COMP3161/COMP9164
Preliminaries Exercises

Liam O'Connor

September 27, 2019

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

$$\frac{}{\texttt{MI}\ \textsc{Miu}}1 \quad \frac{x\texttt{I}\ \textsc{Miu}}{x\texttt{IU}\ \textsc{Miu}}2 \quad \frac{\texttt{M}x\ \textsc{Miu}}{\texttt{M}xx\ \textsc{Miu}}3 \quad \frac{x\texttt{III}y\ \textsc{Miu}}{x\texttt{U}y\ \textsc{Miu}}4 \quad \frac{x\texttt{UU}y\ \textsc{Miu}}{xy\ \textsc{Miu}}5$$

   (a) [$\star$] Is $\texttt{MUII}$ Miu derivable? If so, show the derivation tree. If not, explain why not.

   (b) [$\star\star$] Is $\dfrac{x\texttt{IU}\ \textsc{Miu}}{x\texttt{I}\ \textsc{Miu}}$ admissible? Is it derivable? Justify your answer.

   (c) [$\star\star\star\star$] Perhaps famously, $\texttt{MU}$ Miu is not admissible. Prove this using rule induction. *Hint*: Try proving something related to the number of $\texttt{I}$s in the string.

   (d) Here is another language, which we'll call Mi:

$$\frac{}{\texttt{MI}\ \textsc{Mi}}A \quad \frac{\texttt{M}x\ \textsc{Mi}}{\texttt{M}xx\ \textsc{Mi}}B \quad \frac{x\texttt{IIIIII}y\ \textsc{Mi}}{xy\ \textsc{Mi}}C$$

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

$$\texttt{M}\,\texttt{I}^{2^k-6i}$$

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

$$\texttt{M}\,\texttt{I}^{2^k-6i}\ \textsc{Mi}$$

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

      α) [$\star\star$] Prove, using induction on the natural number k (i.e when $k = 0$ and when $k = k' + 1$), that $\texttt{M}\,\texttt{I}^{2^k}$ Mi

      β) [$\star\star$] Prove, using induction on the natural number i, that $\texttt{M}\,\texttt{I}^{k}$ Mi implies $\texttt{M}\,\texttt{I}^{k-6i}$ Mi, assuming $k - 6i > 0$.

      Hence, as we know $\texttt{M}\,\texttt{I}^{2^k}$ Mi for all k from lemma α, we can conclude from lemma β that $\texttt{M}\,\texttt{I}^{2^k-6i}$ Mi for all k and all i where $2^k - 6i > 0$ by modus ponens.

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

      iii. [$\star$] Hence prove or disprove that the following rule is admissible in Mi:

$$\frac{\texttt{M}xx\ \textsc{Mi}}{\texttt{M}x\ \textsc{Mi}}\textsc{Lem}_1$$

      iv. [$\star$] Why is the following rule **not** admissible in Mi?

$$\frac{xy\ \textsc{Mi}}{x\texttt{IIIIII}y\ \textsc{Mi}}\textsc{Lem}_2$$

      v. [$\star\star\star$] Prove that, for all s, s Mi $\implies$ s Miu. Note that using straightforward rule induction appears to necessitate $\textsc{Lem}_2$ 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 (ϵ). The system is defined as follows:

$$\frac{}{\epsilon\ \Phi\ x\ \Psi\ x}B \qquad \frac{x\ \Phi\ y\ \Psi\ z}{\texttt{-}x\ \Phi\ y\ \Psi\ \texttt{-}z}I$$

   (a) [$\star$] Prove that `--` Φ `---` Ψ `-----`.

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

$$\frac{\texttt{-}x\ \Phi\ y\ \Psi\ \texttt{-}z}{x\ \Phi\ y\ \Psi\ z}I'$$

   (c) [$\star\star$] 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 $x = \texttt{-}x'$).

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

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

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

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

$$\frac{x \in \mathbb{N}}{x\ \mathit{Expr}}\textsc{Var.} \quad \frac{e_1\ \mathit{Expr} \quad e_2\ \mathit{Expr}}{e_1 e_2\ \mathit{Expr}}\textsc{Appl.} \quad \frac{e\ \mathit{Expr}}{\lambda e\ \mathit{Expr}}\textsc{Abst.} \quad \frac{e\ \mathit{Expr}}{(e)\ \mathit{Expr}}\textsc{Paren.}$$

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

   (b) [$\star\star$] 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) [$\star\star\star$] 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 $R \mid R$.
   - Kleene star, written $R\star$.
   - Grouping parentheses.

$$\frac{\texttt{c}\ \mathbf{Char}}{\texttt{c}\ \mathbf{R}} \quad \frac{a\ \mathbf{R} \qquad b\ \mathbf{R}}{a; b\ \mathbf{R}} \quad \frac{a\ \mathbf{R} \qquad b\ \mathbf{R}}{a \mid b\ \mathbf{R}} \quad \frac{a\ \mathbf{R}}{a\star\mathbf{R}} \quad \frac{a\ \mathbf{R}}{(a)\ \mathbf{R}}$$

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

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

$$\texttt{a}; \texttt{b}; \texttt{c}\star \mid \texttt{a}; \texttt{d} \mid \texttt{e}$$

   is parsed with the grouping indicated by the parentheses in:

$$(\texttt{a}; (\texttt{b}; (\texttt{c}\star))) \mid ((\texttt{a}; \texttt{d}) \mid \texttt{e})$$

[1] if 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:

$$\frac{x \in \{\mathrm{a}, \mathrm{b}, \ldots, \mathrm{Shift}\}}{\boxed{x}\ \mathbf{K}}\textit{Key} \qquad \frac{c_1\ \mathbf{K} \quad c_2\ \mathbf{K}}{c_1 \textbf{+} c_2\ \mathbf{K}}\textit{Hold} \qquad \frac{c_1\ \mathbf{K} \quad c_2\ \mathbf{K}}{c_1 c_2\ \mathbf{K}}\textit{Then} \qquad \frac{c\ \mathbf{K}}{(c)\ \mathbf{K}}\textit{Paren}$$

For example $\boxed{\mathrm{Ctrl}} \textbf{+} \boxed{\mathrm{C}}$ is a string in this language.

(a) [$\star$] Find an example of ambiguity in this language.

(b) [$\star$] Eliminate ambiguity such that

$$\boxed{q}\ \boxed{w} \textbf{+} \boxed{e}\ \boxed{r}\ \boxed{t}$$

is parsed with this grouping:

$$(\boxed{q}\ ((\boxed{w} \textbf{+} \boxed{e})(\boxed{r}\ \boxed{t})))$$

and such that

$$\boxed{\mathsf{Ctrl}} \textbf{+} \boxed{\mathsf{Shift} \Uparrow} \textbf{+} \boxed{\mathsf{Q}}$$

is parsed with the following grouping:

$$(\boxed{\mathsf{Ctrl}} \textbf{+} \boxed{\mathsf{Shift} \Uparrow}) \textbf{+} \boxed{\mathsf{Q}}$$