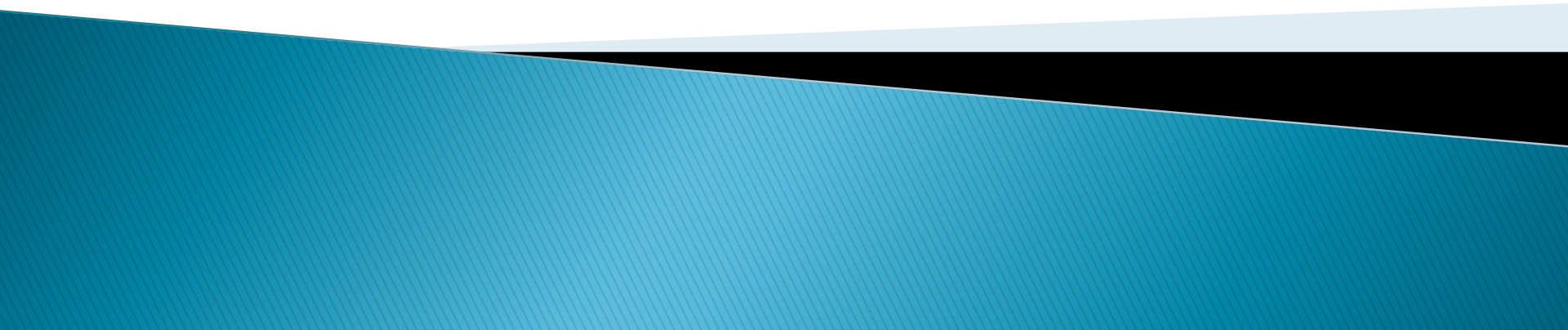
Context-Free Languages



Grammars

- Recall – grammars comprise
  - Terminals (alphabet)
  - Variables
  - Start Variable
  - Rules
- Formally:
  - A grammar is a 4-tuple: (V, Σ, R, S) where
    - V is a set of variables
    - Σ is a set of terminals
    - R is a set of production rules
    - V and Σ are disjoint (i.e. $V \cap \Sigma = \varnothing$)
    - $S \in V$, is the start variable

Grammars

- Recall
  - We say that a string w can be derived from a variable S if S can be transformed into w from a finite application of the rules, R, of the grammar
  - We say that the language of the grammar, L(G), is all strings that can be derived from the starting variable, S, by applying rules of the grammar.
    - $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w\}$
  - A language L is a context-free language if and only if there is a grammar G such that $L(G) = L$.

Grammar Example Session

- $\{ wa \mid w \in \{a,b\}^* \}$ (string has to end in a)
- $\{ w \in \{a,b\}^* \mid w$ has an even number of a's$\}$
- $\{ a^i b^j c^k \mid i=j,\ i,j,k \geq 0 \}$
- $\{ a^i b^j c^k \mid i+j=k,\ i,j,k \geq 0 \}$
- $\{w \in \{a,b\}^* \mid |w|$ is odd and middle symbol is a$\}$

Grammars and Languages

- Note – to formally prove that the language of the grammar L(G), is in fact equal to the language described, L, would typically require proving:
  - $L \subseteq L(G)$
  - $L(G) \subseteq L$
  - Using two separate inductive proofs

Closure Properties of CFLs

- CFLs are closed under union, concatenation, and Kleene star
  - 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
- Later we will see that CFLs are not closed under intersection or complementation or difference

Closure Properties of CFLs

- Formally, Let L_1 and L_2 be CFLs. Then there exist CFGs:
  - $G_1 = (V_1, \Sigma, R_1, S_1)$
  - $G_2 = (V_2, \Sigma, R_2, S_2)$ such that
  - $L(G_1) = L_1$ and $L(G_2) = L_2$
  - Assume that $V_1 \cap V_2 = \varnothing$
- We will define:
  - $G_u = (V_u, \Sigma, R_u, S_u)$ such that $L(G_u) = L_1 \cup L_2$
  - $G_c = (V_c, \Sigma, R_c, S_c)$ such that $L(G_c) = L_1 L_2$
  - $G_k = (V_k, \Sigma, R_k, S_k)$ such that $L(G_c) = L_1^*$

Union, Concatenation, and Kleene Star of CFLs

- Union
  - General 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
    - $G_u = (V_u, \Sigma, R_u, S_u)$
      - $V_u = V_1 \cup V_2 \cup \{S_u\}$
      - S_u is new start variable
      - $R_u = R_1 \cup R_2 \cup \{S_u \rightarrow S_1 \mid S_2\}$

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
    - $G_c = (V_c, \Sigma, R_c, S_c)$
      - $V_c = V_1 \cup V_2 \cup \{S_c\}$
      - S_c is new start variable
      - $R_c = R_1 \cup R_2 \cup \{S_c \rightarrow S_1 S_2\}$

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 ϵ, we must be careful to ensure that there are productions in our new CFG such that ϵ can be derived from the start variable

Union, Concatenation, and Kleene Star of CFLs

- Kleene star
  - Formally
    - $G_k = (V_k, \Sigma, R_k, S_k)$
      - $V_k = V_1 \cup \{S_k\}$
      - S_k is new start variable
      - $R_k = R_1 \cup \{S_k \rightarrow S_1 S_k \mid \epsilon\}$

Union, Concatenation, and Kleene Star of CFLs

- Closure properties can be used in building CFGs:
  - Example:
    - Find a CFG for $L = \{0^i 1^j 0^k \mid j > i + k\}$
      - Number of 1s is greater than the combined number of 0s on both sides
      - 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 CFG 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 the 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 $L_1 = \{0^i 1^i \mid i \geq 0\}$
    - $A \rightarrow 0A1 \mid \epsilon$
  - CFG for $L_2 = \{1^m \mid m > 0\}$
    - $B \rightarrow 1B \mid 1$
  - CFG for $L_3 = \{1^k 0^k \mid k \geq 0\}$
    - $C \rightarrow 1C0 \mid \epsilon$
- CFG for L
  - $S \rightarrow ABC$
  - $A \rightarrow 0A1 \mid \epsilon$
  - $B \rightarrow 1B \mid 1$
  - $C \rightarrow 1C0 \mid \epsilon$