

Decidable Problems about Context-Free Languages (CFLs)

Wed October 28, 2020

HW 5/6 questions?

HW6 out

- Covers material from Chapter 4
- “Show that $\langle \text{LANG} \rangle$ is decidable” ...

Last time: Decidable DFA Languages

- $A_{\text{DFA}} = \{\langle B, w \rangle \mid B \text{ is a DFA that accepts input string } w\}$
- $A_{\text{NFA}} = \{\langle B, w \rangle \mid B \text{ is an NFA that accepts input string } w\}$
- $A_{\text{REX}} = \{\langle R, w \rangle \mid R \text{ is a regular expression that generates string } w\}$
- $E_{\text{DFA}} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \emptyset\}$
- $EQ_{\text{DFA}} = \{\langle A, B \rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$

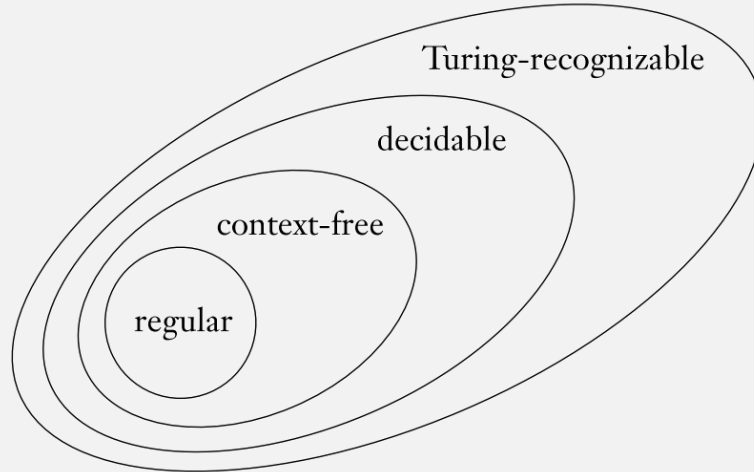
Remember:
TMs = programs
This is your library

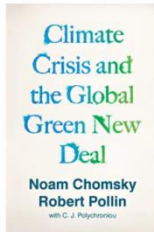
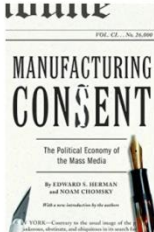
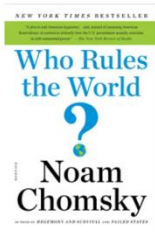

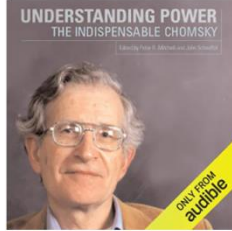
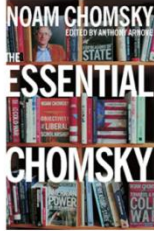
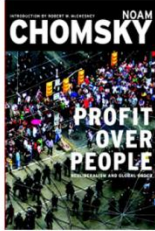

Thm: A_{CFG} is a decidable language

$$A_{CFG} = \{ \langle G, w \rangle \mid G \text{ is a CFG that generates string } w \}$$

- Related to parsing!
 - E.g., is program w a valid Python (with grammar G) program?
- Create a decider TM:
 - Try all possible derivations of G ?
 - But this might never halt, e.g., if there is a rule like: $S \rightarrow 0S$ or $S \rightarrow S$
 - This TM would be a recognizer but not a decider
- Idea: Bound the number of derivation steps?
 - Stop after some length?

Chomsky Normal Form



 <p>Climate Crisis and the Global Green New Deal: The Political Economy of Saving the Planet by Noam Chomsky, Robert Pollin, et al.</p> <p>★★★★★ ~ 25</p> <p>Paperback \$15⁸¹ \$18.95</p> <p>✓prime FREE One-Day Get it Tomorrow, Oct 29</p> <p>More Buying Choices \$13.19 (56 used & new offers)</p> <p>Other formats: Audible Audiobook, Kindle</p>	 <p>MANUFACTURING CONSENT The Political Economy of the Mass Media by Edward S. Herman and Noam Chomsky</p> <p>★★★★★ ~ 795</p> <p>Paperback \$15⁷⁵ \$21.00</p> <p>✓prime FREE One-Day Get it Tomorrow, Oct 29</p> <p>More Buying Choices \$9.39 (64 used & new offers)</p> <p>Other formats: Audible Audiobook, Kindle, Hardcover, Audio CD</p>	 <p>Who Rules the World? (American Empire Project) Part of: American Empire Project (29 Books) by Noam Chomsky</p> <p>★★★★★ ~ 415</p> <p>Paperback \$15⁷⁹ \$19.00</p> <p>✓prime FREE One-Day Get it Tomorrow, Oct 29</p> <p>More Buying Choices \$8.33 (50 used & new offers)</p> <p>Other formats: Audible Audiobook, Kindle, Hardcover, Audio CD</p>	 <p>ON ANARCHISM by Noam Chomsky and Nathan Schneider</p> <p>★★★★★ ~ 250</p> <p>Paperback \$14⁴⁵ \$15.95</p> <p>✓prime FREE Delivery Fri, Oct 30</p> <p>More Buying Choices \$10.00 (37 used & new offers)</p> <p>Other formats: Audible Audiobook, Kindle, Audio CD</p>
 <p>UNDERSTANDING POWER THE INDISPENSABLE CHOMSKY by Noam Chomsky, Peter R. Mitchell (editor), et al.</p> <p>ONLY FROM audible</p>	 <p>THE ESSENTIAL CHOMSKY by Noam Chomsky and Anthony Arnove</p> <p>★★★★★ ~ 132</p>	 <p>CHOMSKY PROFIT OVER PEOPLE by Noam Chomsky and Robert W. McChesney</p>	 <p>ON LANGUAGE Chomsky's Classic Works: Language and Responsibility and Reflections on Language</p> <p>Best Seller</p>

Chomsky Normal Form

DEFINITION 2.8

A context-free grammar is in *Chomsky normal form* if every rule is of the form

$$\begin{array}{l} A \rightarrow BC \\ A \rightarrow a \end{array} \begin{array}{l} \swarrow \\ \searrow \end{array} \boxed{\text{2 kinds of rules}}$$

where a is any terminal and A , B , and C are any variables—except that B and C may not be the start variable. In addition, we permit the rule $S \rightarrow \epsilon$, where S is the start variable.

Chomsky Normal Form: Number of Steps

- To generate a string of length n :
 - n steps: to generate all the terminals
 - $n - 1$ steps: to generate enough variables
 - Total: $2n - 1$ steps to generate length n string

Chomsky normal form

$A \rightarrow BC$

$A \rightarrow a$

Thm: Every CFG has a Chomsky Normal Form

Chomsky normal form

1. Add new start variable S_0 that does not appear on any RHS
 - I.e., add rule $S_0 \rightarrow S$, where S is old start var

$A \rightarrow BC$

$A \rightarrow a$

$$\begin{aligned} S &\rightarrow ASA \mid aB \\ A &\rightarrow B \mid S \\ B &\rightarrow b \mid \epsilon \end{aligned}$$


$$\begin{aligned} S_0 &\rightarrow S \\ S &\rightarrow ASA \mid aB \\ A &\rightarrow B \mid S \\ B &\rightarrow b \mid \epsilon \end{aligned}$$

Thm: Every CFG has a Chomsky Normal Form

Chomsky normal form

1. Add new start variable S_0 that does not appear on any RHS
 - I.e., add rule $S_0 \rightarrow S$, where S is old start var
2. Remove all “empty” rules of the form $A \rightarrow \epsilon$
 - A must not be the start variable
 - Then for every rule with A on RHS, add new rule with A deleted
 - E.g., if $R \rightarrow uAv$ is a rule, add $R \rightarrow uv$
 - Must cover all combinations if A appears more than once in a RHS
 - E.g., if $R \rightarrow uAvAw$ is a rule, add 3 rules: $R \rightarrow uvAw$, $R \rightarrow uAvw$, $R \rightarrow uvw$

$A \rightarrow BC$
 $A \rightarrow a$

$S_0 \rightarrow S$ $S \rightarrow ASA \mid aB \mid \mathbf{a}$ $A \rightarrow B \mid S \mid \epsilon$ $B \rightarrow b \mid \epsilon$		$S_0 \rightarrow S$ $S \rightarrow ASA \mid aB \mid \mathbf{a} \mid \mathbf{SA} \mid \mathbf{AS} \mid \mathbf{S}$ $A \rightarrow B \mid S \mid \epsilon$ $B \rightarrow b$
---	---	---

Thm: Every CFG has a Chomsky Normal Form

Chomsky normal form

$A \rightarrow BC$

$A \rightarrow a$

1. Add new start variable S_0 that does not appear on any RHS
 - I.e., add rule $S_0 \rightarrow S$, where S is old start var
2. Remove all “empty” rules of the form $A \rightarrow \varepsilon$
 - A must not be the start variable
 - Then for every rule with A on RHS, add new rule with A deleted
 - E.g., if $R \rightarrow uAv$ is a rule, add $R \rightarrow uv$
 - Must cover all combinations if A appears more than once in a RHS
 - E.g., if $R \rightarrow uAvAw$ is a rule, add 3 rules: $R \rightarrow uvAw$, $R \rightarrow uAvw$, $R \rightarrow uvw$
3. Remove all “unit” rules of the form $A \rightarrow B$
 - Then, for every rule $B \rightarrow u$, add rule $A \rightarrow u$

$S_0 \rightarrow S$
 $S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$
 $A \rightarrow B \mid S$
 $B \rightarrow b$



$S_0 \rightarrow S \mid ASA \mid aB \mid a \mid SA \mid AS$
 $S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$
 $A \rightarrow B \mid S$
 $B \rightarrow b$



$S_0 \rightarrow ASA \mid aB \mid a \mid SA \mid AS$
 $S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$
 $A \rightarrow S \mid b \mid ASA \mid aB \mid a \mid SA \mid AS$
 $B \rightarrow b$

Thm: Every CFG has a Chomsky Normal Form

Chomsky normal form

1. Add new start variable S_0 that does not appear on any RHS
 - I.e., add rule $S_0 \rightarrow S$, where S is old start var
2. Remove all “empty” rules of the form $A \rightarrow \epsilon$
 - A must not be the start variable

$A \rightarrow BC$
 $A \rightarrow a$

$S_0 \rightarrow ASA \mid aB \mid a \mid SA \mid AS$
 $S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$
 $A \rightarrow S \mid b \mid ASA \mid aB \mid a \mid SA \mid AS$
 $B \rightarrow b$



$S_0 \rightarrow AA_1 \mid UB \mid a \mid SA \mid AS$
 $S \rightarrow AA_1 \mid UB \mid a \mid SA \mid AS$
 $A \rightarrow b \mid AA_1 \mid UB \mid a \mid SA \mid AS$
 $A_1 \rightarrow SA$
 $U \rightarrow a$
 $B \rightarrow b$

3. Remove all empty rules of the form $A \rightarrow \epsilon$
 - Then, for every rule $B \rightarrow u$, add rule $A \rightarrow u$
4. Split up rules with RHS longer than length 2
 - E.g., $A \rightarrow wxyz$ becomes $A \rightarrow wB, B \rightarrow xC, C \rightarrow yz$
5. Replace all terminals on RHS with new rule
 - E.g., for above, add $W \rightarrow w, X \rightarrow x, Y \rightarrow y, Z \rightarrow z$

Thm: A_{CFG} is a decidable language

$$A_{\text{CFG}} = \{\langle G, w \rangle \mid G \text{ is a CFG that generates string } w\}$$

- Create decider:

$S =$ “On input $\langle G, w \rangle$, where G is a CFG and w is a string:

1. Convert G to an equivalent grammar in Chomsky normal form.
2. List all derivations with $2n - 1$ steps, where n is the length of w ; except if $n = 0$, then instead list all derivations with one step.
3. If any of these derivations generate w , *accept*; if not, *reject*.”

Thm: E_{CFG} is a decidable language.

$$E_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG and } L(G) = \emptyset \}$$

• Recall:

$$E_{\text{DFA}} = \{ \langle A \rangle \mid A \text{ is a DFA and } L(A) = \emptyset \}$$

$T =$ “On input $\langle A \rangle$, where A is a DFA:

1. Mark the start state of A .
2. Repeat until no new states get marked:
3. Mark any state that has a transition coming into it from any state that is already marked.
4. If no accept state is marked, *accept*; otherwise, *reject*.”

• “Reachability” (of accept state from start state)

Thm: E_{CFG} is a decidable language.

$$E_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG and } L(G) = \emptyset \}$$

- Create decider that calculates reachability for grammar G
 - Except start from terminals, to avoid looping

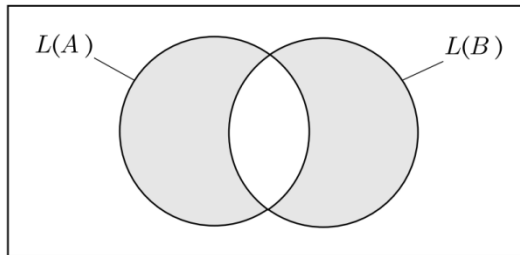
$R =$ “On input $\langle G \rangle$, where G is a CFG:

1. Mark all terminal symbols in G .
2. Repeat until no new variables get marked:
3. Mark any variable A where G has a rule $A \rightarrow U_1U_2 \cdots U_k$ and each symbol U_1, \dots, U_k has already been marked.
4. If the start variable is not marked, *accept*; otherwise, *reject*.”

Thm: EQ_{CFG} is a decidable language?

$$EQ_{CFG} = \{\langle G, H \rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$$

- Recall: $EQ_{DFA} = \{\langle A, B \rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$
- Use Symmetric Difference



$$L(C) = \emptyset \text{ iff } L(A) = L(B)$$

- C = complement, Union, intersection of machines A and B
- Can't do this for CFLs!
 - Intersection and complement are not closed for CFLs!!!

Intersection of CFLs is Not Closed!

- If closed, then intersection of these CFLs should be a CFL:

$$A = \{a^m b^n c^n \mid m, n \geq 0\}$$

$$B = \{a^n b^n c^m \mid m, n \geq 0\}$$

- But $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$

- Not a CFL!

Complement of a CFL is not Closed!

- If CFLs closed under complement:

if G_1 and G_2 context-free

$\overline{L(G_1)}$ and $\overline{L(G_2)}$ context-free

$\overline{\overline{L(G_1)} \cup \overline{L(G_1)}}$ context-free

$\overline{\overline{\overline{L(G_1)} \cup \overline{L(G_1)}}}$ context-free

$L(G_1) \cap L(G_2)$ context-free

DeMorgan's
Law!

Thm: EQ_{CFG} is a decidable language?

$$EQ_{CFG} = \{\langle G, H \rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$$

- No!
- Not recognizable either!
- You cannot decide whether two grammars are equal!
- (Can't prove until Chapter 5)

Decidability of CFGs Recap

- $A_{CFG} = \{ \langle G, w \rangle \mid G \text{ is a CFG that generates string } w \}$
 - Convert grammar to Chomsky Normal Form
 - Then check all possible derivations of length $2|w| - 1$ steps
- $E_{CFG} = \{ \langle G \rangle \mid G \text{ is a CFG and } L(G) = \emptyset \}$
 - Compute “reachability” of start variable from terminals
- $EQ_{CFG} = \{ \langle G, H \rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H) \}$
 - We couldn't prove that this is decidable!
 - (Cant use this when creating a decider)

Next time: Thms: A_{TM} is Turing-recognizable
 A_{TM} is undecidable

???

$$A_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \}$$

$U =$ “On input $\langle M, w \rangle$, where M is a TM and w is a string:

1. Simulate M on input w .
2. If M ever enters its accept state, *accept*; if M ever enters its reject state, *reject*.”

Check-in Quiz 10/28

On gradescope

End of Class Survey 10/28

See course website