Applied Human Language Technology

Lecture 2

Regular Expressions and Finite State Machines

Irene Murtagh

This week:

1. Regular Expressions and Finite State Machines
2. Generative Grammars and Formal Languages
3. Finite State Automata
   - Finite State Transducers
   - English Morphology

   (...i.e. Data structures for handling words of English)
4. Some things to do

The Chomsky Hierarchy of languages

Regular Expressions – What are they?

There are a couple of useful ways of viewing Regular Expressions. They can be described as:

- A language for specifying text search strings.
- As a useful way to specify textual search strings.
- As a way to specify the design of a particular kind of machine.

….As we will see, these are equivalent.

Formally: A RE is an algebraic notation for characterising a set of strings.

A **string** being any sequence of alphanumeric characters (letters, numbers, spaces, tabs and punctuation)

Regular Expressions: Text Searches

The simplest type of regular expression is a sequence of simple characters.

Simple Example

Suppose we wanted to write a regular expression to find cases of the English article *the*

/the/ >>problem: would miss The

/[tT]he/ >> Problem…might return the embedded o**the**r or **the**ology

/\b[tT]he\b/ >> Provides instances with a word boundary on each side

The Two Kinds of Errors

The process we just went through was based on fixing errors in the regular expression.

- Errors where some “the” instances were missed
  (judged to be not instances when they should have been).

- Errors where “the” instances were incl uded
  (when they should not have been).

This is pretty much going to be the story of the next few lectures....

Finite State Automata

REs can be viewed as a way to describe a particular kind of machine, which we'll call Finite State Automata .

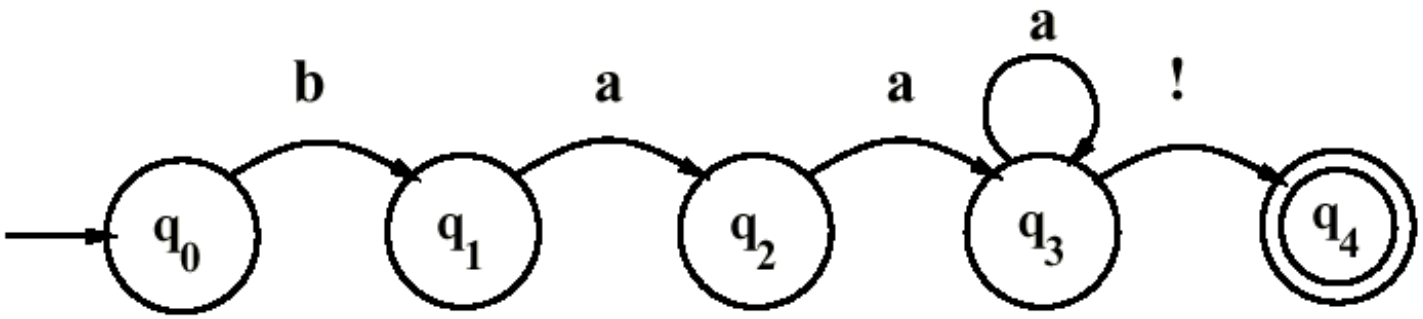
FSAs and their close variants are at the core of much of what we will be doing this semester.

Finite State Automata as Directed Graphs

We'll start with a graphical view of the sheep language

/baa+!/.

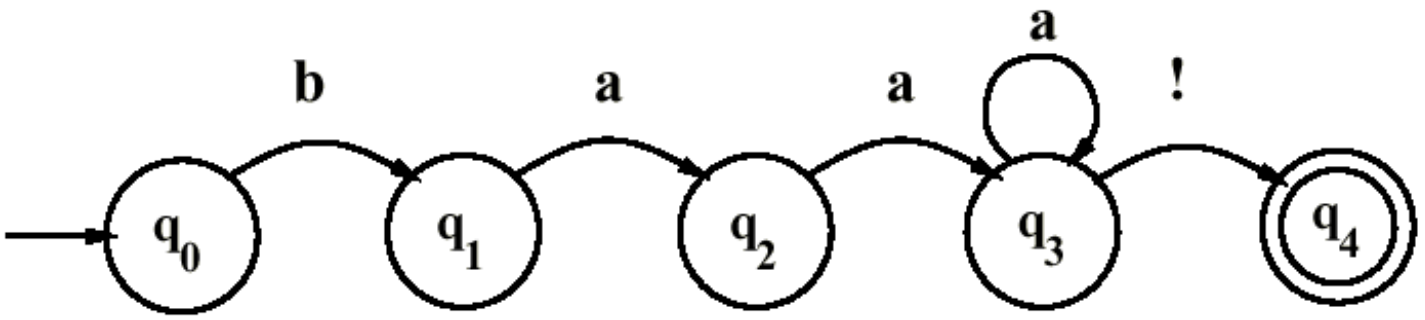
The FSA recognises a set of strings which characterise sheep talk). We represent a finite set of vertices called *nodes* with a set of directed links between pairs called *arcs*.



Finite State Automata as Directed Graphs

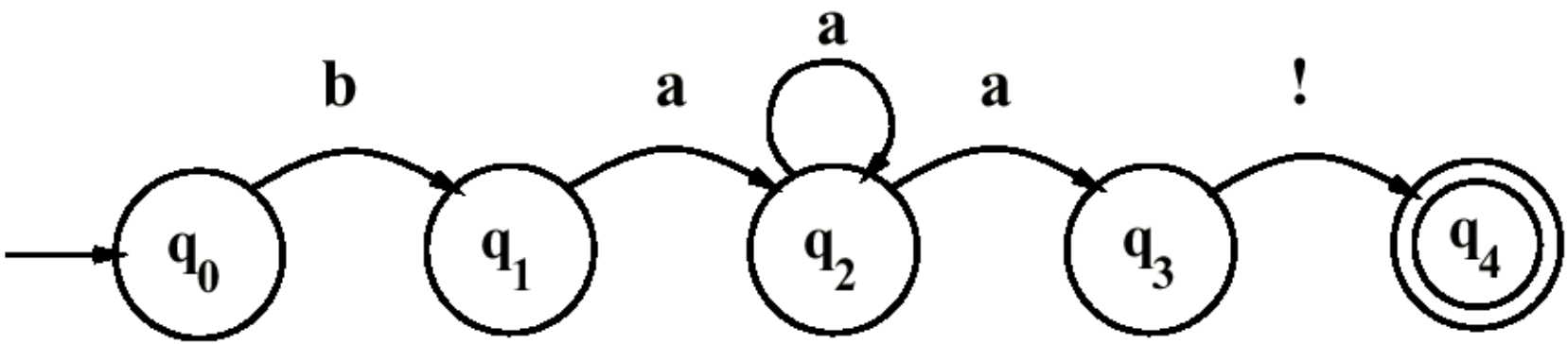
We can say the following things about this machine:

- It has five states $q_0 - q_4$
- q0 is the start state (represented by an incoming arrow)
- q4 is the final state (represented by a double circle)
- It has four transitions (represented by the arcs)



Sheep FSA Machine 2

Of course, there are always alternatives...



A More Formal View

- The set of states: Q
- A finite alphabet: Σ
- A start state
- A set of final states
- A transition function that maps Q × Σ to Q.

Letters		*Names*
A	α	alpha
B	β	beta
Γ	γ	gamma
Δ	δ	delta
E	ε	epsilon
Z	ζ	zeta
H	η	eta
Θ	θ	theta
I	ι	iota
K	κ	kappa
Λ	λ	lambda
M	μ	mu
N	ν	nu
Ξ	ξ	xi
O	o	omicron
Π	π	pi
P	ρ	rho
Σ	σ	sigma
T	τ	tau
Υ	υ	upsilon
Φ	ϕ	phi
X	χ	chi
Ψ	ψ	psi
Ω	ω	omega

The Letters of the Greek Alphabet

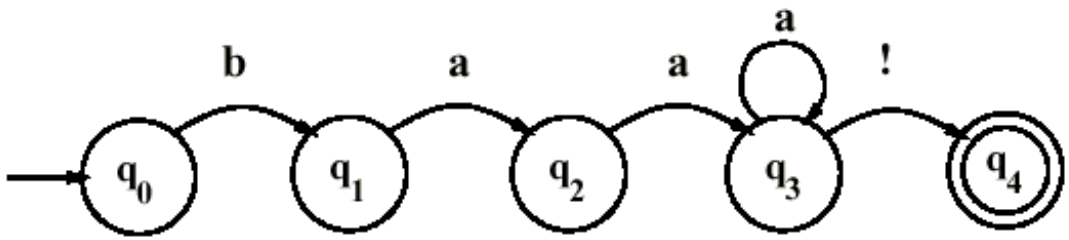
A Note on Alphabets with FSAs

- You shouldn't view the term alphabet too narrowly.
- In particular, you don't have to limit it to letters. Any kind of symbol will do.
- ... and critically, those symbols can stand for objects that themselves have internal str ucture.

Another View...

FSAs can also be encoded as state transition tables.

	Input		
State	b	a	!
0	1	∅	∅
1	∅	2	∅
2	∅	3	∅
3	∅	3	4
4:	∅	∅	∅



Another View...

FSAs can also be encoded as tables.

The first row will read like:

If we are in state 0 and we see the input B we must go to state 1

- NOTE: If we are in state 0 and we see the input **a** or **!** we fail.
- Ø indicates an illegal or missing transition
- The colon after 4 indicates that 4 is a final state
- You may have as many of the as you wish

FSAs can also be encoded as state transition tables.

The algorithm for representing the string using the previous state transition table is referred to as **Deterministic**.

A deterministic algorithm is one that has no choice points. The algorithm always knows what to do for any input.

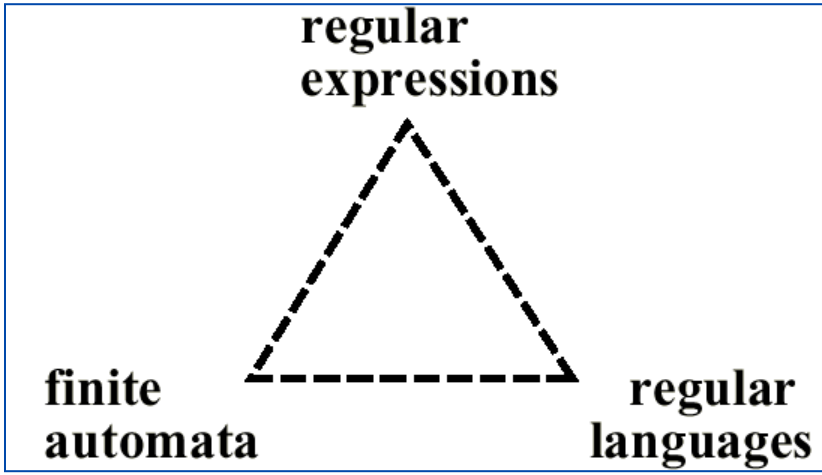
Generative Grammars and Formal Languages

A **Formal Language** is a set of strings composed of symbols from a finite set of symbols.

- FSAs (and regular expressions) 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.
- This dual view will pop up again and again.
- To generate, you traverse the machine and write transition symbols on the tape rather than reading them.

A Number of Views...

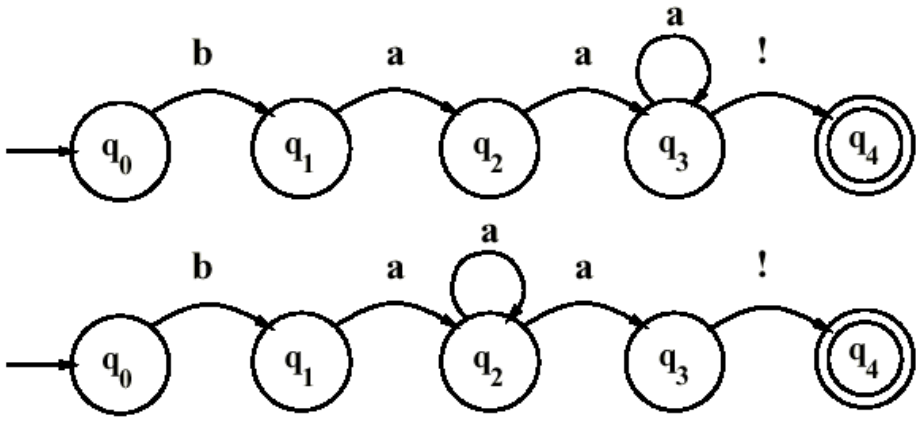
There are three formally equivalent ways of (not including tables) looking at what we're doing. Both REs and FSAs can be used to describe regular languages



Actually, ... there are four ways...
We'll talk later about representing regular languages with rules like these...

S → b a aA
A → a A
A → !

Non-Determinism in FSAs

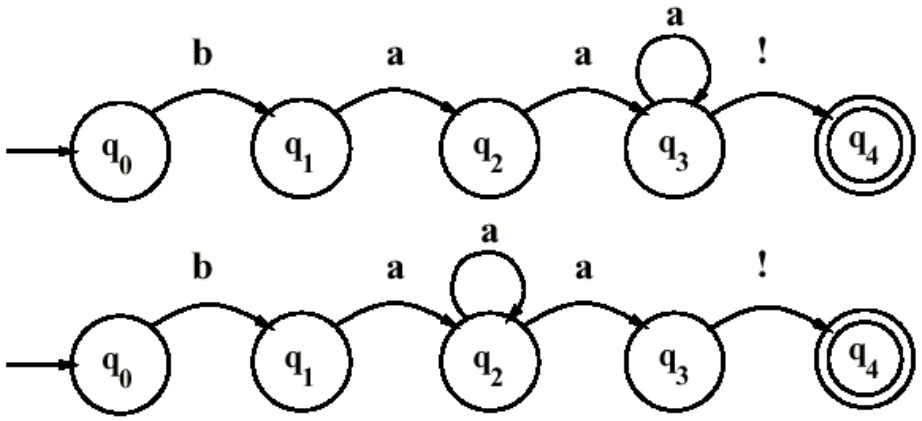


A way of introducing non-determinism is to use a self loop at state 2 instead of state 3.

When we get to state 2 – if we see an a we don't know to remain at 2 or move to state 3.

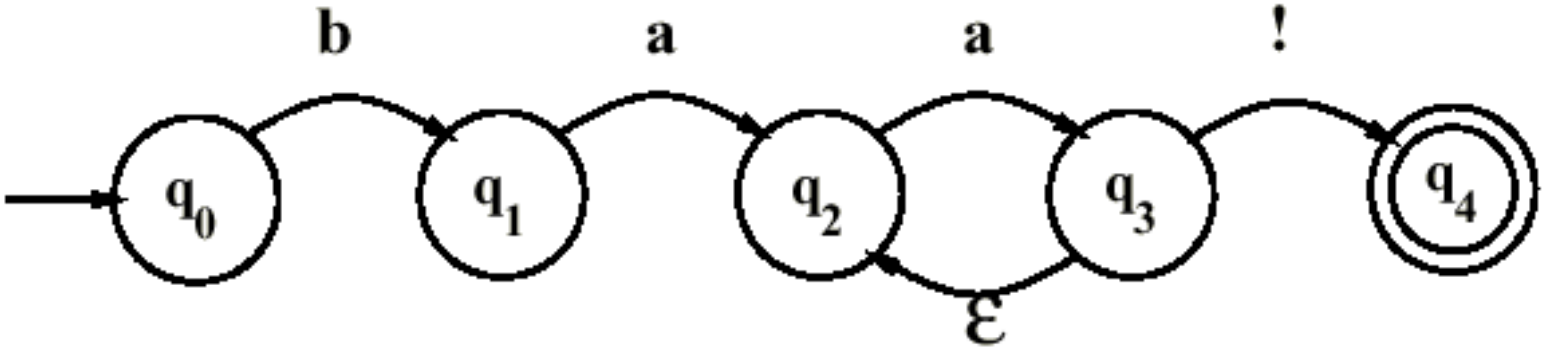
This is a non-deterministic FSA or NFSA

Non-Determinism in FSAs



Anothe way of introducing non-determinism is to use ε-transitions.

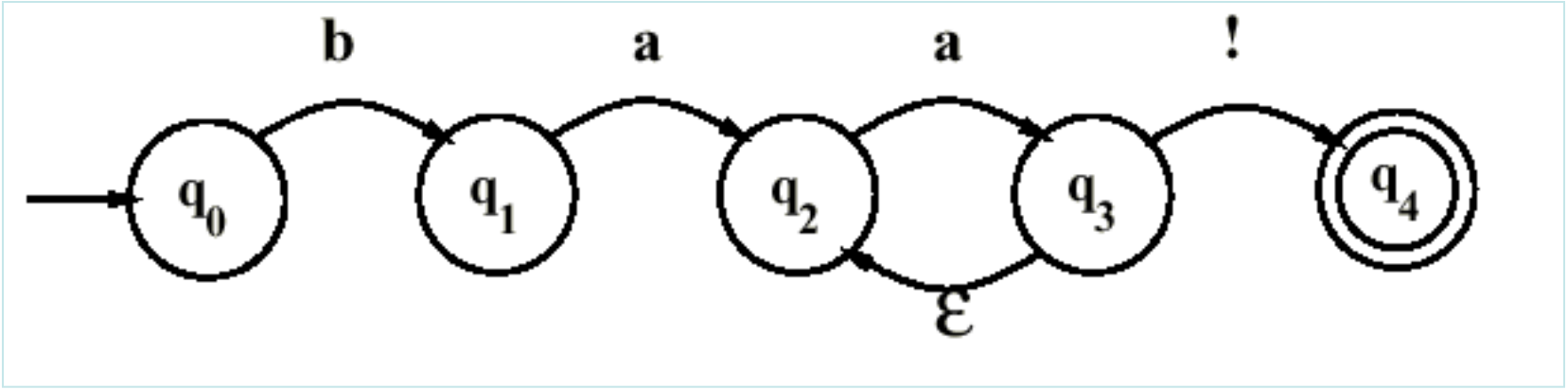
Key Point: ε-transitions do not examine/advance the tape.



Non-Determinism in FSAs

We interpret this as: if we are in state 3, we are allowed to move to state 2 without looking at the input, or advancing our input pointer.
This introduces another type of non-determinism …we might not know whether to follow the e transition or the **!**

We may make the wrong choice



Fun With Automata

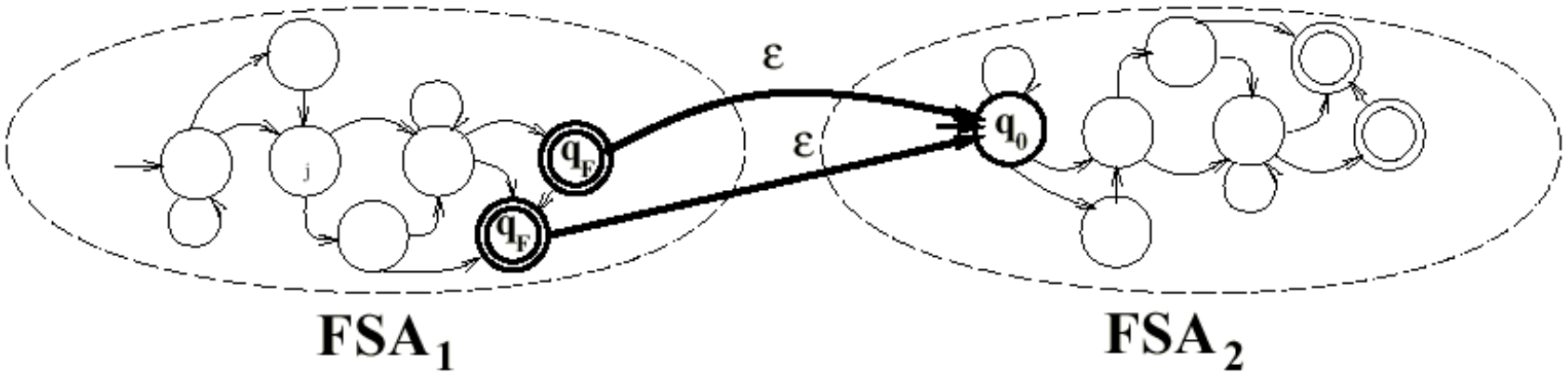
It turns out to be interesting (and useful) to consider the implications of various operations that combine regular languages into new languages.

In particular, we may want to know if the resulting combined language is still regular.

Why?

Concatenation

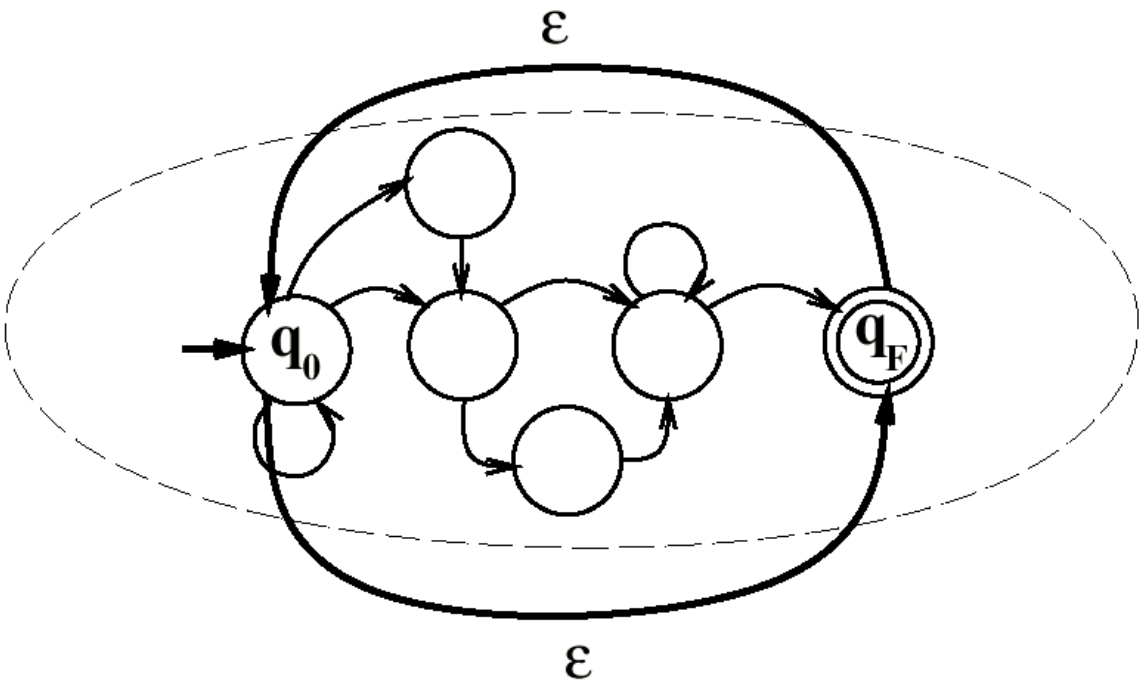
$L_1 \cdot L_2$



We string two FSAs next to each other by connecting all the **final** states of FSA1 to the **initial** state of FSA2 by an e-transition

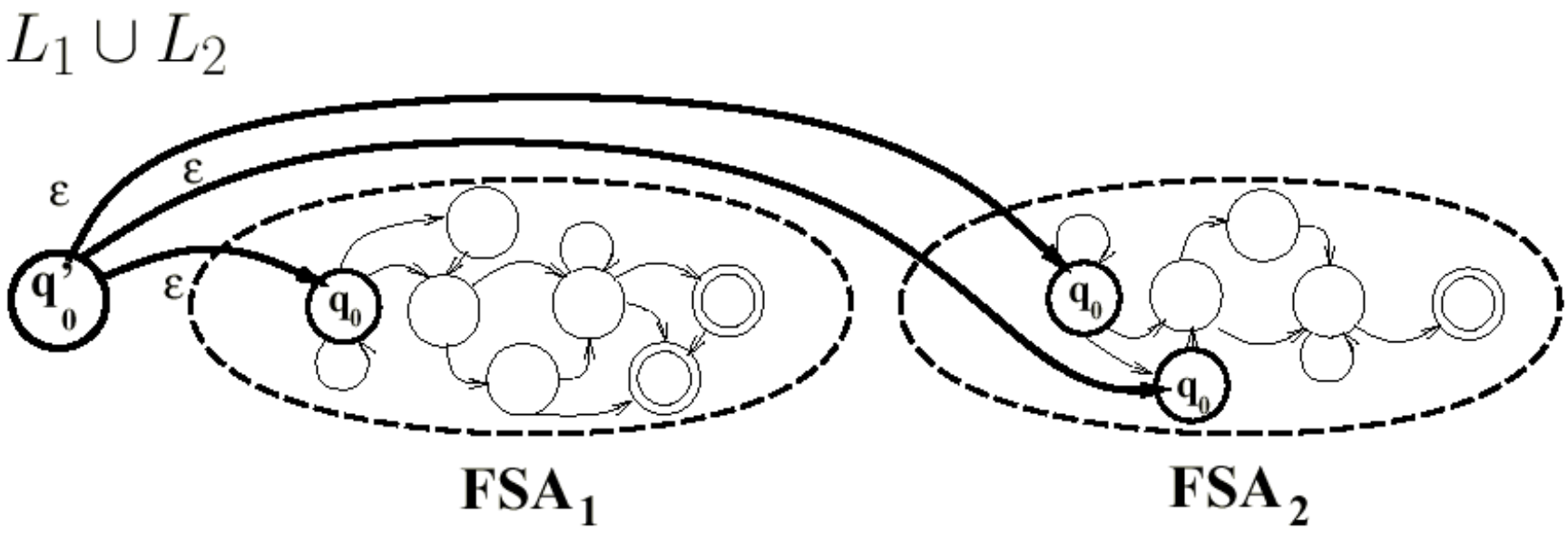
Kleene * (zero or more occurrences of the previous char or expression)

L_1*



Closure: We connect all the **final** states of the FSA back to the **initial** states by e-transitions (this implements part of the Kleene*), and then put direct links between the **initial** and **final** states by e-transitions (this implements the possibilities of having two zero occurrences

Union



We **add** a single **new initial state q'0**, and **add new transitions** from it to all of the former initial states of the two machines to be joined

Intersection

What about L1 ∩ L2
This turns out to be a tricky (and useful one).

If L1 and L2 are regular languages,
then .... so is L1 ∩ L2, the language consisting of the set of strings that are in both L1 and L2

Let us go back and
look at human language for a little while....

We will need to relate algorithms and structures in computing
with algorithms and structures human language

The generative approach to language processing

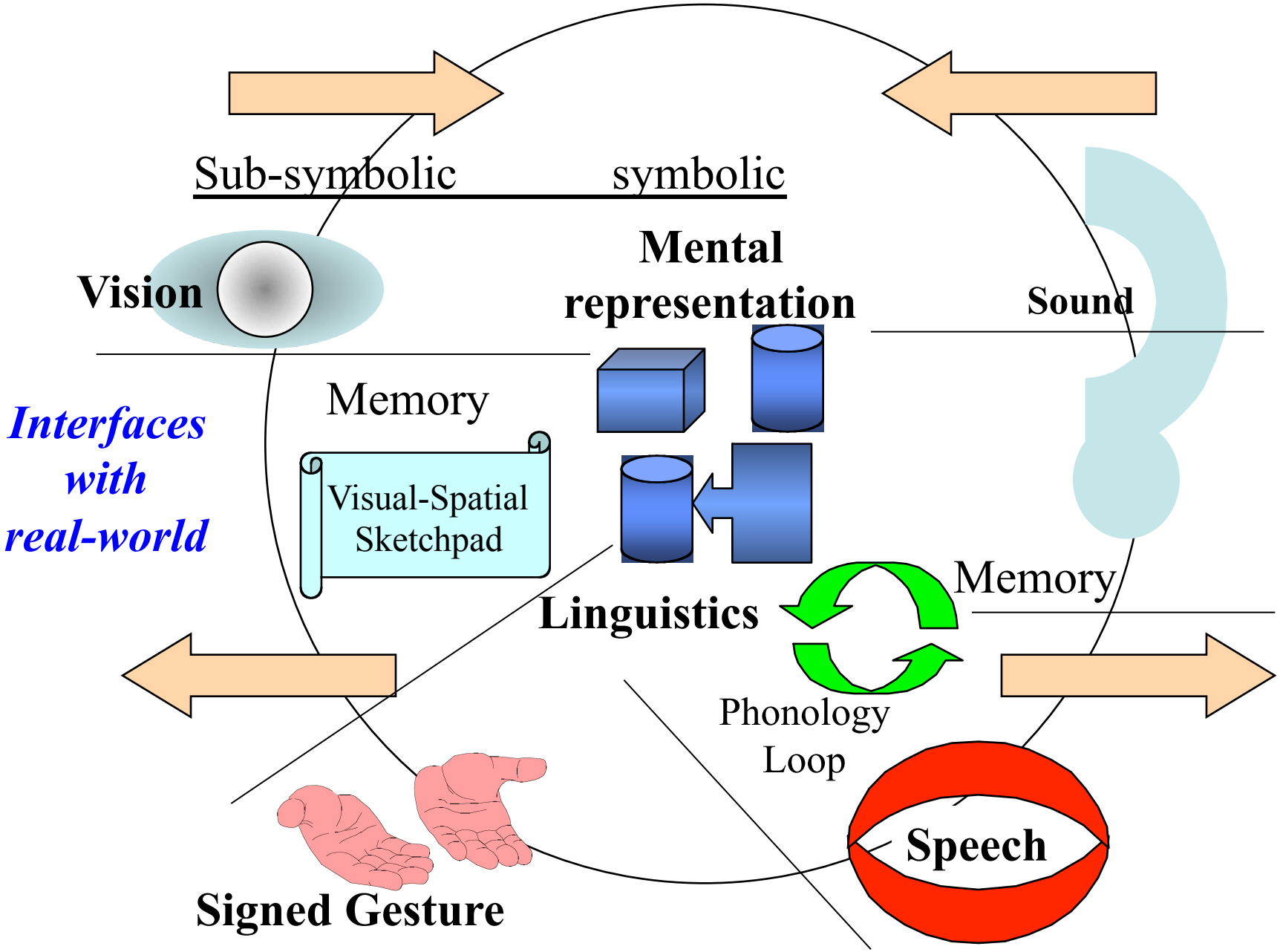
- The object of study
- The innateness hypothesis
- Competence and Performance
- Evidence: Creativity, grammaticality and acceptability

Computer science meets Linguistics and Language Engineering

Generative Grammar

Q:
What are we studying when we study grammar and syntax?

We are trying to investigate knowledge ……..human knowledge


Sub-symbolic
symbolic
Vision
Mental representation
Sound
Memory
Visual-Spatial Sketchpad
Interfaces with real-world
Linguistics
Memory
Phonology Loop
Speech
Signed Gesture

(2) **Linguistics - the scientific study of language**

"Linguistics may be defined as the scientific study of language. For the moment, it will be enough to say that by the scientific study of language is meant its investigation by means of controlled and emp irically verifiable observations and with reference to some general theory of language structure" (Lyons 1968: 1)

Competence vs. Perfor mance

- Abstract vs. concrete use
- Social vs. individual use
- Langue vs. parole

Competence vs. performance

"We thus make a fundamental distinction between **competence** (the speaker -hearer's knowledge of his language) and **performance** (the actual use of language in concrete situations)" (Chomsky 1965: 4)

(8) **A grammar of a language**

"A grammar of a language purports to be a description of the ideal speaker -hearer's competence " (Chomsky 1965: 4)

(9) **The human language faculty**

"The human mind/brain is a complex system with various interacting components, one of which we may call the language faculty . This system appears to be unique in essentials to the human species and common to members of the species. Presented with data, the language faculty determines a particular language Spanish, English, etc." (Chomsky 1988: 35)

Syntax

Creativity, Grammaticality and Acceptability

By the syntax of a language, we mean the body of rules that speakers of the language follow when they combine words into sentences.

At first glance, it may not be obvious just how much there is to be said abo ut English rules.

(1) Martha lives in the house that John sold to her

A speaker of this sentence would not necessarily be under the impression that they were following rules, but simply letting the thought to be expressed dictate the choice of words and their arrangement.

Compare this with:

(2) Martha John her to sold house in lives

Which of the sentences below are more correct?

Which sound distinctly odd, and why?

Even though some sentences are odd, we still understand what the sentences are supposed to communicate.

(3) a. The mayor gave John some good advices.
b. The mayor gave John some good advice.

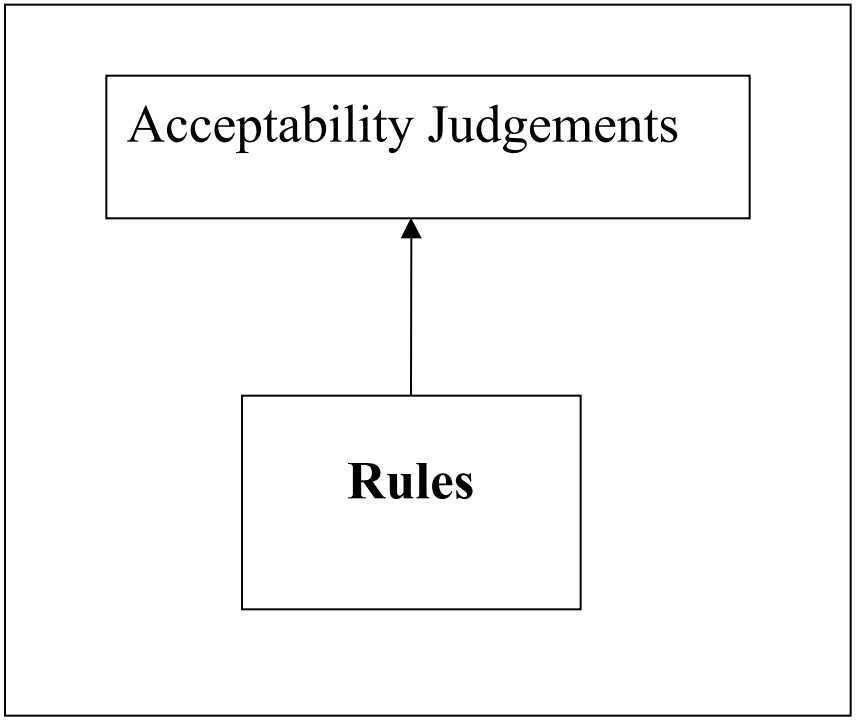
(4) a. This man going to the station.
b. This man is goin g to the station.

(5) a. Jack read the book that Marsha bought it for him.
b. Jack read the book that Marsha bought for him.

(6) a. Anyone didn't see the accident.
b. No one saw the accident.

These provide some evidence that speakers of the langua ge follow a set of syntactic rules.

(7)



The acceptability **judgements** that we made are accessible to our introspection in that we can observe our differing reactions to the sentences in the above pairs.

(8) a. The mayor gave John some good advices.
b. The mayor gave John some good advice.

(9) a. This man going to the station.
b. This man is going to the station.

(10) a. Jack read the book that Marsha bought it for him.
b. Jack read the book tha t Marsha bought for him.

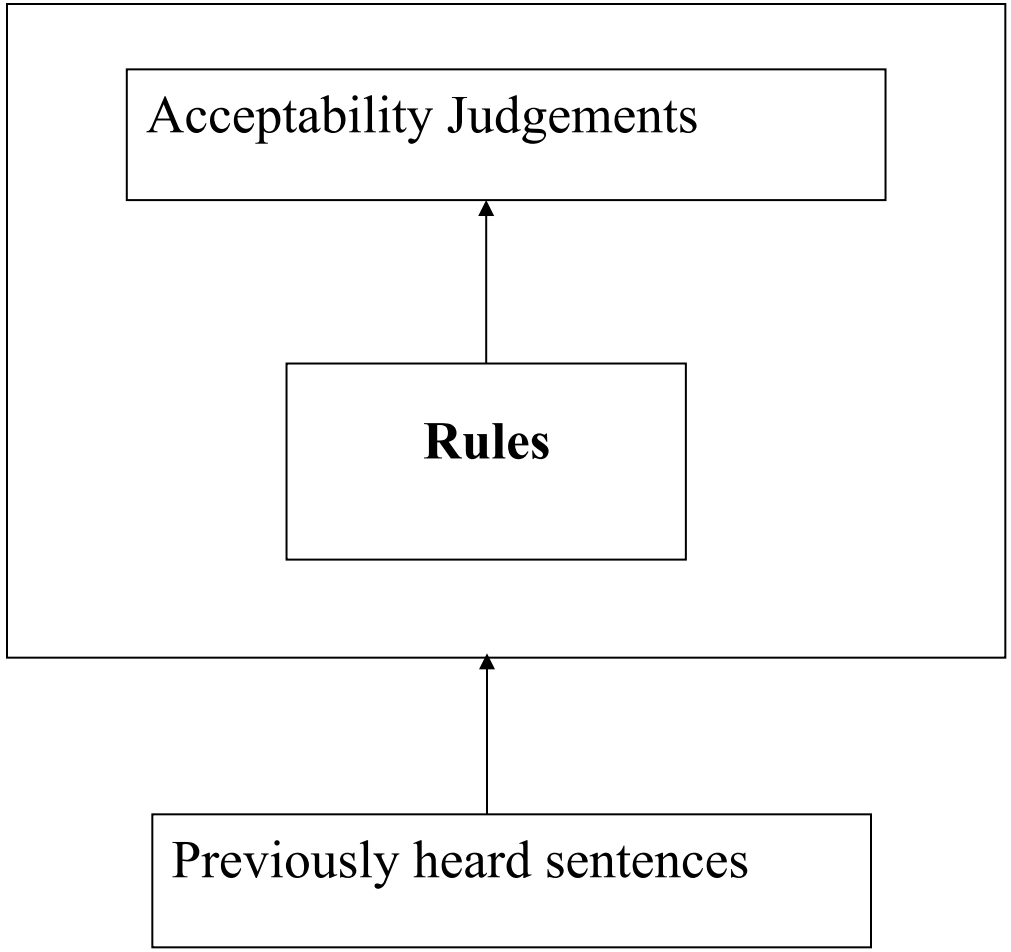
(11) a. Anyone didn't see the accident.
b. No one saw the accident.

By contrast, the **rules** that determine these judgements are inaccessible to our introspection. We cannot look into our minds and see what these rules are.

"… I don't really think that I followed any rules when I picked the good b. sentences earlier. They w ere just similar to good sentences that I heard before ".

The sentences heard in childhood determine our mental rules , and these rules in turn determine the sentences that we accept in adulthood.

(12)



Syntax → The investigation of systems of knowledge in minds of native speakers.

How do we discover the rules?

- We cannot look into our minds and observe them directly.
- We can, however, look at our acceptability judgements and make observations as to what the rules might be.

To build formal models of a particular area of knowledge, we need to know:

- A means of Knowledge Representation
- How to test the model

Linguistics is empirical study

→ Therefore, can be tested in the field for acceptability by native speakers

- What we are after cannot be touched as it is deep in someone's mind.
- It can only be measured indirectly!

Computer based models can test our assumptions

Creativity of Speaker

- No corpus of data of language was sufficient to form a grammar of a language.

- Speakers can always produce new sentences.

- The number of possible sentences is infinite.

  BUT….

  The knowledge necessary to produce these sentences need n ot be so large

- Data not as interesting as the process

Grammatically

- A sentence is grammatical if it follows the rules of the language.

- There may be a gap between acceptability and grammaticality

An example of this is Garden Path Sentences

Garden Path Sentences

- Structure misleads the reader, but sentences are grammatical
- Intonation helps meaning

(13) a. The horse raced past the barn fell.

versus.

(14) a. The horse that was walked past the fence proceeded steadily,
but the horse raced past the barn fell.

(Pinker: 1994: 212ff)

TO DO THIS WEEK

Read: Ch.2 & 3: *Speech and Language Processing* by Jurafsky & Martin

Then Do exercise 3.1 and exercise 3.2 in J+M Ch3.

AND …..Add 6 extra elements to each

Define and Describe using an example what a
Kleene* and a Kleene + is and how they are used.