MATH 3TP3 Assignment #8 Solutions

1. A PROP derivation would read like this:

$$
\begin{array}{ll}
[& \\
\quad P & \text{(push)} \\
\quad [& \\
\quad\quad \langle P \supset P' \rangle & \text{(push)} \\
\quad\quad P & \text{(carryover)} \\
\quad\quad P' & \text{(detachment)} \\
\quad] & \\
\quad \langle \langle P \supset P' \rangle \supset P' \rangle & \text{(pop)} \\
\quad] & \\
\langle P \supset \langle \langle P \supset P' \rangle \supset P' \rangle \rangle & \text{(pop)}
\end{array}
$$

We translate this into a derivation in this system S:

$$
\begin{array}{ll}
WFF : P & \\
WFF : P' & \\
WFF : \langle P \supset P' \rangle & \\
?P \vdash P & \text{(push)} \\
?P?\langle P \supset P' \rangle \vdash \langle P \supset P' \rangle & \text{(push)} \\
?P?\langle P \supset P' \rangle \vdash P & \text{(carryover)} \\
?P?\langle P \supset P' \rangle \vdash P' & \text{(detachment)} \\
?P \vdash \langle \langle P \supset P' \rangle \supset P' \rangle & \text{(pop)} \\
\vdash \langle P \supset \langle \langle P \supset P' \rangle \supset P' \rangle \rangle & \text{(pop)}
\end{array}
$$

So we can think of a string "$WFF : \phi$" as having the meaning that ϕ is a propositional wff, and "$?\phi?\psi \vdash \theta$" as corresponding to our mathematical notation $\{\phi, \psi\} \vdash \theta$, meaning that from premises ϕ and ψ we can deduce θ, corresponding to a line PROP derivation which reads "θ" and is in two fantasies, the outer with premise ϕ and the inner with premise ψ. Of course we can have more than two premises.

The production rules parallel the rules of PROP. If we added the other rules of PROP (joining, de Morgan etc) rather than just detachment, we would arrive at the Post formal system version of PROP described in the lecture notes. Doing “the same thing” for TNT, which involves some extra work since we have to handle the RESTRICTIONs in the quantifier manipulation rules, we obtain the system FormalTNT.

2. Let’s take a boring Gödel numbering:

$$\begin{array}{rcl} \mathtt{P} & \mapsto & 10 \\ \mathtt{'} & \mapsto & 11 \\ \supset & \mapsto & 12 \\ \langle & \mapsto & 13 \\ \rangle & \mapsto & 14 \\ \vdash & \mapsto & 15 \\ ? & \mapsto & 16 \\ \mathtt{W} & \mapsto & 17 \\ \mathtt{F} & \mapsto & 18 \\ \mathtt{F} & \mapsto & 19 \\ : & \mapsto & 20 \end{array}$$

(the only important point is that all numbers have the same length when written in base 10 (that’s why we started with 10 rather than 1); this is needed so we know how to read numbers as strings (here we write the number in base 10, then take pairs of numbers to correspond to symbols; if we’d had a symbol with Gödel number 1 but another with Gödel number 11, then we wouldn’t know whether 11 was coding one symbol or two))

We will show that there are formulas

$$\text{Produces}_I(x, z), \text{Produces}_{II}(x, y, z), \text{Produces}_{III}(x, y, z), \ldots, \text{Produces}_{VI}(x, y, z)$$

which express that a string can be produced by the production rules from given inputs, i.e. which are such that for $N = II, ..., VI$, if x and y are Gödel numbers of strings S_1 and S_2, then $\text{Produces}_N(x, y, z)$ is true (in $\mathbb{N}$) precisely when z is the Gödel number of a string which can be produced by rule N from inputs S_1 and S_2 (in that order); and similarly for $\text{Produces}_I(x, z)$.

Given this, we can define $\text{Theorem}_S(x)$ to be

$$\exists y : \text{ProofPair}(y, x),$$

where ProofPair(y, x) is the following formula expressing that y codes for a derivation of x:

$$
\begin{aligned}
&\exists z : \langle [y]_z = x \wedge \\
&\quad \forall z' : \langle z' \leq z \supset \\
&\qquad \langle\langle [y]_{z'} = \overline{\ulcorner \texttt{WFF:P} \urcorner} \vee [y]_{z'} = \overline{\ulcorner \vdash \urcorner} \rangle \vee \\
&\qquad\quad \exists z'' : \exists z''' : \langle z'' \langle z' \wedge z''' < z' \rangle \wedge \\
&\qquad\qquad \exists x' : \exists x'' : \exists x''' : \langle\langle [y]_{z'} = x' \wedge \langle [y]_{z''} = x'' \wedge [y]_{z'''} = x''' \rangle\rangle \wedge \\
&\qquad\qquad\quad \langle \mathrm{Produces}_I(x'', x') \vee \\
&\qquad\qquad\quad \langle \mathrm{Produces}_{II}(x'', x''', x') \vee \\
&\qquad\qquad\quad \langle \mathrm{Produces}_{III}(x'', x''', x') \vee \\
&\qquad\qquad\quad \langle \mathrm{Produces}_{IV}(x'', x''', x') \vee \\
&\qquad\qquad\quad \langle \mathrm{Produces}_{V}(x'', x''', x') \vee \\
&\qquad\qquad\quad \mathrm{Produces}_{VI}(x'', x''', x') \rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle
\end{aligned}
$$

(Yow!)

Here we're using the notation $[x]_y = z$ for ListElement(x, y, z), which recall was defined in terms of Gödel's β function.

It remains to see that there are the promised formulas Produces$_N$. I'll give them for III and VI; the others are similar but simpler.

First, it will reduce pain if we define some auxiliary formulas to handle concatenating many strings together. So define Concat$_3(x, x', x'', y)$ to be

$$\exists z : \langle \mathrm{Concat}(x, x', z) \wedge \mathrm{Concat}(z, x'', y) \rangle,$$

and define Concat$_4(x, x', x'', x''', y)$ to be

$$\exists z : \langle \mathrm{Concat}_3(x, x', x'', z) \wedge \mathrm{Concat}(z, x''', y) \rangle,$$

and so on.

Now let Produces$_{III}(x, y, z)$ be

$$
\begin{aligned}
&\exists x' : \exists y' : \exists z' : \langle\langle \mathrm{Concat}_3(x', \overline{\ulcorner \vdash \urcorner}, y', x) \wedge \\
&\qquad \mathrm{Concat}(\overline{\ulcorner \texttt{WFF:} \urcorner}, z', y) \rangle \wedge \mathrm{Concat}_5(x', \overline{\ulcorner ? \urcorner}, z', \overline{\ulcorner \vdash \urcorner}, z', z) \rangle\rangle
\end{aligned}
$$

and let Produces$_{VI}(x, y, z)$ be

$$\exists x' : \exists y' : \exists z' : \langle\langle \mathrm{Concat}_6(x', \overline{\ulcorner \vdash \langle \urcorner}, y', \overline{\ulcorner \supset \urcorner}, z', \overline{\ulcorner \rangle \urcorner}, x) \wedge \\ \mathrm{Concat}_3(x', \overline{\ulcorner \vdash \urcorner}, y', y)\rangle \wedge \mathrm{Concat}_3(x', \overline{\ulcorner \vdash \urcorner}, z', z)\rangle\rangle.$$

That's it! Painful, but essentially straightforward.