Linguistics

- Phonology – sound
- Morphology – word composition
- Syntax – structure
- Semantics – meaning
- Pragmatics – implications

Syntax

- Grammar – discrete combinatorial system
  - Finite number of elements sampled & combined to create larger structures
    - Words → sentences

Traditionally

coronations
engagements
funerals → See folder 32!
weddings → See folder 33!
comings-of-age
births
deaths

are occasions for mourning #32

are occasions for rejoicing #33

The wedding of X and Y

is no exception.
is an exception.

Finite State Model (aka Markov Model)

Finite State Model (Informal Definition)

- Lists of words
- Directions for going from list to list

Fun with Finite State Models

The				boy				ice cream
A	→	happy	→	girl	→	eats	→	hot dogs
One				dog				candy

S: A B C D E
A: {the, a, one}
B: {happy}
C: {boy, girl, dog}
D: {eats}
E: {ice cream, hot dogs, candy}

Perspectives on Language

- Encoding Perspective

Concept → Encoder/Grammar → Encoded Concept

- Language characterized by grammatical rules that dictate form encodings follow

Perspectives

- Decoding Perspective
- Language characterized by computational complexity of algorithm or computer required to decode concepts

Encoding Perspective ⟺ Decoding Perspective
Grammatical Viewpoint ⟺ Computability Viewpoint

Rewriting Systems

$\phi \rightarrow \psi$

- Terminal Vocabulary
  - Finite set of words that compose strings in language e.g. "dog"
- Nonterminal Vocabulary
  - Symbols used in rules e.g. Noun

Definitions

- Derived – one string obtained from another via application of a finite sequence of rules
- Generated – derivable from start symbol
- Language – set of all strings generated by a grammar

A Simple Grammar

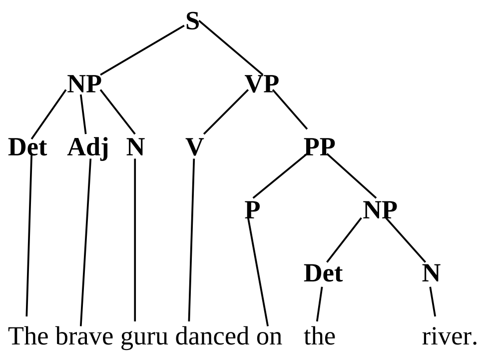
S → NP VP
NP → (Det)(Adj) N
NP → Pronoun
VP → V NP
VP → V PP
PP → P NP
V → Aux V

V → {hit, saved, cooking, danced}
N → {dog, child, guru, girl,apples, river}
Det → {the, a}
Adj → {brave, drowning, cooking}
Pronoun → {he, she, they}
P → {on, by}
Aux → {was,were}

The brave guru danced on the river.

S → NP VP
NP VP
NP → (Det) (Adj) N
Det Adj N VP
VP → V PP
Det Adj N V PP
PP → P NP
Det Adj N V P NP

NP → (Det) (Adj) N
Det Adj N V P Det N
The brave guru danced on the river.



Computability Viewpoint

- Case 1: string S is in L and encodes concept C

String S → Decoder/Algorithm → Concept C (ACCEPTS S)

- Case 2: String S is not in L

String S → Decoder/Algorithm → REJECTION

- Machine recognizes L if it both accepts all legal strings of L and rejects strings not in L

Automaton

Hypothetical Machine w/

- Input tape
  - Finite length w/string of symbols printed on it
- Reading head
  - Scans input tape one symbol at a time
- Finite set of internal states
- Internal memory structure
  - Differs in different types of automata
- Finite set of instructions
  - Function of input, state, memory

Perspectives

- Build an automaton that can decide whether or not any given string s belongs in L
- Unnatural but not insane
  - Recognition Problem → Decoding Problem
  - You can't decode if you can't recognize
- Grammars can be transformed into machines that recognize strings in their language

Types of Grammars

- Type 0 Grammars
  - No restrictions on rules: rules may be recursive, and any number of symbols may occur on either side of a rule
- Type 1 Grammars *Context-Sensitive Grammars*
  - Grammars in which every rule is of the form

  $\sigma A \tau \rightarrow \sigma \phi \tau$
  - Where A is nonterminal and σ and τ are arbitrary strings of terminals and nonterminals, with ϕ nonempty

Types of Grammars

- Type 2 *Context-Free Grammars*
  - Grammars in which every rule is of the form

  $A \rightarrow \phi$
  - Where A is a nonterminal and ϕ is an arbitrary nonempty string of terminals and nonterminals
- Type 3 *Finite State Grammars*
  - Grammars in which every rule is of the form

  $A \rightarrow xB$ or $A \rightarrow x$
  - A and B are single nonterminals
  - x is an arbitrary string of terminals

Intuitions

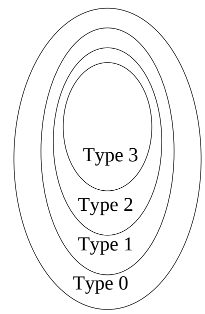
- Type 1 and Type 2 Grammars
  - Sentences made up of phrases
  - Phrases made up of smaller phrases
- Type 2
  - A   Prep Phrase
  - □ ϕ   in the doghouse
- Type 1
  - Certain types of phrases differ in different grammatical environments

  NP VP → N Det VP

  V NP → V Det N
- Type 3
  - Generate sentences left to right

Relationships between Languages

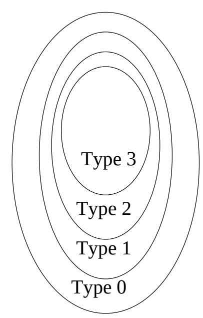
- Type 0 grammars with rules of equivalent length on the left & right sides generate all the Type 1 languages
  - Type 1 languages a subset of Type 0 languages
- Type 1 grammars in which σ and τ are always empty generate all the Type 2 languages
  - Context-Sensitive versus Context-Free
  - Type 2 subset Type 1



Automata

- Turing Machine
  - Infinite tape!
- Linear Bounded Automata
  - Available memory is a linear function of length of input
- Pushdown Automata
  - Stack memory with last in first out pattern
- Finite State Automata
  - No internal memory

Chomsky Hierarchy



- Type 0
  - Recursively Enumerable Grammar
  - Turing Machine
- Type 1
  - Context-Sensitive Grammar
  - Linear Bounded Automata
- Type 2
  - Context Free Grammar
  - Pushdown Automata
- Type 3
  - Finite State Grammar
  - Finite State Automata

What can a finite state automaton do?

L1: a^nb^n n>=1   Can FSA handle this?

ab

aabb

aaabbb

*aab

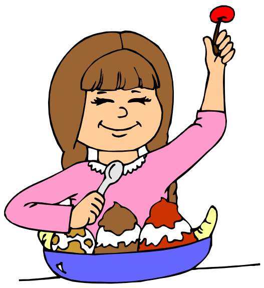
*abbb

How to generate a^nb^n

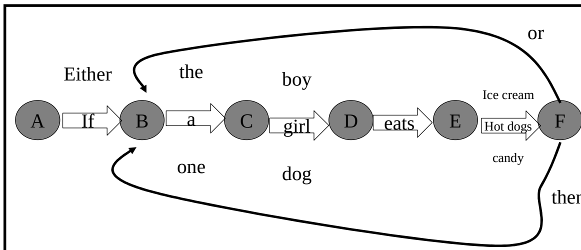
- S → aSb
- S → ab

aSb	aSb
aabb	aaSbb
	aaaSbbb
	etc.

Ice Cream Sentences



- Either the girl eats ice cream, or the girl eats candy.
- If the girl eats ice cream, then the boy eats hot dogs.



A→	Either B	B→	the C	C→	boy D
	If B		a C		girl D
			one C		dog D
D → eats E		E →	ice cream F	F →	or B
			hot dogs F		then B
			candy F		<stop>