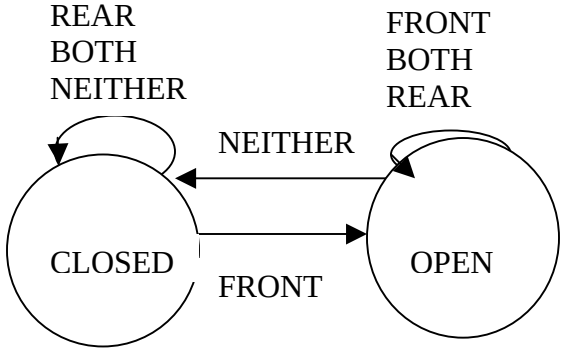
- If it is CLOSED and a person is standing on the pad in FRONT of the doorway, the door OPENs.
- If it is CLOSED and a person is standing on the pad to the REAR of the doorway, it remains CLOSED.
- If it is CLOSED and a person is standing on NEITHER pad, it remains CLOSED
- If it is CLOSED and people are standing on BOTH pads, it remains CLOSED
- If it is OPEN and a person is standing on the pad in FRONT of the doorway, it remains OPEN
- If it is OPEN and a person is standing on the pad to the REAR of the doorway, it remains OPEN
- If it is OPEN and a person is standing on NEITHER pad, it becomes CLOSED
- If it is OPEN and if people are standing on BOTH pads, it remains OPEN

- Behavior of automatic doors in tabular form

	NEITHER	FRONT	REAR	BOTH
CLOSED	CLOSED	OPEN	CLOSED	CLOSED
OPEN	CLOSED	OPEN	OPEN	OPEN

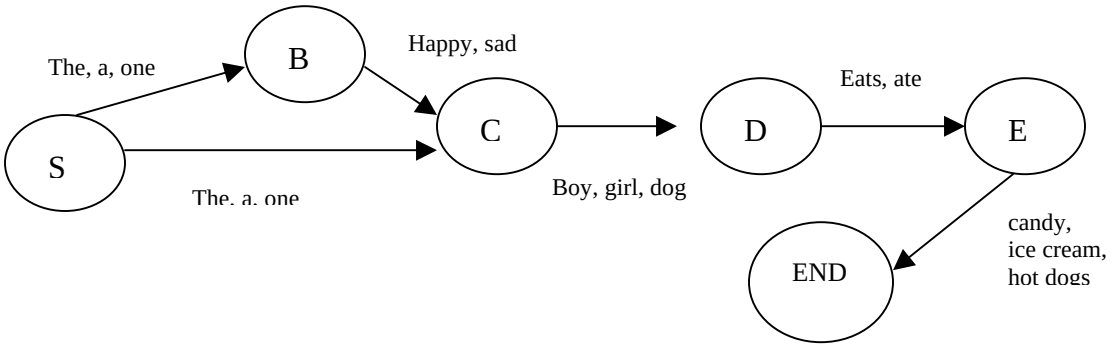
- Behavior of automatic doors in graph form.

States: OPEN, CLOSED
Input signals: FRONT, REAR, NEITHER, BOTH



3. A Grammar as a finite state automaton

- We could view a grammar for natural language as a finite state automaton with a number of states and a number of input signals or operations of the shape “add x”, where x is a word. For example, the following FSA can generate several English sentences.



Which of the following sentences are generated by this FSA?

(2) a. A happy boy eats sad ice cream.
b. The boy eats ice cream.
c. A boy ate dogs.
d. A happy boy ate hot dogs.
e. One ate candy.
f. One happy girl eats candy
g. One happy happy girl eats hot gods.
h. One happy girl eats hot hot dogs
i. A sad dog ate ice cream.

FSA based on grammatical categories.
The FSA above fails to capture the fact that some words pattern alike in terms of their distributional properties. Two words belong to the same **grammatical category** if one can be substituted for the other in sentences in a way that preserves grammaticality.

(3). a. The happy boy _____ the girl.
b. The happy _____ kisses the girl.
c. The happy boy kisses _____ girl.
d. The _____ boy kisses the girl.

When people learn language, they do not learn which word can follow which other word. Rather, they learn which grammatical category can follow which other category.

4. String, Alphabet, Language

Alphabet is any finite set. The members of the alphabet are the **symbols** of the alphabet.

$\Sigma_1 = \{0, 1\}$
$\Sigma_2 = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, r, s, t, u, v, w, x, y, z\}$
$\Gamma = \{0, 1, x, y, z\}$

A **string over an alphabet** is a finite sequence of symbols from that alphabet.

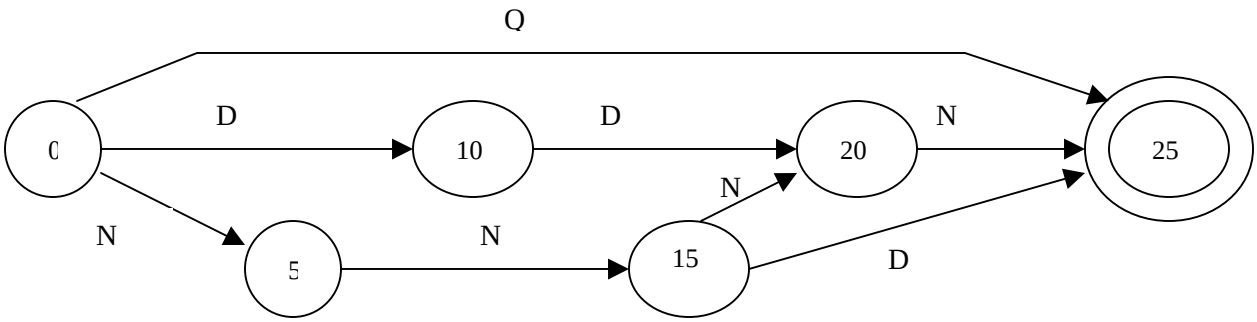
If $\Sigma_1 = \{0, 1\}$, then 011001 is a string over Σ_1.
If $\Sigma_2 = \{a, b, c, d, e, f, g, h, \ldots z\}$, then abracadabra is a string over Σ_2.
If w is a string over some alphabet, the **length** of w, written |w|, is the number of symbols that it contains.
|011001|=6
|abracadabra|=11
The string of length zero is called the **empty string** and is written ϵ.
The **reverse of string w,** written w^R, is the string obtained by writing w in the opposite order.
011001^R=100110
String z is a **substring** of w if z appears consecutively within w.
11 is a substring of 011001.
Given two strings x and y, the **concatenation** of x and y is the string obtained by appending y to the end of x.
Concatenation of 011001 and abracadabra is 011001abracadabra.

A **language** is a (possibly infinite) set of strings.

5. An example: the cola machine

- Characteristics of Cola machine
  1. Drinks cost 25 cents in US currency
  2. The only coins accepted by the machine are quarter (Q), dime (D), and nickel (N).
  3. The machine accepts any combination of these coins, in any order, that add up to 25 cents.
  4. The machine requires exact change.

- This cola machine is a finite state automaton. Before we put any money into it, it will be in a **start state.** Our job is to add appropriate coins that change the state of the machine until it is in a special **final state.** The machine will reach the final state (or **accept state)** when the total amount of money you inserted into the machine adds exactly up to 25 cents. By convention, we will use a double circle to mark such a final state.
- We can get to the final state in one step by inserting a quarter. But there are other ways of getting to the final state by inserting various combinations and permutations of nickels and dimes.



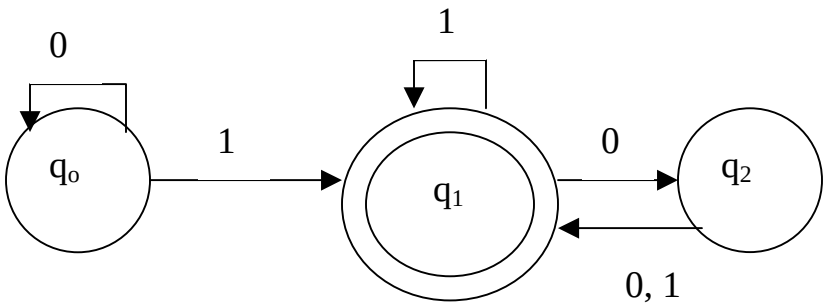
- List of all possible sequences of coins that the cola machine will accept, where acceptance means reaching the final state and allowing us to get a cola.
- Let us make a transition from a mechanical machine to a linguistic one.
  Think of the inputs to the machine not as coins but as symbols like Q, D, and N.
  The set of valid symbols that the machine will accept is its alphabet.
  The sequence of symbols that the machine will accept are strings.

  The entire set of words that the machine accepts or recognizers is its language.

  Questions:
  1. What is the alphabet of the cola machine?
  2. What are the strings that the cola machine accepts?
  3. What is the language of the cola machine?

6. How to read state diagrams

The following depicts a finite state automaton called M:



M has three states, labeled q0, q1, q2.
The start state is q0.
The final state (or accept state) is the one with a double circle, q1.
The arrows going from one state to another are called **transitions.**

When this automaton receives an input signal, it processes each symbol in that string from left to right, and produces an output. The output is either ACCEPT or REJECT. The processing begins in the machine start state, and after reading each symbol, it moves from one state to another along the transition that has that symbol as its label. When the machine reads the last symbol, the output is ACCEPT if the machine is in a final (accept) state, and REJECT if it is not.

When we feed the input string 1101 to machine M:

1. Start in state q0.
2. read 1, follow transition from q0 to q1.
3. read 1, follow transition from q1 to q1.

4. read 0, follow transition from q1 to q2.
5. read 1, follow transition from q2 to q1;
6. accept, because M is in an accept state q1 at the end of the input.

Questions:
Which of the following strings are accepted by the machine M?

a. 1
b. 01
c. 101000
d. 11
e. 0101010101
f. 10
g. 0101000000
h. 100
i. 0
j. 0100
k. 110000
l. 111010100000