BU CS 332 – Theory of Computation

Lecture 7:

- Distinguishing sets, non-regularity
- Context-free grammars

Reading:
"Myhill-Nerode" note
Sipser Ch 1.4 (optional)
Sipser Ch 2.1

Mark Bun

February 16, 2021

Distinguishing Sets

Motivating Questions

- How can we tell if we've found the smallest DFA recognizing a language?

  Last time: Introduced distinguishing set method

- Are all languages regular? How can we prove that a language is not regular?

A General Technique

$A = \{ w \in \{0,1\}^* \mid w \text{ ends with } 01 \}$

Definition: Strings x and y are **distinguishable** by L if there exists z such that exactly one of xz or yz is in L.

Ex. $x = \varepsilon, \quad y = 0$

Definition: A set of strings S is **pairwise distinguishable** by L if every pair of distinct strings $x, y \in S$ is distinguishable by L.

Ex. $S = \{\varepsilon, 0, 01\}$

Theorem: If S is pairwise distinguishable by L, then every DFA recognizing L needs at least $|S|$ states

Another Example

$B = \{ w \in \{0,1\}^* \mid |w| = 2 \}$

Theorem: If S is pairwise distinguishable by L, then every DFA recognizing L needs at least $|S|$ states

$S = \{ \qquad \}$

Distinguishing Extension

Which of the following is a distinguishing extension for $x = 0$ and $y = 00$ for language $B = \{ w \in \{0, 1\}^* \mid |w| = 2 \}$?

a) $z = \varepsilon$
b) $z = 0$
c) $z = 1$
d) $z = 00$

Historical Note

Converse to the distinguishing set method:

If L has no distinguishing set of size $> k$, then L is recognized by a DFA with k states

Myhill-Nerode Theorem (1958): L is recognized by a DFA with $\leq k$ states iff L does not have a distinguishing set of size $> k$

Non-Regularity

Theorem: If S is pairwise distinguishable by L, then every DFA recognizing L needs at least $|S|$ states

Corollary: If S is an **infinite** set that is pairwise distinguishable by L, then no DFA recognizes L

The Classic Example

Theorem: $A = \{0^n 1^n \mid n \geq 0\}$ is not regular

Proof: Construct an infinite pairwise distinguishable set

Palindromes

Theorem: $L = \{w \in \{0,1\}^* \mid w = w^R\}$ is not regular

Proof: Construct an infinite pairwise distinguishable set

Now you try!

Use the distinguishing set method to show that the following languages are not regular

$L_1 = \{0^i 1^j \mid i > j \geq 0\}$

$L_2 = \{ww \mid w \in \{0,1\}^*\}$

$L_3 = \{1^{n^2} \mid n \geq 0\}$

Reusing a Proof

Finding a distinguishing set can take some work…

Let's try to reuse that work!

How might we show that

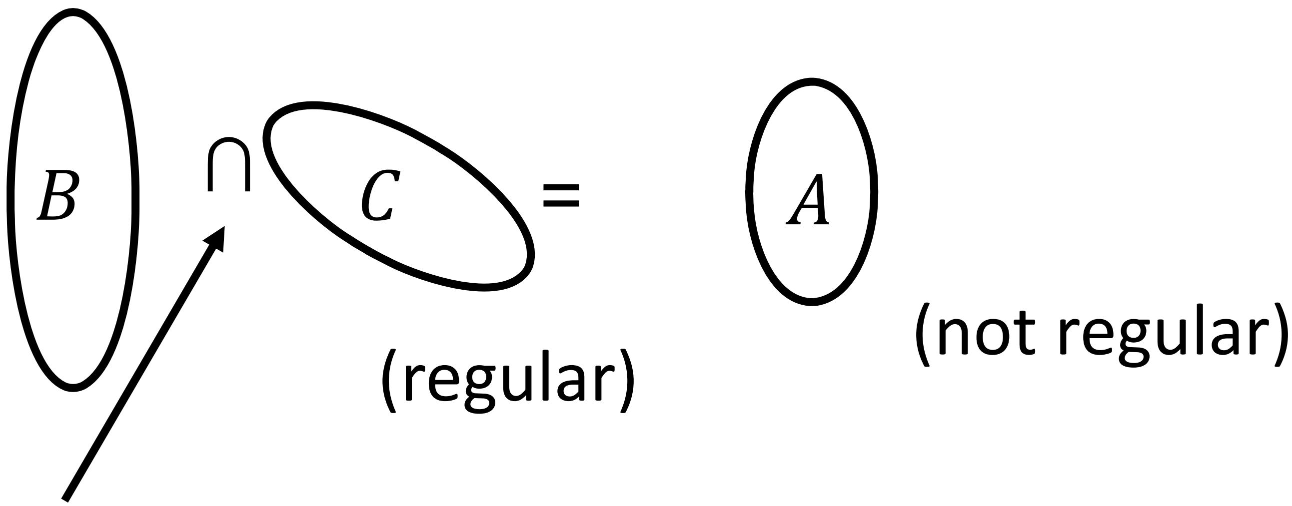
$BALANCED$ = $\{w \mid w$ has an equal # of 0s and 1s$\}$

is not regular?

$\{0^n1^n \mid n \geq 0\}$ = $BALANCED \cap \{w \mid$ all 0s in w appear before all 1s$\}$

Using Closure Properties

If A is not regular, we can show a related language B is not regular



any of $\{\circ, \cup, \cap\}$ or, for one language, $\{\neg, ^R, *\}$

<u>By contradiction:</u> If B is regular, then $B \cap C$ $(= A)$ is regular.

But A is not regular so neither is B!

Example

Prove $B = \{0^i 1^j | i \neq j\}$ is not regular using

- nonregular language

  $A = \{0^n 1^n | n \geq 0\}$ and

- regular language

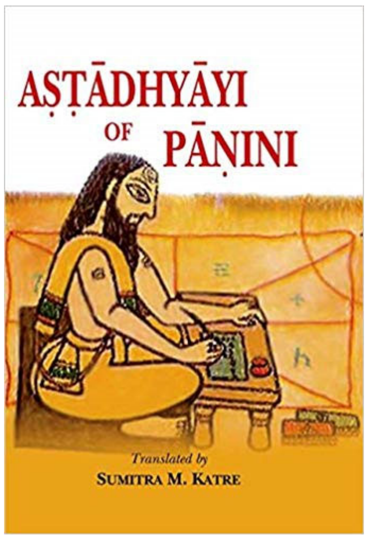
  $C = \{w \mid \text{all 0s in } w \text{ appear before all 1s}\}$

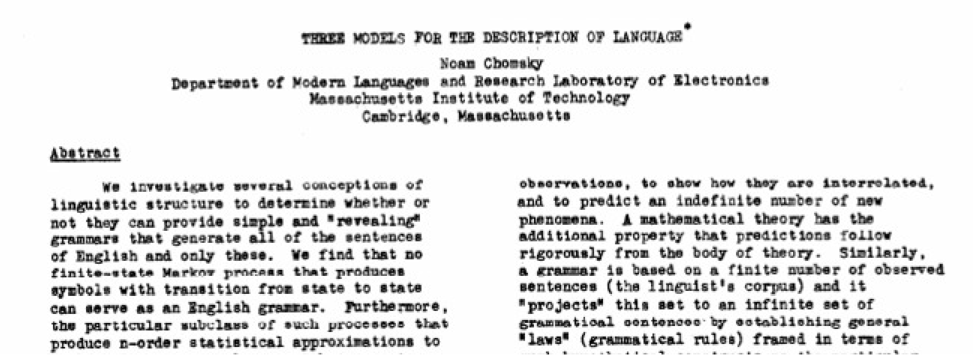
Context-Free Grammars

Some History

An abstract model for two distinct problems

Rules for parsing natural languages





THREE MODELS FOR THE DESCRIPTION OF LANGUAGE*

Noam Chomsky
Department of Modern Languages and Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts

Abstract

We investigate several conceptions of linguistic structure to determine whether or not they can provide simple and "revealing" grammars that generate all of the sentences of English and only these. We find that no finite-state Markov process that produces symbols with transition from state to state can serve as an English grammar. Furthermore, the particular subclass of such processes that produce n-order statistical approximations to

observations, to show how they are interrelated, and to predict an indefinite number of new phenomena. A mathematical theory has the additional property that predictions follow rigorously from the body of theory. Similarly, a grammar is based on a finite number of observed sentences (the linguist's corpus) and it "projects" this set to an infinite set of grammatical sentences by establishing general "laws" (grammatical rules) framed in terms of

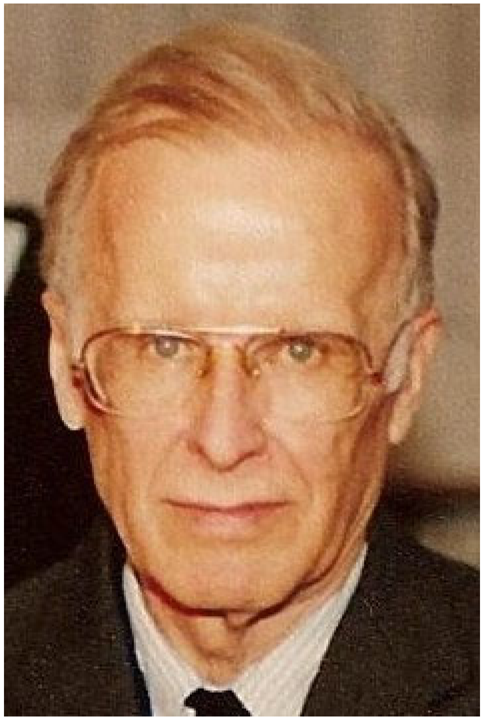
Some History

An abstract model for two distinct problems

Specification of syntax and compilation for programming languages

1977 ACM Turing Award citation
(John Backus)

For profound, influential, and lasting contributions to the design of practical high-level programming systems, notably through his work on FORTRAN, and for seminal publication of formal procedures for the specification of programming languages.



Context-Free Grammar (Informal)

Example Grammar G

$A \rightarrow 0A1$
$A \rightarrow B$
$B \rightarrow \#$

Derivation

$L(G) =$

Context-Free Grammar (Informal)

Example Grammar G

$$E \rightarrow E + T$$
$$E \rightarrow T$$
$$T \rightarrow T \times F$$
$$T \rightarrow F$$
$$F \rightarrow (E)$$
$$F \rightarrow a$$
$$F \rightarrow b$$

Derivation

$L(G) =$

Socially Awkward Professor Grammar

<PHRASE> → <FILLER><PHRASE>

<PHRASE> → <START><END>

<FILLER> → LIKE

<FILLER> → UMM

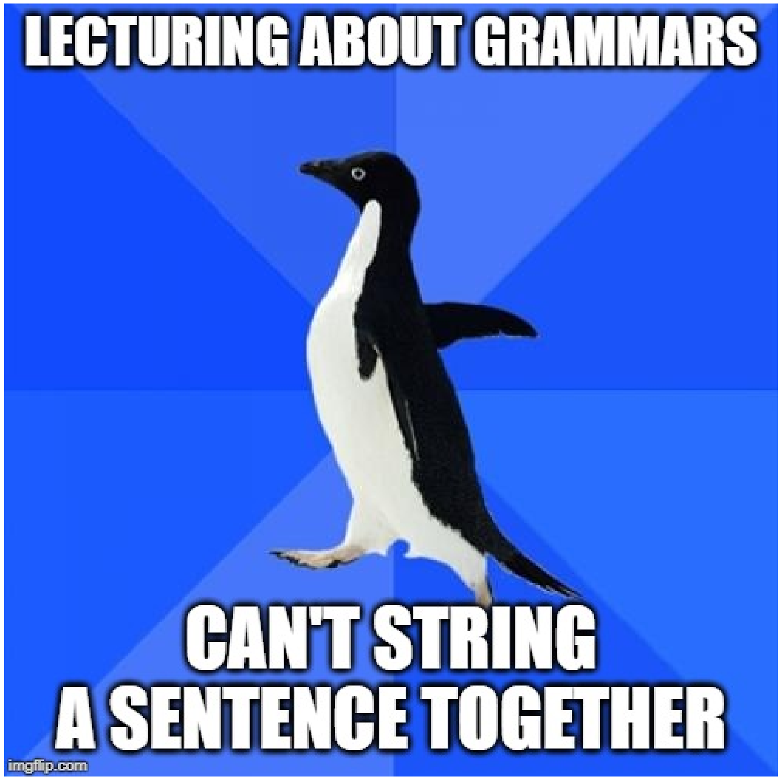
<START> → YOU KNOW

<START> → ε

<END> → WHOOPS

<END> → SORRY

<END> → $#@!



Socially Awkward Professor Grammar

<PHRASE> → <FILLER><PHRASE> | <START><END>

<FILLER> → LIKE | UMM

<START> → YOU KNOW | $\mathcal{E}$

<END> → WHOOPS | SORRY | $#@!

Context-Free Grammar (Formal)

A CFG is a 4-tuple $G = (V, \Sigma, R, S)$

- V is a finite set of variables
- Σ is a finite set of terminal symbols (disjoint from V)
- R is a finite set of production rules of the form $A \rightarrow w$, where $A \in V$ and $w \in (V \cup \Sigma)^*$
- $S \in V$ is the start symbol

Example: $G = (\{S\}, \Sigma, R, S)$ where $R = \{S \rightarrow aSb, S \rightarrow \varepsilon\}$

CFG Examples

Give context-free grammars for the following languages

1. The empty language

2. Strings of properly nested parentheses

3. Strings with equal # of a's and b's

Regular vs. Context-Free Languages

A language L is context free if it is generated by a context-free grammar