1 Regular Languages

1.1 Regular Expressions

Robots speak a language consisting of the alphabet $\Sigma = \{\texttt{b}, \texttt{p}, \texttt{o}, \texttt{e}\}$. A valid sentence in robot-speak consists of 0 or more words concatenated together. A word consists of a `b`, followed by 1 or more instances of a vowel, followed by a `p`. For example, beep, boop, and beeepboopbep are valid sentences in robotspeak. However, beop, bp, and oop are not valid words (or sentences). Write a regular expression for valid sentences in robotspeak.

1.2 NFAs

Construct NFAs for the following regular expressions: `a|b`, `ab`, `a*` (they should be very simple). Then, construct an NFA equivalent to your robotspeak regular expression.

1.3 DFAs

Construct a DFA equivalent to your NFA for robotspeak.

2 Context-Free Languages

2.1 Parse Tree Ambiguity

The following context-free grammar describes a few operations that can be done to the numbers 2, 3, and 5:

$$
\begin{aligned}
\langle \text{Expr} \rangle &\rightarrow \langle \text{Expr} \rangle\ \langle \text{Op} \rangle\ \langle \text{Expr} \rangle \\
\langle \text{Expr} \rangle &\rightarrow \texttt{2} \mid \texttt{3} \mid \texttt{5} \\
\langle \text{Op} \rangle &\rightarrow \texttt{*} \mid \texttt{+}
\end{aligned}
$$

As you can probably tell, this grammar is ambiguous! Draw two possible parse trees for the following expression:

```
2 + 3 * 5
```

2.2 Right Associativity

We want to resolve this grammar's ambiguity and produce the correct parse tree for this expression. Modify this grammar so that it always evaluates the rightmost operator first.

2.3 Hacking the Grammar

Modify this grammar so that it always evaluates multiplications before additions.