1 Termination for Gödel’s T

Recall the statics and dynamics for Gödel’s T:

\[ \Gamma, x : \tau \vdash x : \tau \quad \Gamma \vdash \lambda x : \tau_1. M : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash M : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash N : \tau_1 \]

\[ \Gamma \vdash M : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash MN : \tau_2 \]

\[ \Gamma \vdash z : \text{nat} \quad \Gamma \vdash M : \text{nat} \quad \Gamma \vdash s(M) : \text{nat} \quad \Gamma \vdash \text{natind}(M; N_0; x.N_1) : \tau \]

\[ \frac{M \rightarrow_{\beta} M'}{M N \rightarrow_{\beta} M' N} \quad \frac{M \text{ val}}{M N \rightarrow_{\beta} M' N'} \quad \frac{N \text{ val}}{(\lambda x : \tau. M) N \rightarrow_{\beta} [N/x]M} \quad \begin{array}{c} \text{natind}(z; N_0; x.N_1) \rightarrow_{\beta} N_0 \\ \text{natind}(s(M); N_0; x.N_1) \rightarrow_{\beta} \frac{\text{natind}(M; N_0; x.N_1)/x}{N_1} \\ M \rightarrow_{\beta} M' \\ s(M) \rightarrow_{\beta} M' \\ \text{natind}(M; N_0; x.N_1) \rightarrow_{\beta} \frac{\text{natind}(M'; N_0; x.N_1)}{\text{natind}(M'; N_0; x.N_1)} \\ M \text{ val} \\ s(M) \text{ val} \\ \lambda x : \tau_1. M \text{ val} \end{array} \]

In class, we proved the progress and preservation theorems for this system.

**Lemma 1.1** (Preservation). If \( \cdot \vdash M : \tau \) and \( M \rightarrow_{\beta} M' \) then \( \Gamma \vdash M' : \tau \).

**Lemma 1.2** (Progress). If \( \cdot \vdash M : \tau \) then either \( M \text{ val} \) or \( M \rightarrow_{\beta} M' \), for some \( M' \).

Use these theorems to prove type safety for Gödel’s T.

**Task 1** (Type Safety). Show that if \( \cdot \vdash M : \tau \) and \( M \rightarrow_{\beta} M' \), then \( M' \text{ val} \) or \( M' \rightarrow_{\beta} M'' \) for some \( M'' \).

Type safety ensures that well-typed terms reduce until they reach a value, but it’s possible that some term might never reach a value. In the next few tasks, we will prove termination, the property that every well-typed term reduces to a value.

We saw in class that a simple inductive proof of termination breaks down for application. Our fix will be to strengthen the inductive hypothesis to a property of terms called hereditary.
termination. Hereditary termination is defined at each type (inductively on the structure of types) as a predicate on closed terms of that type, as follows:

**Definition 1** (Hereditary Termination).

1. \( \text{HT}_{\text{nat}}(M) \) iff \( M \rightarrow^{*}_{\beta} N \) and \( N \) \val{}.

2. \( \text{HT}_{\tau_{1} \rightarrow \tau_{2}}(M) \) iff \( M \rightarrow^{*}_{\beta} \lambda x : \tau_{1}. M' \), and for any \( M_1 \) such that \( \text{HT}_{\tau_{1}}(M_1) \), \( \text{HT}_{\tau_{2}}([M_1/x]M') \).

For any context \( \Gamma \), a total substitution \( \gamma \) is a mapping from \( \text{dom}(\Gamma) \) to closed terms which preserves types; that is, for each \( x : \tau \) in \( \Gamma \), \( \gamma(x) : \tau \). Given a total substitution \( \gamma \) and a term \( \Gamma \vdash M : \tau \), we can form a closed term \( \hat{\gamma}(M) \) by performing the substitution. For example, if \( \Gamma = (x : \tau \rightarrow \tau', y : \tau) \), \( M = xy \), and \( \gamma \) is a total substitution for \( \Gamma \), then \( \hat{\gamma}(M) = \gamma(x) \gamma(y) \).

We say that \( \text{HT}_{\Gamma}(\gamma) \) iff for all \( x : \tau \) in \( \Gamma \), we have \( \text{HT}_{\tau_{x}}(\gamma(x)) \).

We need one more lemma before proving our main theorem.

**Task 2** (Head Expansion). Show that if \( \text{HT}_{\tau_{x}}(M), \cdot \vdash M' : \tau, \) and \( M' \rightarrow_{\beta} M \) then \( \text{HT}_{\tau_{x}}(M') \).

We can now prove all terms are hereditarily terminating. You may use without proof a canonical forms lemma.

**Task 3.** Show that if \( \Gamma \vdash M : \tau \) and \( \text{HT}_{\Gamma}(\gamma) \) then \( \text{HT}_{\tau_{x}}(\hat{\gamma}(M)) \).

(Hint) Use induction on the typing derivation. You will need an additional induction inside the case for the recursor.

**Task 4** (Termination). Show that if \( \cdot \vdash M : \tau \), then \( M \) terminates.

(Hint) Prove that if \( \text{HT}_{\tau_{x}}(M) \), then \( M \) terminates.

### 2 Big-step operational semantics

Because reduction in Gödel’s T is deterministic and terminating, we know that we can compute any term to its unique fully-evaluated form by reducing until we reach a value. Because we are describing computation by means of an iterated process of reduction, \( \rightarrow_{\beta} \) is called a small-step operational semantics.

Another approach is to directly define the relation between any term and its fully-evaluated form, all in one go; this is called a big-step operational semantics, and is written \( M \downarrow M' \). It is defined by the following rules:

\[
\begin{array}{c}
M \downarrow \lambda x : \tau. M' \\
N \downarrow N' \\
\text{natind}(M; N_0: x.N_1) \downarrow N'_0
\end{array} \quad \begin{array}{c}
M \downarrow M'' \\
N \downarrow N'' \\
\text{natind}(M'; N_0: x.N_1) \downarrow N'_1
\end{array} \quad \begin{array}{c}
M \downarrow s(M') \\
\text{natind}(M'; N_0: x.N_1) \downarrow N'_1
\end{array}
\]

We want to show that the notions of computation described by \( \downarrow \) and \( \rightarrow_{\beta} \) coincide, in the sense that they associate the same fully-evaluated form to each term.

As in the previous homework, you may use the fact that \( \rightarrow^{*}_{\beta} \) is transitive, and various compatibility lemmas (below).

**Lemma 2.1.** If \( M \rightarrow^{*}_{\beta} M' \) and \( M' \rightarrow^{*}_{\beta} M'' \) then \( M \rightarrow^{*}_{\beta} M'' \).

**Lemma 2.2.** If \( M \rightarrow^{*}_{\beta} M' \), then \( s(M) \rightarrow^{*}_{\beta} s(M') \).
Lemma 2.3. If $M \rightarrow^* M'$, then for any $x, N_0, N_1$, we have $\text{natind}(M; N_0; x. N_1) \rightarrow^* \text{natind}(M'; N_0; x. N_1)$.

Lemma 2.4. If $M \rightarrow^* M'$ then for any $M''$ we have $M M'' \rightarrow^* M' M''$.

Lemma 2.5. If $M \rightarrow^* M'$ then for any $M''$ we have $M'' M \rightarrow^* M'' M'$.

Lemma 2.6. If $M \rightarrow^* M'$ then for any $x$ we have $\lambda x. M \rightarrow^* \lambda x. M'$.

One direction of the correspondence between $\downarrow$ and $\rightarrow^*$ is the following:

Task 5. Show that if $M \downarrow M'$, then $M \rightarrow^* M'$ and $M'$ val.

To prove the other direction, we need two lemmas:

Task 6. Show that if $M \text{ val}$, then $M \downarrow M$.

Task 7. Show that if $M \rightarrow^* M'$ and $M' \downarrow M''$, then $M \downarrow M''$.

(Hint) Use induction on the derivation of $M \rightarrow^* M'$. In each case, consider which rules of $\downarrow$ could apply to $M'$.

And now we can complete the proof of the other direction:

Task 8. Show that if $M \rightarrow^* M'$ and $M'$ val, then $M \downarrow M'$.

3 Halting Problem in PCF

In this exercise we will show that there is no PCF term $T$ which, given a PCF term $M : \text{nat}$, decides whether or not $M$ reduces to a value. Such a term $T : \text{nat} \rightarrow \text{nat}$ must obey the following specification:

$$
T M \rightarrow^* 0 \quad \text{or} \quad T M \rightarrow^* 1 \\
T M \rightarrow^* 1 \quad \text{iff} \ M \text{ reduces to a value (converges)} \\
T M \rightarrow^* 0 \quad \text{iff} \ M \text{ does not reduce to a value (diverges)}
$$

To prove that no such $T$ is definable in the $\lambda$-calculus, we will replicate the diagonal argument used to prove the undecidability of the Halting Problem, but in a higher-order setting.

Task 9. Use the fix operator to write a divergent closed term of type $\text{nat}$.

Task 10. Show that $T$ is not definable in PCF.

(Hint) Assume that $T$ is definable, and define another term $D : \text{nat}$ that applies $T$ to $D$ itself, such that if the outcome is 1 then $D$ diverges, and otherwise it is equal to 0. (Use $\text{natrec}$ to check the value of $TD$.) Since $D$ is a PCF term, we can try and observe the behavior of applying $T$ to $D$ itself, from which a contradiction should arise.