Artificial Intelligence

Subject Code: 20A05502T

UNIT IV - Natural Language for Communication

PHRASE STRUCTURE GRAMMARS

- Generative Capacity
  - Recursively Enumerable Grammars
  - Context-sensitive Grammars
  - Context-free Grammars
  - Regular Grammar
  - Probabilistic Context-free Grammar
- The lexicon of E_0
- The Grammar of E_0
- Parse Tree for a sentence
- Over-generation & Under-generation of Grammar

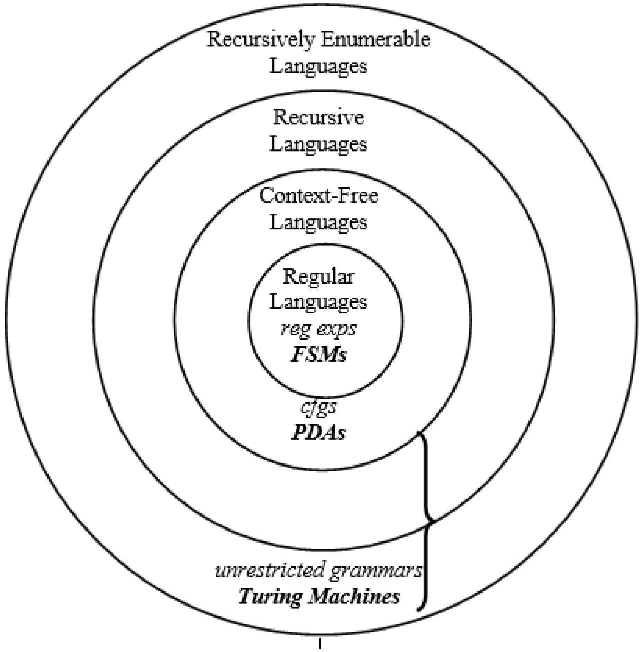
$\mathcal{E}_0$:	S	→	$NP\ VP$	[0.90]	I + feel a breeze
		|	$S\ Conj\ S$	[0.10]	I feel a breeze + and + It stinks
	NP	→	$Pronoun$	[0.30]	I
		|	$Name$	[0.10]	John
		|	$Noun$	[0.10]	pits
		|	$Article\ Noun$	[0.25]	the + wumpus
		|	$Article\ Adjs\ Noun$	[0.05]	the + smelly dead + wumpus
		|	$Digit\ Digit$	[0.05]	3 4
		|	$NP\ PP$	[0.10]	the wumpus + in 1 3
		|	$NP\ RelClause$	[0.05]	the wumpus + that is smelly
	VP	→	$Verb$	[0.40]	stinks
		|	$VP\ NP$	[0.35]	feel + a breeze
		|	$VP\ Adjective$	[0.05]	smells + dead
		|	$VP\ PP$	[0.10]	is + in 1 3
		|	$VP\ Adverb$	[0.10]	go + ahead
	$Adjs$	→	$Adjective$	[0.80]	smelly
		|	$Adjective\ Adjs$	[0.20]	smelly + dead
	PP	→	$Prep\ NP$	[1.00]	to + the east
	$RelClause$	→	$RelPro\ VP$	[1.00]	that + is smelly

PHRASE STRUCTURE GRAMMARS

- A **grammar** is a collection of rules that defines a **language** as a set of allowable strings of words.
- Rules for allowable characters, words and sentences.
- The notion of a **lexical category** (also known as a **part of speech**) such as *noun* or *adjective* is a useful generalization
- **syntactic categories :** string together lexical categories such as *noun phrase* or *verb phrase*, and
- **phrase structure:** combine these syntactic categories into trees representing the **phrase structure** of sentences:
- nested phrases, each marked with a category.

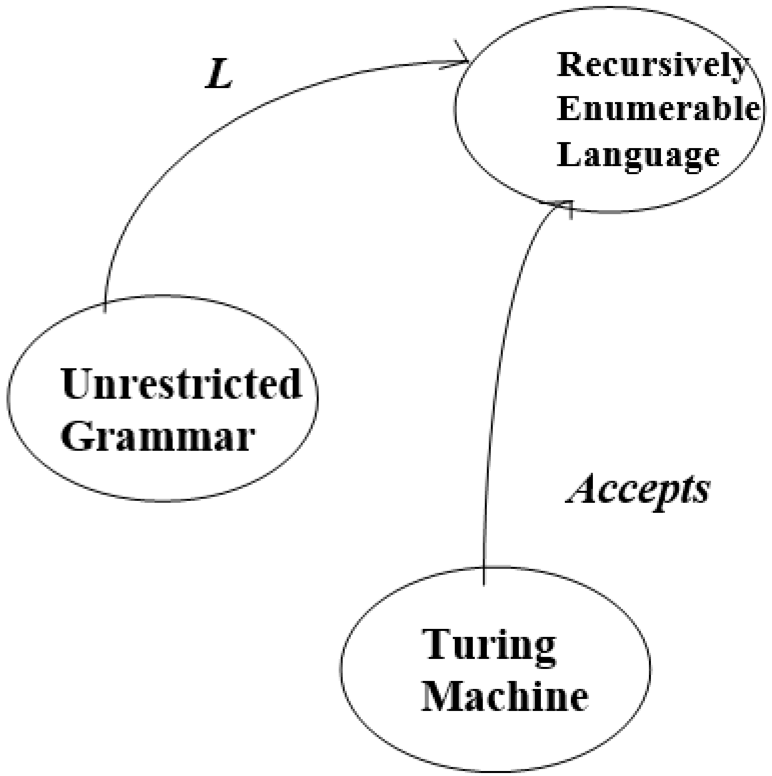
Generative Capacity

- Grammatical formalisms can be classified by their **generative capacity**: the set of languages they can represent.
- Four classes of grammatical formalisms that differ only in the form of the rewrite rules.
- Here we list the hierarchy, most powerful class first:
- Recursively enumerable
- Context-sensitive grammars
- Context-free grammars (or CFGs),
- Regular grammars (Finite State Machine FSM)



Recursively Enumerable Grammars

- **Recursively Enumerable** grammars use unrestricted rules:
- both sides of the rewrite rules can have any number of terminal and nonterminal symbols, as in the rule
- A B C → D E.
- These grammars are equivalent to Turing machines in their expressive power.



Context-sensitive Grammars

- **Context-sensitive grammars** are restricted only in that the right-hand side must contain at least as many symbols as the left-hand side.
- The name “context-sensitive” comes from the fact that a rule such as
- A X B → A Y B
- an X can be rewritten as a Y in the context of a
- preceding A and a following B.
- Context-sensitive grammars can represent languages such as
- $a^n b^n c^n$
- (a sequence of n copies of a followed by the same number of bs and then cs).
- E.g. aaabbbccc

Context-free Grammars

- In **context-free grammars** (or **CFG**s), the left-hand side consists of a single nonterminal symbol.
- Thus, each rule licenses rewriting the nonterminal as the right-hand side in *any* context.
- CFGs are popular for natural-language and programming-language grammars
- it is now widely accepted
- Context-free grammars can represent
- a^nb^n,
- but not $a^nb^nc^n$.

Regular Grammar

- **Regular** grammars are the most restricted class.
- Every rule has a single nonterminal on the left-hand side and a terminal symbol optionally followed by a nonterminal on the right-hand side.
- Regular grammars are equivalent in power to Finite State Machines (FSM).
- They are poorly suited for programming languages, because they cannot represent constructs such as balanced opening and closing parentheses
- (a variation of the $a^n b^n$ language).
- The closest they can come is representing $a * b *$, a sequence of any number of 'a's followed by any number of 'b's.
- aaaabbbbbb

Probabilistic Context-free Grammar (PCFG)

- This language model based on the idea of phrase structure;
- "Context-free" - the left-hand side consists of a single nonterminal symbol.
- "probabilistic" means that the grammar assigns a probability to every string.
- Grammar refers Non terminal symbols and Terminal symbols.
- PCFG rule:
- VP → Verb [0.70] | VP NP [0.30] .
- Here VP (*verb phrase*) and NP (*noun phrase*) are **non-terminal symbols**.
- actual words are called **terminal symbols**.
- This rule is saying that with probability 0.70 a verb phrase consists of a verb, and with probability 0.30 it is a VP followed by an NP.

The lexicon of E_0

- The **lexicon**, or list of allowable words.
- The words are grouped into the lexical categories familiar to dictionary users:
- nouns, pronouns, and names to denote things;
- verbs to denote events;
- adjectives to modify nouns;
- adverbs to modify verbs; and
- function words:
- articles (such as *the*),
- prepositions (*in*), and
- conjunctions (*and*).

Example

Noun	→	**stench** [0.05] | **breeze** [0.10] | **wumpus** [0.15] | **pits** [0.05] | ...
Verb	→	**is** [0.10] | **feel** [0.10] | **smells** [0.10] | **stinks** [0.05] | ...
Adjective	→	**right** [0.10] | **dead** [0.05] | **smelly** [0.02] | **breezy** [0.02] ...
Adverb	→	**here** [0.05] | **ahead** [0.05] | **nearby** [0.02] | ...
Pronoun	→	**me** [0.10] | **you** [0.03] | **I** [0.10] | **it** [0.10] | ...
RelPro	→	**that** [0.40] | **which** [0.15] | **who** [0.20] | **whom** [0.02] ∨ ...
Name	→	**John** [0.01] | **Mary** [0.01] | **Boston** [0.01] | ...
Article	→	**the** [0.40] | **a** [0.30] | **an** [0.10] | **every** [0.05] | ...
Prep	→	**to** [0.20] | **in** [0.10] | **on** [0.05] | **near** [0.10] | ...
Conj	→	**and** [0.50] | **or** [0.10] | **but** [0.20] | **yet** [0.02] ∨ ...
Digit	→	**0** [0.20] | **1** [0.20] | **2** [0.20] | **3** [0.20] | **4** [0.20] | ...

- Each of the categories ends in . . . to indicate that there are other words in the category.
- For nouns, names, verbs, adjectives, and adverbs, it is infeasible even in principle to list all the words.
- Not only are there tens of thousands of members in each class, but new ones– like *iPod* or *biodiesel*—are being added constantly.
- These five categories are called **open classes**.
- For the categories of pronoun, relative pronoun, article, preposition, and conjunction.
- These are called **closed classes**; they have a small number of words (a dozen or so).
- Closed classes change over the course of centuries, not months.
- For example, “thee” and “thou” were commonly used pronouns in the 17th century, were on the decline in the 19th, and are seen today only in poetry and some regional dialects.

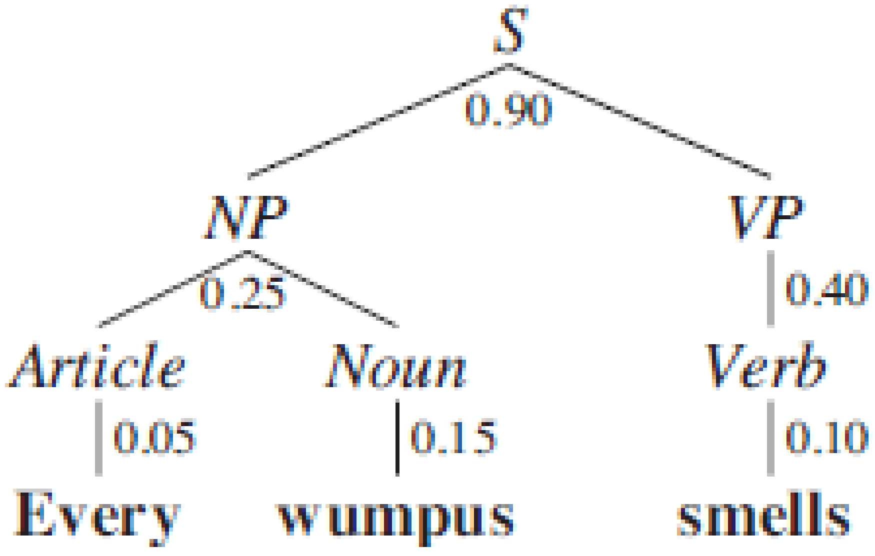
The Grammar of E_0

- to combine the words into phrases Grammar is required.
- A grammar for E_0,
- with rules for each of the six syntactic categories

$\mathcal{E}_0$:	S	$\rightarrow$	$NP\ VP$	[0.90]	I + feel a breeze
		$\mid$	$S\ Conj\ S$	[0.10]	I feel a breeze + and + It stinks
	NP	$\rightarrow$	$Pronoun$	[0.30]	I
		$\mid$	$Name$	[0.10]	John
		$\mid$	$Noun$	[0.10]	pits
		$\mid$	$Article\ Noun$	[0.25]	the + wumpus
		$\mid$	$Article\ Adjs\ Noun$	[0.05]	the + smelly dead + wumpus
		$\mid$	$Digit\ Digit$	[0.05]	3 4
		$\mid$	$NP\ PP$	[0.10]	the wumpus + in 1 3
		$\mid$	$NP\ RelClause$	[0.05]	the wumpus + that is smelly
	VP	$\rightarrow$	$Verb$	[0.40]	stinks
		$\mid$	$VP\ NP$	[0.35]	feel + a breeze
		$\mid$	$VP\ Adjective$	[0.05]	smells + dead
		$\mid$	$VP\ PP$	[0.10]	is + in 1 3
		$\mid$	$VP\ Adverb$	[0.10]	go + ahead
	$Adjs$	$\rightarrow$	$Adjective$	[0.80]	smelly
		$\mid$	$Adjective\ Adjs$	[0.20]	smelly + dead
	PP	$\rightarrow$	$Prep\ NP$	[1.00]	to + the east
	$RelClause$	$\rightarrow$	$RelPro\ VP$	[1.00]	that + is smelly

Parse Tree for the sentence “Every wumpus smells”

- The parse tree gives a constructive proof that the string of words is indeed a sentence according to the rules of E_0.
- Each interior node of the tree is labeled with its probability.
- The probability of the tree as a whole is $0.9 \times 0.25 \times 0.05 \times 0.15 \times 0.40 \times 0.10 = 0.0000675$.
- Since this tree is the only parse of the sentence, that number is also the probability of the sentence.
- The tree can also be written in linear form as
- [S [NP [Article **every**] [Noun **wumpus**]][VP [Verb **smells**]]].



- The E0 grammar generates a wide range of English sentences such as the following:
- John is in the pit
- The wumpus that stinks is in 2 2
- Mary is in Boston and the wumpus is near 3 2

Over-generation and Under-generation of Grammar

- the grammar **overgenerates**:
- it generates sentences that are not grammatical, such as
- “Me go Boston” and “I smell pits wumpus John.”
- It also **undergenerates**:
- there are many sentences of English that it rejects, such as
- “I think the wumpus is smelly.”
- We will see how to learn a better grammar later; for now we concentrate on what we can do
- with the grammar we have.

Thank You

- **UNIT IV - Natural Language for Communication**
- PHRASE STRUCTURE GRAMMARS
- Generative Capacity
  - Recursively Enumerable Grammars
  - Context-sensitive Grammars
  - Context-free Grammars
  - Regular Grammar
  - Probabilistic Context-free Grammar
- The lexicon of E_0
- The Grammar of E_0
- Parse Tree for a sentence
- Over-generation & Under-generation of Grammar