1. **Strange Loops:** The following system, based on a system called Miu, is perhaps famously mentioned in Douglas Hofstadter’s book, *Gödel, Escher, Bach.*

\[
\begin{align*}
\text{M} & \xrightarrow{1} \text{I} \\
\text{I} & \xrightarrow{2} \text{M} \\
\text{M} & \xrightarrow{3} \text{III} \cdot \text{Y} \\
\text{Y} & \xrightarrow{4} \text{xy} \\
\text{xy} & \xrightarrow{5} \text{M}\text{I} \\
\end{align*}
\]

(a) ★ Is \text{MII} derivable? If so, show the derivation tree. If not, explain why not.

(b) ★★ Is \text{XI} admissible? Is it derivable? Justify your answer.

(c) ★★★☆ Perhaps famously, \text{MU} is not admissible. Prove this using rule induction. *Hint:* Try proving something related to the number of Is in the string.

(d) Here is another language, which we’ll call \text{Mi}:

\[
\begin{align*}
\text{M} & \xrightarrow{A} \text{I} \\
\text{I} & \xrightarrow{B} \text{M} \\
\text{M} & \xrightarrow{C} \text{III} \cdot \text{Y} \\
\text{Y} & \xrightarrow{D} \text{xy} \\
\text{xy} & \xrightarrow{E} \text{M} \\
\end{align*}
\]

i. ★★★ Prove using rule induction that all strings in \text{Mi} could be expressed as follows, for some \(k\) and some \(i\), where \(2^k - 6i > 0\) (where \(\mathcal{C}^n\) is the character \(\mathcal{C}\) repeated \(n\) times):

\[
\text{M} \cdot \text{I}^{2^k - 6i}
\]

ii. We will now prove the opposite claim that, for all \(k\) and \(i\), assuming \(2^k - 6i > 0\):

\[
\text{M} \cdot \text{I}^{2^k - 6i} \rightarrow \text{Mi}
\]

To prove this we will need a few lemmas which we will prove separately.

\(\alpha\) ★★ Prove, using induction on the natural number \(k\) (i.e when \(k = 0\) and when \(k = k' + 1\)), that \(\text{M} \cdot \text{I}^{2^k}\) \(\text{Mi}\)

\(\beta\) ★★ Prove, using induction on the natural number \(i\), that \(\text{M} \cdot \text{I}^k \cdot \text{Mi}\) implies \(\text{M} \cdot \text{I}^{k - 6i}\) \(\text{Mi}\), assuming \(k - 6i > 0\).

Hence, as we know \(\text{M} \cdot \text{I}^{2^k}\) \(\text{Mi}\) for all \(k\) from lemma \(\alpha\), we can conclude from lemma \(\beta\) that \(\text{M} \cdot \text{I}^{2^k - 6i}\) \(\text{Mi}\) for all \(k\) and all \(i\) where \(2^k - 6i > 0\) by modus ponens.

These two parts prove that the language \(\text{Mi}\) is exactly characterised by the formulation \(\text{M} \cdot \text{I}^{2^k - 6i}\) where \(2^k - 6i > 0\). A very useful result!

iii. ★ Hence prove or disprove that the following rule is admissible in \(\text{Mi}\):

\[
\frac{\text{M}}{\text{M}} \rightarrow \text{I}_{\text{LEMM}_1}
\]

iv. ★ Why is the following rule **not** admissible in \(\text{Mi}\)?

\[
\frac{\text{M}}{\text{I}_{\text{LEMM}_2}}
\]

v. ★★★ Prove that, for all \(s\), \(s \cdot \text{Mi} \rightarrow \text{I}_{\text{LEMM}_2}\). Note that using straightforward rule induction appears to necessitate \(\text{LEMM}_2\) above, which we know is not admissible. Try proving using the characterisation we have already developed.
2. **Counting Sticks:** The following language (also presented in a similar form by Douglas Hofstadter, but the original invention is not his) is called the $\Phi\Psi$ system. Unlike the Miu language discussed above, this language is not comprised of a single judgement, but of a ternary relation, written $x \Phi y \Psi z$, where $x$, $y$ and $z$ are strings of hyphens (i.e ‘-’), which may be empty ($\epsilon$). The system is defined as follows:

$$
\begin{align*}
&\text{\epsilon} \Phi_x \Psi_x \epsilon \\
&\frac{x \Phi y \Psi z \rightarrow B}{-x \Phi y \Psi z \rightarrow f}
\end{align*}
$$

(a) [⋆] Prove that $-- \Phi --- \Psi ------$.

(b) [⋆] Is the following rule admissible? Is it derivable? Explain your answer

$$
\frac{-x \Phi y \Psi z \rightarrow f'}{x \Phi y \Psi z}
$$

(c) [★★] Show that $x \Phi \epsilon \Psi x$, for all hyphen strings $x$, by doing induction on the length of the hyphen string (where $x = \epsilon$ and $x = \epsilon'$)

(d) [★★★] Show that if $-x \Phi y \Psi z$ then $x \Phi -y \Psi z$, for all hyphen strings $x$, $y$ and $z$, by doing induction on the size of $x$.

(e) [★★] Show that $x \Phi y \Psi z$ implies $y \Phi x \Psi z$.

(f) [★★] Have you figured out what the $\Phi\Psi$ system actually is? Prove that if $-x \Phi -y \Psi -z$, then $z = -x + y$ (where $-x$ is a hyphen string of length $x$).

3. **Ambiguity and Simultaneity:** Here is a simple grammar for a functional programming language ¹:

$$
\begin{align*}
&x \in \mathbb{N} \\
x &\in Expr \\
&\text{VAR.} \\
e_1 &\in Expr e_2 \\
e &\in Expr \\
&A P P L. \\
e &\in \text{Abst.} \\
e &\in \text{Paren.}
\end{align*}
$$

(a) [⋆] Is this grammar ambiguous? If not, explain why not. If so, give an example of an expression that has multiple parse trees.

(b) [★★] Develop a new (unambiguous) grammar that encodes the left associativity of application, that is $1 2 3 4$ should be parsed as $((1 2) 3) 4$ (modulo parentheses). Furthermore, lambda expressions should extend as far as possible, i.e $\lambda 1 2$ is equivalent to $\lambda (1 2)$ not $(\lambda 1) 2$.

(c) [★★★] Prove that all expressions in your grammar are representable in Expr, that is, that your grammar describes only strings that are in Expr.

4. **Regular Expressions:** Consider this language used to describe regular expressions consisting of:

- single characters, written $c$
- Sequential composition, written $R; R$
- Nondeterministic choice, written $R | R$.
- Kleene star, written $R^*$.
- Grouping parentheses.

$$
\begin{align*}
&c \in \text{Char} \\
c &\in R \\
a &\in R b &\in R \\
a &\in R b &\in R a &\in R \\
a &\in R a &\in R (a) &\in R
\end{align*}
$$

(a) [⋆] In what way is this grammar ambiguous? Identify an expression with multiple parse trees.

(b) [⋆] Devise an alternative grammar that is unambiguous, order of operations should be such that

$$
\begin{align*}
a; b; c &\in R a; d &\in R e
\end{align*}
$$

is parsed with the grouping indicated by the parentheses in:

$$
(a; (b; (c*))) | ((a; d) | e)
$$

¹If you're interested, it's called lambda calculus, with de Bruijn indices syntax, not that it's relevant to the question!
5. **Key Combinations:** Consider the language used to document key combinations:

\[
x \in \{a, b, \ldots, \text{Shift}\} \\
\text{Key} \quad c_1 \ K \quad c_2 \ K \quad \text{Hold} \quad c_1 \ K \quad c_2 \ K \quad \text{Then} \quad c \ K \quad \text{Paren}
\]

For example, `Ctrl + C` is a string in this language.

(a) [•] Find an example of ambiguity in this language.

(b) [•] Eliminate ambiguity such that

\[
\text{is parsed with this grouping:} \quad (\text{Ctrl} \ ((\text{Ctrl} + \text{Shift} + \text{Q})))
\]

and such that

\[
\text{is parsed with the following grouping:} \quad (\text{Ctrl} \ + \text{Shift} + \text{Q})
\]