Concepts of Programming Language Design

Exercises

Liam O'Connor-Davis
Gabriele Keller

November 6, 2024

1. **Strange Loops:** The following system, based on a system called MIU, is perhaps famously mentioned in Douglas Hofstadter's book, Gödel, Escher, Bach (see, for example, "MU puzzle" on Wikipedia).

$$\frac{}{\texttt{MI}\ \textbf{MIU}}\ (\textit{MIU-1})$$

$$\frac{x\texttt{I}\ \textbf{MIU}}{x\texttt{IU}\ \textbf{MIU}}\ (\textit{MIU-2})$$

$$\frac{\texttt{M}x\ \textbf{MIU}}{\texttt{M}xx\ \textbf{MIU}}\ (\textit{MIU-3})$$

$$\frac{x\texttt{III}y\ \textbf{MIU}}{x\texttt{U}y\ \textbf{MIU}}\ (\textit{MIU-4})$$

$$\frac{x\texttt{UU}y\ \textbf{MIU}}{xy\ \textbf{MIU}}\ (\textit{MIU-5})$$

   (a) Is $\texttt{MUII}\ \textbf{MIU}$ derivable? If so, show the derivation tree. If not, explain why not.

   (b) Is $\dfrac{x\texttt{IU}\ \textbf{MIU}}{x\texttt{I}\ \textbf{MIU}}$ admissible? Is it derivable? Justify your answer.

   (c) **Tricky:** Perhaps famously, $\texttt{MU}\ \textbf{MIU}$ 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:

$$\frac{}{\texttt{MI}\ \textbf{MI}}\ (\textit{MI-1})$$

$$\frac{\texttt{M}x\ \textbf{MI}}{\texttt{M}xx\ \textbf{MI}}\ (\textit{MI-2})$$

$$\frac{x\texttt{IIIIII}y\ \textbf{MI}}{xy\ \textbf{MI}}\ (\textit{MI-3})$$

      i. 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 $2^k - 6i > 0$:

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

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

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

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

Hence, as we know $\texttt{M I}^{2^k}$ **MI** for all k from lemma α, we can conclude from lemma β that $\texttt{M I}^{2^k-6i}$ **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 $\texttt{M I}^{2^k-6i}$ where $2^k - 6i > 0$. A very useful result!

iii. Hence prove or disprove that the following rule is admissible in MI:

$$\frac{\texttt{M}xx\ \textbf{MI}}{\texttt{M}x\ \textbf{MI}} \quad (\text{Lemma}_1)$$

iv. Why is the following rule **not** admissible in MI?

$$\frac{xy\ \textbf{MI}}{x\texttt{IIIIII}y\ \textbf{MI}} \quad (\text{Lemma}_2)$$

v. Prove that, for all s, s **MI** $\implies$ s **MIU**. You can prove it using the characterisation we have already developed, or directly by induction.

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:

$$\frac{}{\epsilon\ \Phi\ x \Psi\ x} \quad (\Phi\Psi - 1)$$

$$\frac{x\ \Phi\ y\ \Psi\ z}{\texttt{-}x\ \Phi\ y\ \Psi\ \texttt{-}z} \quad (\Phi\Psi - 2)$$

(a) Prove that -- Φ --- Ψ -----.

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

$$\frac{\texttt{-}x\ \Phi\ y\ \Psi\ \texttt{-}z}{x\ \Phi\ y\ \Psi\ z} \quad (\Phi\Psi - 2')$$

(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 $x = \texttt{-}x'$).

(d) 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 a rule induction on the premise.

(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 $\texttt{-}^x\ \Phi\ \texttt{-}^y\ \Psi\ \texttt{-}^z$, 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]:

$$\frac{x \in \mathbb{N}}{x\ \textit{Expr}} \quad (\text{E-1})$$

$$\frac{e_1\ \textit{Expr} \quad e_2\ \textit{Expr}}{e_1 e_2\ \textit{Expr}} \quad (\text{E-2})$$

$$\frac{e\ \textit{Expr}}{\lambda e\ \textit{Expr}} \quad (\text{E-3})$$

$$\frac{e\ \textit{Expr}}{(e)\ \textit{Expr}} \quad (\text{E-4})$$

[1] if you're interested, it's called lambda calculus, with de Bruijn indices syntax, not that it's relevant to the question!

(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 λ `1 2` is equivalent to λ `(1 2)` not `(`λ `1) 2`.

(c) **Tricky** Prove that all expressions in your grammar are representable in ***Expr***, that is, that your grammar describes only strings that are in ***Expr***.