1 Forward Logic with Inference Rules

Task 1. Give the inference rules for a forward logic program \text{length}(l, n) which derives the atom no if and only if \(n\) (which satisfies the \text{nat} predicate) is not the length of list \(l\). You may assume that \(n\) and \(l\) are ground.

Solution 1:

\[
\begin{array}{l}
\frac{\text{length}([], s(N))}{\text{no}} \quad \text{zero} \quad \\
\frac{\text{length}([X | L], z)}{\text{no}} \quad \text{zero} \\
\frac{\text{length}(X, N)}{\text{dec}}
\end{array}
\]

Task 2. Recall the grammar representing natural numbers:

\[ n ::= z | s(n) \]

Give the inference rules for a forward logic program \text{factor}(m, n) which derives the atom no if and only if \(m\) does not evenly divide \(n\). You may assume that \(m\) and \(n\) are ground.

Solution 2:

\[
\begin{array}{l}
\frac{\text{factor}(m, n)}{\text{div}(m, n, m)} \quad \text{def} \\
\frac{\text{div}(s(m), s(n), d)}{\text{div}(m, n, d)} \quad \text{decr} \\
\frac{\text{div}(z, s(n), d)}{\text{div}(d, s(n), d)} \quad \text{rep} \\
\frac{\text{div}(s(m), z, d)}{\text{no}} \quad \text{nz}
\end{array}
\]

2 Functional Evaluation with Forward Chaining

Consider the language of the untyped lambda calculus.

\[ e ::= x | \lambda x.e | e_1 e_2 \]

We can write a set of rules using three predicates

\[
\begin{array}{l}
\text{eval}(e) \\
\text{e} \mapsto^* e' \quad \text{e reduces to } e' \\
\text{e} \mapsto v \quad \text{e evaluates to } v
\end{array}
\]

so that we can evaluate \(e\) with forward chaining, by seeding the system with eval(e) and waiting for a fact of the form \(e \mapsto v\) to appear.

Task 3. Define such a set of rules.

3 Return to Focusing

Recall the rules for even and odd natural numbers:

\[
\frac{\text{even}(z)}{\text{ev}_z} \quad \frac{\text{odd}(N)}{\text{ev}_s} \quad \frac{\text{even}(N)}{\text{od}_s}
\]

**Task 4.** Give a proof, using the focusing rules, of odd(s(s(z))).

**Solution 4:** Define \( \Gamma_{eo} = \text{even}(z), \forall n. \text{odd}(n) \supset \text{even}(s(n)), \forall n. \text{even}(n) \supset \text{odd}(s(n)). \)

Our goal sequent is \( \Gamma_{eo} \rightarrow \text{even}(s(s(z))). \)

\[
\begin{align*}
\frac{\Gamma_{eo} \quad \text{id}}{\text{id}} & \quad \frac{\Gamma_{eo} \quad \text{blur}}{\text{blur}_L} \\
\frac{\Gamma_{eo} [\text{ev}_z] \quad \text{id}}{\text{id}} & \quad \frac{\Gamma_{eo} \quad \text{blur}}{\text{blur}_L} \\
\frac{\Gamma_{eo} [\text{od}_s(z)] \quad \text{id}}{\text{id}} & \quad \frac{\Gamma_{eo} \quad \text{blur}}{\text{blur}_L} \\
\frac{\Gamma_{eo} [\forall n. \text{even}(n) \supset \text{odd}(s(n))]}{\forall L} & \quad \frac{\Gamma_{eo} \quad \text{focus}}{\text{focus}_L} \\
\frac{\Gamma_{eo} [\forall n. \text{odd}(n) \supset \text{even}(s(n))]}{\forall L} & \quad \frac{\Gamma_{eo} \quad \text{focus}}{\text{focus}_L} \\
\frac{\Gamma_{eo} \quad \text{id}}{\text{id}} & \quad \frac{\Gamma_{eo} \quad \text{blur}}{\text{blur}_L}
\end{align*}
\]