Tutorial # 4

Context Free Grammar (CFG)

Exercise 1

- Which language is generated by the grammar $\boldsymbol{G}$ given by each of the following productions:

  $S \rightarrow aSa \mid aBa$
  $B \rightarrow bB \mid b$

  L(G)={$a^n b^m a^n$: n,m>0}

  $S \rightarrow abScB \mid \lambda$
  $B \rightarrow bB \mid b$

~~L(G)={$(ab)^n(c(b)^m)^n$: n$\geq$ 0,m>0}~~

{$(ab)^n cb^{m1} cb^{m2} \ldots\ldots cb^{mn}$|n>=0,m1,m2,....mn≥1 number of c's equal to n }

Exercise 2

- Find a CFG that generates each of the following languages over Σ = {a, b, c, d}:

1. $L(G)=\{a^n b^m c^m d^{2n} | n \geq 0, m > 0\}$

S→aSdd|A

A→bAc|bc

Exercise 2

- Find a CFG that generates each of the following languages over Σ = {a, b, c, d}:

1. $L(G)$={$a^n b^m$ |0≤n≤m≤2n}

n=0,m=0 → λ

n=1, m∈ [1,2] →ab,abb

n=2, m∈ [2,4] →aabb,aabbb,aabbbb

S→ aSb|aSbb|λ

Exercise 2

- Find a CFG that generates each of the following languages over Σ = {a, b, c, d}:

1. $L(G)=\{a^n b^m c^k \mid k=n+m\}$

S→ aSc|B

B→bBc| λ

Exercise 3

- Construct a CFG to generate the following languages over Σ = {0,1}:
- $L(G)$ = {w | w $starts\ and\ ends\ with\ the\ same\ symbol$}

S→ 0A0|1A1

A→ 0A|1A| λ

- $L(G)$={w:|w|$is\ odd$}

S→ 0A|1A

A→ 0S|1S| λ

Exercise 4

- Explain why the grammar below is ambiguous:

$\boldsymbol{S \rightarrow 0A \mid 1B}$

$\boldsymbol{A \rightarrow 0AA \mid 1S \mid 1}$

$\boldsymbol{B \rightarrow 1BB \mid 0S \mid 0}$

Exercise 4

- Explain why the grammar below is ambiguous:

$\boldsymbol{S \rightarrow 0A \mid 1B}$

$\boldsymbol{A \rightarrow 0AA \mid 1S \mid 1}$

$\boldsymbol{B \rightarrow 1BB \mid 0S \mid 0}$

001101

S→0A→ →00AA→001A→0011S→ 00110A→ 001101

S→0A→ →00AA→001SA→0011BA→ 00110A→001101

Exercise 5

- Given the following ambiguous CFG:

  $\boldsymbol{S \rightarrow Ab|aaB}$

  $\boldsymbol{A \rightarrow a|Aa}$

  $\boldsymbol{B \rightarrow b}$

- Find the string $\boldsymbol{s}$ generated by the grammar that has two leftmost derivations and show them.
- Show the two derivation trees for the string $\boldsymbol{s}$.
- Find an equivalent un-ambiguous CFG.
- Give the unique leftmost derivation and derivation tree for the string $\boldsymbol{s}$ generated from the un-ambiguous grammar above.

Exercise 5

- Given the following ambiguous CFG:

  $\boldsymbol{S \rightarrow Ab \mid aaB}$

  $\boldsymbol{A \rightarrow a \mid Aa}$

  $\boldsymbol{B \rightarrow b}$
- Find the string $\boldsymbol{s}$ generated by the grammar that has two leftmost derivations and show them.
- Show the two derivation trees for the string $\boldsymbol{s}$.
- Find an equivalent un-ambiguous CFG.
- Give the unique leftmost derivation and derivation tree for the string $\boldsymbol{s}$ generated from the un-ambiguous grammar above.

Exercise 5

- Given the following ambiguous CFG:

  $\boldsymbol{S \rightarrow Ab \,|\, aaB}$

  $\boldsymbol{A \rightarrow a \,|\, Aa}$

  $\boldsymbol{B \rightarrow b}$

- Find the string $\boldsymbol{s}$ generated by the grammar that has two leftmost derivations and show them.
- s= aab
- S→Ab→Aab→aab
- S→aaB→aab

Exercise 5

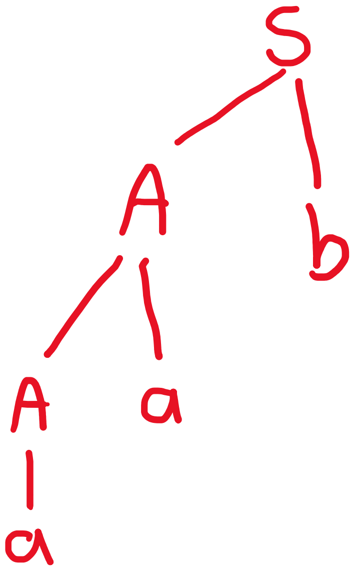
- Given the following ambiguous CFG:

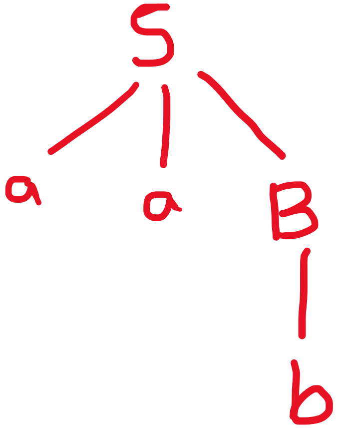
  $\boldsymbol{S \rightarrow Ab \mid aaB}$

  $\boldsymbol{A \rightarrow a \mid Aa}$

  $\boldsymbol{B \rightarrow b}$

- Show the two derivation trees for the string $\boldsymbol{s}$. {there is derivation tree for each left most derivation}





Exercise 5

- Given the following ambiguous CFG:

  $\boldsymbol{S \rightarrow Ab|aaB}$

  $\boldsymbol{A \rightarrow a|Aa}$

  $\boldsymbol{B \rightarrow b}$

- Find an equivalent un-ambiguous CFG.
- Give the unique leftmost derivation and derivation tree for the string $\boldsymbol{s}$ generated from the un-ambiguous grammar above.

S→Ab

A→a|Aa

S→Ab→Aab→aab

Exercise 6

- Convert the following ambiguous grammar into un-ambiguous grammar
  bexp → bexp or bexp | bexp and bexp |not bexp| T |F
- where bexp represents Boolean expression, T represents True and F represents False.
- **To convert the given grammar into its corresponding unambiguous grammar, we implement the precedence and associativity constraints. We have-**
- **Given grammar consists of the following operators-**

  or , and , not
- **Given grammar consists of the following operands-**

  T , F
- **The priority order is-**

  (T , F) > not > and > or

bexp → bexp or M | M

M → M and N | N

N → not N | T | F