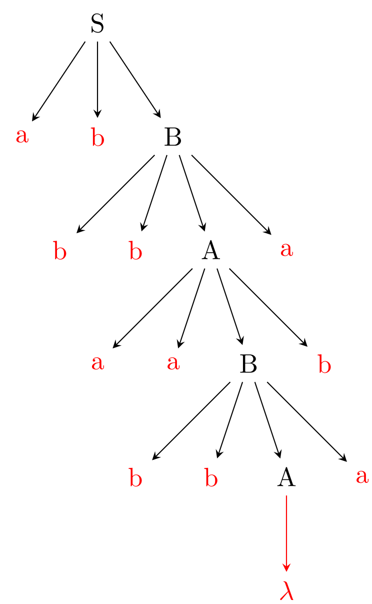
Problem 4

Give a derivation tree for w = abbbaabbaba using the grammar $\mathbf{G} = (\{\mathbf{S}\}, \{\mathbf{a}, \mathbf{b}\}, \mathbf{S}, \mathbf{P})$ with productions: $\mathbf{S} \to \mathbf{abB},\ \mathbf{A} \to \mathbf{aaBb},\ \mathbf{B} \to \mathbf{bbAa},\ \mathbf{A} \to \lambda$



Problem 9(d)

Find a context free grammar for $\mathbf{L} = \{\mathbf{a^n b^m} \ :\ \mathbf{2n \leq m \leq 3n},\ \mathbf{n \geq 0},\ \mathbf{m \geq 0}\}$

Consider the grammar $G = \{\{S\}, \{a, b\}, S, P\}$, with productions:

$$S \to aSbb$$
$$S \to aSbbb$$
$$S \to \lambda$$

If you only ever apply the first rule, you will have twice as many b's as a's. If you only ever apply the second rule, you will have three times as many b's as a's.

If you mix them up, you cannot go below the lower bound of $2n$, and cannot go above the upper bound of $3n$, so it will be somewhere between $2n$ and $3n$, so the grammar defines the language.

Problem 12(b)

Find a context free grammar for
$\mathbf{L = \{a^n b^m c^k \ : \ n = m \text{ or } m \neq k, n \geq 0, m \geq 0, k \geq 0\}}$

Consider the grammar $G = \{\{S, E, N, D, A, B^+, C, C^+\}, \{a, b, c\}, S, P\}$, with productions:

$$
\begin{aligned}
S &\to EC \mid AN & A &\to aA \mid \lambda \\
E &\to aEb \mid \lambda & B^+ &\to bB^+ \mid b \\
N &\to B^+D \mid DC^+ & C &\to cC \mid \lambda \\
D &\to bDc \mid \lambda & C^+ &\to cC^+ \mid c
\end{aligned}
$$

This one isn't very concise or elegant, but the idea is that you:

- Decide whether you want $n = m$ (E) or $m \neq k$ (N)
- If Equal:
  - Leave room for as many c's as you want on the end
  - Nest as many aEb's as you want
- If Not equal
  - Leave room for as many a's as you want at the front
  - Choose which letter is going to have more than the other, generate at least one of those
  - Nest Down as many bDc's as you want

Problem 18

Show that the following language is context-free:

$$\mathbf{L = \{uvwv^R \ : \ u, v, w \in \{a, b\}^+, |u| = |w| = 2\}}$$

To show that it's a context-free language, let's construct a context-free grammar for it.

Let that grammar be $G = \{S, U, W, V, a, b, S, P\}$, with productions:

$$
\begin{aligned}
S &\to UV \\
U &\to aa \mid ab \mid ba \mid bb \\
V &\to aVa \mid bVb \\
V &\to U
\end{aligned}
$$

This grammar:

- Starts by letting you generate whatever u is (using U, which exhausts all two character strings over $\{a, b\}$).
- Lets you generate v and v^R by nesting V's
- Only lets you get rid of V by replacing it with w

Problem 8

Show that the following grammar is ambiguous:

$$\begin{aligned} S &\to AB \mid aaaB \\ A &\to a \mid Aa \\ B &\to b \end{aligned}$$

Just looking at it, it seems like we'll be able to generare $aaab$ in two different ways.

The easy way is to go $S \Rightarrow aaaB \Rightarrow aaab$.

You can also get there from $S \Rightarrow AB \Rightarrow AaB \Rightarrow AaaB \Rightarrow aaaB \Rightarrow aaab$.

These derivations are clearly different, you make completely different choices for the replacement of S, but the generated string is the same.

Because there are two distinct ways to generate this string, the grammar is ambiguous

Problem 3

Transform the grammar $\mathbf{S \to aSaaA \mid A}$, $\mathbf{A \to abA \mid bb}$ into Chomsky normal form

CNF requires two non-terminals or a single terminal on the right hand side of every rule.

We can take rules one by one.

$$\begin{aligned} A \to abA &\implies A \to TA,\ T \to \Sigma\Omega,\ \Sigma \to a,\ \Omega \to b \\ A \to bb &\implies A \to \Omega\Omega \\ S \to aSaaA &\implies S \to UV,\ U \to \Sigma W,\ W \to \Sigma\Sigma,\ V \to \Sigma A \\ S \to A &\implies S \to A\Lambda,\ \Lambda \to \lambda \end{aligned}$$

So our final productions are

$$
\begin{aligned}
\Sigma &\to a \\
\Omega &\to b \\
S &\to UV \mid A\Lambda \\
\Lambda &\to \lambda \\
U &\to \Sigma W \\
W &\to S\Sigma \\
V &\to \Sigma A \\
A &\to TA \mid \Omega\Omega \\
T &\to \Sigma\Omega
\end{aligned}
$$