

Context Free Languages II

CFLs and Regular Languages

Announcement

- About homework #4
 - The FA corresponding to the RE given in class may indeed already be minimal.
 - Non-regular language proofs
 - Use Pumping Lemma
- Homework session after lecture

Announcement

- Note about homework #2
 - For a deterministic FA, there must be a transition for *every* character from *every* state.

Before we begin

- Questions from last time?

Languages

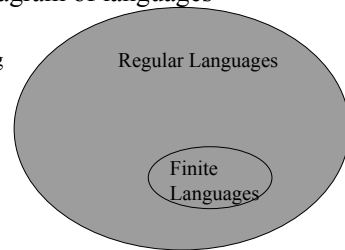
- Recall.
 - What is a language?
 - What is a class of languages?

CFLs and Regular Languages

- Venn diagram of languages

Is there something out here?

CFLs



Context Free Languages

- Context Free Languages(CFL) is the next class of languages outside of Regular Languages:
 - Means for defining: Context Free Grammar
 - Machine for accepting: Pushdown Automata

Context Free Grammars

- Let's formalize this a bit:
 - A context free grammar (CFG) is a 4-tuple: (V, Σ, S, P) where
 - V is a set of variables
 - Σ is a set of terminals
 - V and Σ are disjoint (i.e. $V \cap \Sigma = \emptyset$)
 - $S \in V$, is your start symbol

Context Free Grammars

- Let's formalize this a bit:
 - Production rules
 - Of the form $A \rightarrow \beta$ where
 - $A \in V$
 - $\beta \in (V \cup \Sigma)^*$ string with symbols from V and Σ
 - We say that γ can be derived from α in one step:
 - $A \rightarrow \beta$ is a rule
 - $\alpha = \alpha_1 A \alpha_2$
 - $\gamma = \alpha_1 \beta \alpha_2$
 - $\alpha \Rightarrow \gamma$

Context Free Grammars

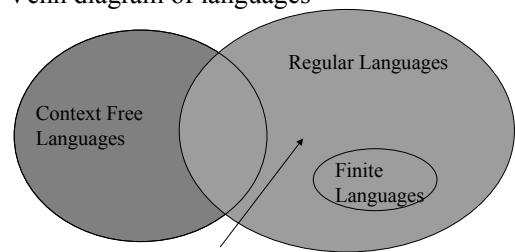
- Let's formalize this a bit:
 - Production rules
 - We say that the grammar is context-free since this substitution can take place regardless of where A is.
 - We write $\alpha \Rightarrow^* \gamma$ if γ can be derived from α in zero or more steps.

Context Free Grammars

- The language generated by a CFG
 - Let $G = (V, \Sigma, S, P)$
 - The language generated by G , $L(G)$
 - $L(G) = \{ x \in \Sigma^* \mid S \Rightarrow^* x \}$
 - A language L is a Context Free Language (CFL) iff there is a CFG G , such that
 - $L = L(G)$

CFLs and Regular Languages

- Venn diagram of languages



Is there something in here?

CFLs and Regular Languages

- Will show that all Regular Languages are CFLs
 - If L_1 and L_2 are CFLs then
 - $L_1 \cup L_2$ is a CFL
 - $L_1 L_2$ is a CFL
 - L_1^* is a CFL
 - With the above shown, showing every Regular Language is also a CFL can be shown using a basic inductive proof.

Union, Concatenation, and Kleene Star of CFLs

- Formally, Let L_1 and L_2 be CFLs. Then there exists CFGs:
 - $G_1 = (V_1, \Sigma, S_1, P_1)$
 - $G_2 = (V_2, \Sigma, S_2, P_2)$ such that
 - $L(G_1) = L_1$ and $L(G_2) = L_2$
 - Assume that $V_1 \cap V_2 = \emptyset$
- We will define:
 - $G_u = (V_u, \Sigma, S_u, P_u)$ such that $L(G_u) = L_1 \cup L_2$
 - $G_c = (V_c, \Sigma, S_c, P_c)$ such that $L(G_c) = L_1 L_2$
 - $G_k = (V_k, \Sigma, S_k, P_k)$ such that $L(G_k) = L_1^*$

Union, Concatenation, and Kleene Star of CFLs

- Union
 - Basic Idea
 - Define the new CFG so that we can either
 - start with the start variable of G_1 and follow the production rules of G_1 , or
 - start with the start variable of G_2 and follow the production rules of G_2
 - The first case will derive a string in L_1
 - The second case will derive a string in L_2

Union, Concatenation, and Kleene Star of CFLs

- Union
 - Formally
 - $G_u = (V_u, \Sigma, S_u, P_u)$
 - $V_u = V_1 \cup V_2 \cup \{S_u\}$
 - $S_u = S_u$
 - $P_u = P_1 \cup P_2 \cup \{S_u \rightarrow S_1 \mid S_2\}$

Union, Concatenation, and Kleene Star of CFLs

- Concatenation
 - General Idea
 - Define the new CFG so that
 - We force a derivation starting from the start variable of G_1 using the rules of G_1
 - After that...
 - We force a derivation starting from the start variable of G_2 using the rules of G_2

Union, Concatenation, and Kleene Star of CFLs

- Concatenation
 - Formally
 - $G_c = (V_c, \Sigma, S_c, P_c)$
 - $V_c = V_1 \cup V_2 \cup \{S_c\}$
 - $S_c = S_c$
 - $P_u = P_1 \cup P_2 \cup \{S_c \rightarrow S_1 S_2\}$

Union, Concatenation, and Kleene Star of CFLs

- Kleene Star
 - General Idea
 - Define the new CFG so that
 - We can repeatedly concatenate derivations of strings in L_1
 - Since L^* contains Λ , we must be careful to assure that there are productions in our new CFG such that Λ can be derived from the start variable

Union, Concatenation, and Kleene Star of CFLs

- Kleene star
 - Formally
 - $G_k = (V_k, \Sigma, S_k, P_k)$
 - $V_k = V_1 \cup \{S_k\}$
 - $S_k = S_k$
 - $P_k = P_1 \cup \{S_k \rightarrow S_1 S_k \mid \Lambda\}$

CFLs and Regular Languages

- Now we can complete the proof
 - Use an inductive proof

Regular Expression

- Recursive definition of regular languages / expression over Σ :
 1. \emptyset is a regular language and its regular expression is \emptyset
 2. $\{\Lambda\}$ is a regular language and Λ is its regular expression
 3. For each $a \in \Sigma$, $\{a\}$ is a regular language and its regular expression is a

Regular Expression

4. If L_1 and L_2 are regular languages with regular expressions r_1 and r_2 then
 - $L_1 \cup L_2$ is a regular language with regular expression $(r_1 + r_2)$
 - $L_1 L_2$ is a regular language with regular expression $(r_1 r_2)$
 - L_1^* is a regular language with regular expression (r_1^*)

Only languages obtainable by using rules 1-4 are regular languages.

CFLs and Regular Languages

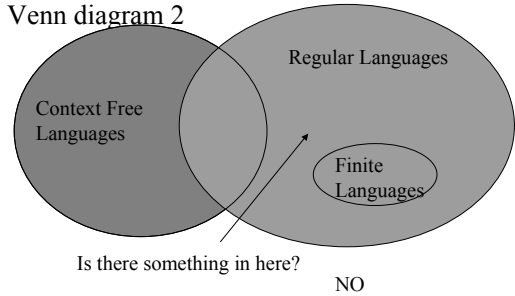
- RE \rightarrow CFG
 - Base cases
 1. \emptyset can be expressed as a CFG with no productions
 2. $\{\Lambda\}$ can be expressed by a CFG with the single production $S \rightarrow \Lambda$
 3. For each $a \in \Sigma$, $\{a\}$ can be expressed by a CFG with the single production $S \rightarrow a$

Union, Concatenation, and Kleene Star of CFLs

- RE \rightarrow CFG
 - Assume R_1 and R_2 are regular expressions that describe languages L_1 and L_2 . Then, by the induction hypothesis, L_1 and L_2 are CFLs and as such there are CFGs that describe L_1 and L_2
 - Create CFGs that describe the the languages:
 - $L_1 \cup L_2$
 - $L_1 L_2$
 - L_1^*
 - Which we just did... We are done!

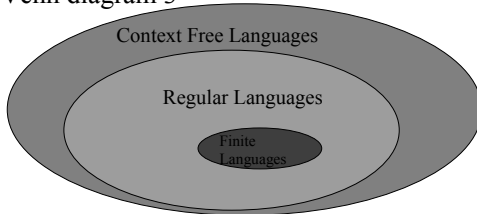
CFLs and Regular Languages

- Venn diagram 2



CFLs and Regular Languages

- Venn diagram 3



CFLs and Regular Languages

- What have we learned?
 - CFLs are closed under union, concatenation, and Kleene Star
 - Every Regular Language is also a CFL
 - We now have an algorithm, given a Regular Expression, to construct a CFG that describes the same language

CFLs and Regular Languages

- Example
 - Find a CFG for the $L = (011 + 1)^*(01)^*$
 - $(011 + 1)$ can be described by the CFG with productions:
 - $A \rightarrow 011 \mid 1$
 - $(011 + 1)^*$ can be described by the CFG with productions:
 - $B \rightarrow AB \mid \Lambda$
 - $A \rightarrow 011 \mid 1$

CFLs and Regular Languages

- Example
 - Find a CFG for the $L = (011 + 1)^*(01)^*$
 - (01) can be described by the CFG with productions:
 - $D \rightarrow 01$
 - $(01)^*$ can be described by the CFG with productions:
 - $C \rightarrow DC \mid \Lambda$
 - $D \rightarrow 01$

CFLs and Regular Languages

- Example
 - Find a CFG for the $L = (011 + 1)^*(01)^*$
 - Putting it all together
 - $(011 + 1)^*(01)^*$ can be described by the CFG with productions:
 - $S \rightarrow BC$
 - $B \rightarrow AB \mid \Lambda$
 - $A \rightarrow 011 \mid 1$
 - $C \rightarrow DC \mid \Lambda$
 - $D \rightarrow 01$
 - Questions?

Union, Concatenation, and Kleene Star of CFLs

- You can use proof of closure properties in building CFLs:
 - Example:
 - Find a CFL for $L = \{0^i 1^j 0^k \mid j > i + k\}$
 - Number of 1s is greater than the combined number of 0s
 - This language can be expressed as
 - » $L = \{0^i 1^i 1^m 1^k 0^k \mid m > 0\}$

Union, Concatenation, and Kleene Star of CFLs

- Example:
 - Find a CFL for $L = \{0^i 1^j 0^k \mid j > i + k\}$
 - This language can be expressed as
 - » $L = \{0^i 1^i 1^m 1^k 0^k \mid m > 0\}$
 - This is concatenation of 3 languages $L_1 L_2 L_3$ where
 - » $L_1 = \{0^i 1^i \mid i \geq 0\}$
 - » $L_2 = \{1^m \mid m > 0\}$
 - » $L_3 = \{1^k 0^k \mid k \geq 0\}$

Union, Concatenation, and Kleene Star of CFLs

- Example
 - CFG for $L_1 = \{0^i 1^i \mid i \geq 0\}$
 - $A \rightarrow 0A1 \mid \Lambda$
 - CFG for $L_2 = \{1^m \mid m > 0\}$
 - $B \rightarrow 1B \mid 1$
 - CFG for $L_3 = \{1^k 0^k \mid k \geq 0\}$
 - $C \rightarrow 1C0 \mid \Lambda$
- CFG for L
 - $S \rightarrow ABC$
 - $A \rightarrow 0A1 \mid \Lambda$
 - $B \rightarrow 1B \mid 1$
 - $C \rightarrow 1C0 \mid \Lambda$

Union, Concatenation, and Kleene Star of CFLs

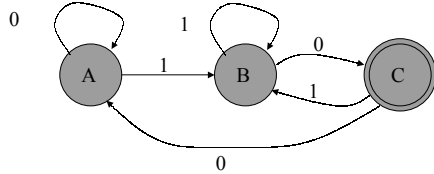
- Example
 - Formally
 - $G = (V, \Sigma, S, P)$ where
 - $V = \{S, A, B, C\}$
 - $\Sigma = \{0, 1\}$
 - $P = \{S \rightarrow ABC$
 - $A \rightarrow 0A1 \mid \Lambda$
 - $B \rightarrow 1B \mid 1$
 - $C \rightarrow 1C0 \mid \Lambda\}$

CFLs and Regular Languages

- Questions?

Regular Grammars

- Another means to show that all Regular Languages are CFL is to start with the FA for the regular language.



Regular Grammars

- Sequence of states to accept 00110
 - $A \Rightarrow 0A$
 - $\Rightarrow 00A$
 - $\Rightarrow 001B$
 - $\Rightarrow 0011B$
 - $\Rightarrow 00110C$

Regular Grammars

- Looks suspiciously like a grammar derivation where
 - CFG Variables \leftrightarrow FA States
 - CFG Productions \leftrightarrow FA Transitions
 - CFG Start Variable \leftrightarrow FA Start State

Regular Grammars

- Let's Formalize this a bit
 - Given an FA $M = (Q, \Sigma, q_0, \delta, A)$
 - Define a CFG $G = (V, \Sigma, S, P)$
 - $V = Q$
 - $S = q_0$
 - $\Sigma = \Sigma$

Regular Grammars

- Productions
 - Normal
 - Define a production
 - $D \rightarrow aB$ for each transition from D to B on input a
 - I.e. when $\delta(D, a) = B$
 - Final State
 - $D \rightarrow a$ for each transition to a final state
 - I.e. when $\delta(D, a) = C$ and $C \in A$

Regular Grammars

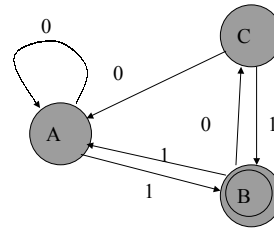
- Note that all productions are of the form
 - $A \rightarrow aB$
 - $A \rightarrow a$
- Such a grammar is called a regular grammar

Regular Grammars

- Regular Grammars accept the class of regular languages
 - L is regular iff there is a regular grammar that accepts it.
 - This regular grammar is the one we just constructed.

Regular Grammars

- Example



Regular Grammars

- $G = (V, \Sigma, S, P)$
 - $V = \{A, B, C\}$
 - $S = A$
 - $\Sigma = \{0,1\}$

Regular Grammars

- Productions P
 - Regular
 - $A \rightarrow 0A$ $B \rightarrow 1A$
 - $A \rightarrow 1B$ $C \rightarrow 0A$
 - $B \rightarrow 0C$ $C \rightarrow 1B$
 - Final
 - $A \rightarrow 1$
 - $C \rightarrow 1$

Regular Grammars

- Derivation using the CFL for 0101
 - $A \Rightarrow 0A$
 - $\Rightarrow 01B$
 - $\Rightarrow 010C$
 - $\Rightarrow 0101$
- Questions?

Regular Grammars

- Means of specifying a regular language
 1. Regular Expression
 2. Finite Automata (Deterministic)
 3. Non-deterministic Finite Automata
 4. Non-deterministic Finite Automata with Λ transitions
 5. Regular grammar
- We can convert from one specification to another.

Summary

- Context Free Languages
 - Described by Context Free Grammars
 - $G = (V, \Sigma, S, P)$
 - Closed under Union, Concatenation, and Kleene Star
 - All Regular Languages are CFLs
 - Regular Grammars
 - All Regular Languages can be described by a regular grammar.

Next time

- Next Week
 - Midterm Review
 - Midterm
- Following Week
 - Parse Trees
 - Ambiguous CFGs
 - Pushdown Automata
- Questions?
- Stick around for homework help