

Homework 3: Termination Proofs using Logical Relations, Recursive Types

15-814: Type Systems for Programming Languages

TA: Kumar Avijit (kavijit@cs.cmu.edu)

Out: October 2, 2009

Due: October 9, 2009 (11:59 PM)

1 Proving Termination

In this section, we will revisit the proof of termination of weak-head reduction for $\lambda^{\rightarrow \times}$ from the class, and augment the proof for sums.

1.1 Positive and negative interpretations

Consider $\lambda^{\rightarrow \times}$ from Figures 1, 2 in Appendix. The termination theorem states

Theorem 1 (Termination). *If $\cdot \vdash m : A$, then $m \downarrow$, where $m \downarrow$ is defined as $m \rightarrow_{\text{wh}}^* n \not\rightarrow_{\text{wh}}$.*

In the class, we noticed that we cannot prove this theorem directly by induction on typing because our induction hypothesis is too weak. The way out was to define a type-indexed family of predicates $\text{HT}_A(\cdot)$ by induction on structure of type A , such that the following hold:

1. If $\cdot \vdash m : A$ then $\text{HT}_A(m)$.
2. If $\text{HT}_A(m)$ then $m \downarrow$.

In the class, we used a “negative” definition of $\text{HT}_A(\cdot)$. We shall use the notation $\text{HT}_A^-(m)$ to denote this definition:

- $\text{HT}_{A_1 \rightarrow A_2}^-(m)$ iff for all m_1 , $\text{HT}_{A_1}^-(m_1)$ implies $\text{HT}_{A_2}^-(\text{ap}(m; m_1))$.
- $\text{HT}_{A_1 \times A_2}^-(m)$ iff $\text{HT}_{A_1}^-(\text{fst}(m))$ and $\text{HT}_{A_2}^-(\text{snd}(m))$.

Alternatively, we can give a definition of hereditary termination using “positive” interpretation, which is given as follows:

- $\text{HT}_{A_1 \rightarrow A_2}^+(m)$ iff $m \rightarrow_{\text{wh}}^* \lambda x:A_1.m_2$ and $\text{HT}_{A_1}^+(m_1)$ implies $\text{HT}_{A_2}^+([m_1/x]m_2)$.
- $\text{HT}_{A_1 \times A_2}^+(m)$ iff $m \rightarrow_{\text{wh}}^* \langle m_1; m_2 \rangle$ and $\text{HT}_{A_1}^+(m_1)$, and $\text{HT}_{A_2}^+(m_2)$.

Task 1. *Prove that if $m : A$ then $\text{HT}_A^-(m)$ iff $\text{HT}_A^+(m)$.*

1.2 Termination for sums

Now we will extend the calculus with sums and prove termination for the whole system. We add the following new types and terms:

Types $A ::= \dots \mid \mathbf{0} \mid A_1 + A_2$
 Terms $m ::= \dots \mid \text{abort}_A(m) \mid \text{inl}_{A_1, A_2}(m) \mid \text{inr}_{A_1, A_2}(m) \mid \text{case } m\{\text{inl } x \Rightarrow m_1 \mid \text{inr } y \Rightarrow m_2\}$

The typing rules corresponding to the new types are given below:

$$\frac{\Gamma \vdash m : \mathbf{0}}{\Gamma \vdash \text{abort}_A(m) : A} (\mathbf{0}\text{-E}) \quad \frac{\Gamma \vdash m : A_1}{\Gamma \vdash \text{inl}_{A_1, A_2}(m) : A_1 + A_2} (+\text{-I}_1)$$

$$\frac{\Gamma \vdash m : A_2}{\Gamma \vdash \text{inr}_{A_1, A_2}(m) : A_1 + A_2} (+\text{-I}_2)$$

$$\frac{\Gamma \vdash m : A_1 + A_2 \quad \Gamma, x:A_1 \vdash m_1 : A \quad \Gamma, y:A_2 \vdash m_2 : A}{\Gamma \vdash \text{case } m\{\text{inl } x \Rightarrow m_1 \mid \text{inr } y \Rightarrow m_2\} : A} (+\text{-E})$$

Weak-head reduction for sums and abort are defined as follows:

$$\frac{}{\text{case } \text{inl}_{A_1, A_2}(m)\{\text{inl } x \Rightarrow m_1 \mid \text{inr } y \Rightarrow m_2\} \rightsquigarrow [m/x]m_1} (+ \rightsquigarrow_1)$$

$$\frac{}{\text{case } \text{inr}_{A_1, A_2}(m)\{\text{inl } x \Rightarrow m_1 \mid \text{inr } y \Rightarrow m_2\} \rightsquigarrow [m/y]m_2} (+ \rightsquigarrow_2)$$

$$\frac{m \rightarrow_{\text{wh}} m'}{\text{case } m\{\text{inl } x \Rightarrow m_1 \mid \text{inr } y \Rightarrow m_2\} \rightarrow_{\text{wh}} \text{case } m'\{\text{inl } x \Rightarrow m_1 \mid \text{inr } y \Rightarrow m_2\}} \text{case } \rightarrow_{\text{wh}}$$

We define the hereditary termination predicate for $\mathbf{0}$ and $A_1 + A_2$ as follows:

- $\text{HT}_{\mathbf{0}}(m)$ for no m .
- $\text{HT}_{A_1 + A_2}(m)$ iff either
 - $m \rightarrow_{\text{wh}}^* \text{inl}_{A_1, A_2}(m')$ and $\text{HT}_{A_1}(m')$, or
 - $m \rightarrow_{\text{wh}}^* \text{inr}_{A_1, A_2}(m')$ and $\text{HT}_{A_2}(m')$.

Task 2 (Fundamental Lemma). *Prove that if $\Gamma \vdash m : A$ and $\text{HT}_{\Gamma}(\gamma)$, where γ assigns closed terms to all variables in Γ , then $\text{HT}_A(\widehat{\gamma}(m))$.*

The proof proceeds by induction on derivation of $\Gamma \vdash m : A$. We proved this lemma for $\lambda^{\rightarrow \times}$ in the class. Here you need to show only the cases corresponding to A being $\mathbf{0}$ and $A_1 + A_2$.

At a stage in the proof, you will need a head-expansion lemma, which states that hereditary substitution is preserved under reverse head-reduction (or head-expansion). You will prove this lemma separately in Task 3.

Task 3 (Head-expansion Lemma). *Prove that if $\text{HT}_A(m)$ and $m' \rightarrow_{\text{wh}} m$ then $\text{HT}_A(m')$. (Hint) Begin by induction on the type A . You only need to show cases for $A = \mathbf{0}, A_1 + A_2$.*

Task 4 (Typing implies hereditary termination). *Prove that if $m : A$ then $\text{HT}_A(m)$. Proceed by induction on typing of m . You only need to show the cases corresponding to intro- and elim- rules for $\mathbf{0}, A_1 + A_2$. You can use Head-expansion Lemma and Fundamental Lemma here.*

Lemma 1 (Hereditary termination implies termination). *If $\text{HT}_A(m)$ then $m \downarrow$.*

Proof. Here we only show the cases for $A = \mathbf{0}, A_1 + A_2$.

Case: $A = \mathbf{0}$: This case never arises because of definition of $\text{HT}_{\mathbf{0}}(\cdot)$.

Case: $A = A_1 + A_2$: From the definition of $\text{HT}_{A_1+A_2}(m)$, we get the subcases:

Subcase: $m \rightarrow_{\text{wh}}^* \text{inl}_{A_1, A_2}(m_1)$, in which case $\text{inl}_{A_1, A_2}(m_1) \downarrow$.

Subcase: $m \rightarrow_{\text{wh}}^* \text{inr}_{A_1, A_2}(m_2)$, in which case $\text{inr}_{A_1, A_2}(m_2) \downarrow$.

□

Lemma 1 and Task 4 together prove termination.

2 Infinite Streams

2.1 Definition

Let us consider an extension to \mathbf{T} using a type of streams of natural numbers.

Types $A ::= \dots \mid \text{stream}$
 Terms $m ::= \dots \mid \text{hd}(m) \mid \text{tl}(m) \mid \text{corec } m\{\text{hd} \Rightarrow h \ \& \ \text{tl} \Rightarrow t\}$

A stream is a data-structure defined “co-inductively” using two elimination functions:

$$\frac{\Gamma \vdash m : \text{stream}}{\Gamma \vdash \text{hd}(m) : \omega} (\text{stream-E}_{\text{hd}}) \qquad \frac{\Gamma \vdash m : \text{stream}}{\Gamma \vdash \text{tl}(m) : \text{stream}} (\text{stream-E}_{\text{tl}})$$

An intuitive difference between inductive and co-inductive definitions is that while an inductive definition defines all possible ways to *generate* elements, and an elimination form just case-analyses on the ways, a co-inductive definition defines all possible elimination forms. Given a stream, one can project a head or a tail. Given a head function h and a tail function t , an intro form just provides an initial element for the stream:

$$\frac{\Gamma \vdash m_s : A \quad \Gamma \vdash h : A \rightarrow \omega \quad \Gamma \vdash t : A \rightarrow A}{\Gamma \vdash \text{corec } m_s\{\text{hd} \Rightarrow h \ \& \ \text{tl} \Rightarrow t\} : \text{stream}} (\text{stream-I})$$

The term m_s is the *seed* element. The head and tail of the stream are obtained by applying h and t resp. to the seed element.

The reduction steps are defined as follows:

$$\frac{}{\text{hd}(\text{corec } m\{\text{hd} \Rightarrow h \ \& \ \text{tl} \Rightarrow t\}) \rightsquigarrow \text{ap}(h; m)} (\text{stream} \rightsquigarrow_{\text{hd}})$$

$$\frac{}{\text{tl}(\text{corec } m\{\text{hd} \Rightarrow h \ \& \ \text{tl} \Rightarrow t\}) \rightsquigarrow \text{corec } \text{ap}(t; m)\{\text{hd} \Rightarrow h \ \& \ \text{tl} \Rightarrow t\}} \text{stream} \rightsquigarrow_{\text{tl}}$$

$$\frac{m \rightarrow_{\text{wh}} m'}{\text{hd}(m) \rightarrow_{\text{wh}} \text{hd}(m')} (\text{hd} \rightarrow_{\text{wh}}) \qquad \frac{m \rightarrow_{\text{wh}} m'}{\text{tl}(m) \rightarrow_{\text{wh}} \text{tl}(m')} (\text{tl} \rightarrow_{\text{wh}})$$

Task 5. Define a stream of natural numbers using this extension of \mathbf{T} . That is give a term m such that $\text{hd}(m) \rightarrow_{\text{wh}}^* z$, $\text{hd}(\text{tl}(m)) \rightarrow_{\text{wh}}^* s(z)$, $\text{hd}(\text{tl}(\text{tl}(m))) \rightarrow_{\text{wh}}^* s(s(z))$,

2.2 Termination for streams

We will now try to prove that all terms in the calculus $\mathbf{T} + \text{stream}$ terminate. We specify the hereditary termination predicate for streams using its elimination forms, co-inductively as follows:

$\text{HT}_{\text{stream}}(\cdot)$ is the *largest* predicate $P(\cdot)$ such that the following two conditions hold:

- **C1:** $P(m)$ implies $\text{HT}_{\omega}(\text{hd}(m))$
- **C2:** $P(m)$ implies $P(\text{tl}(m))$

Task 6 (Fundamental Lemma). Prove that if $\Gamma \vdash m : A$ and $\text{HT}_{\Gamma}(\gamma)$, then $\text{HT}_A(\hat{\gamma}(m))$.

The proof proceeds by induction on derivation of $\Gamma \vdash m : A$. You only need to show cases corresponding to Rules (stream-I), (stream-E₁) and (stream-E₂). We define $\hat{\gamma}(\text{corec } m\{\text{hd} \Rightarrow h \ \& \ \text{tl} \Rightarrow t\})$ as $\text{corec } \hat{\gamma}(m)\{\text{hd} \Rightarrow \hat{\gamma}(h) \ \& \ \text{tl} \Rightarrow \hat{\gamma}(t)\}$, $\hat{\gamma}(\text{hd}(m))$ as $\text{hd}(\hat{\gamma}(m))$, and $\hat{\gamma}(\text{tl}(m))$ as $\text{tl}(\hat{\gamma}(m))$.

Task 7 (Head expansion). Prove that if $m \rightarrow_{\text{wh}} m'$ and $\text{HT}_{\text{stream}}(m')$, then $\text{HT}_{\text{stream}}(m)$.

Task 8. Prove that if $m : A$ then $\text{HT}_A(m)$.

Here again, you need only show the cases corresponding to Rules (stream-I), (stream-E₁) and (stream-E₂).

(Hint) The main exercise in the above tasks is the proof of $\text{HT}_{\text{stream}}(m)$. The predicate $\text{HT}_{\text{stream}}(\cdot)$ is defined co-inductively, i.e. it is the largest predicate that satisfies conditions C1 and C2. In order to show $\text{HT}_{\text{stream}}(m)$ for a term m , it suffices to show $P(m)$ for some property P that satisfies C1 and C2. For instance, consider head-expansion lemma. Suppose $m \rightarrow_{\text{wh}} n$ and $\text{HT}_{\text{stream}}(n)$. In order to show $\text{HT}_{\text{stream}}(m)$, you may consider P to be the predicate $P(m) = \exists n. (m \rightarrow_{\text{wh}} n \wedge \text{HT}_{\text{stream}}(n))$.

3 Mutual recursion

In this part, we will work with recursive types. As we saw in the class, the fixed-point operator `fix` from Plotkin's PCF is definable using recursive types. We shall however continue to use `fix x:A in m` as a syntactic sugar. The language we are considering here is:

$$\begin{array}{l} \text{Types } A ::= \dots \mid X \mid \mu X.A \\ \text{Terms } m ::= \dots \mid \text{fold } m \mid \text{unfold } m \mid \text{fix } x:A \text{ in } m \end{array}$$

where \dots denotes the fragment $\lambda^{1 \rightarrow \times^+}$. Refer to the lecture on PCF and recursive types for the static and dynamic semantics.

As done in the class, we define $\omega = \mu X.1+X$, and z and $s(m)$ as `fold in1,ω(⟨⟩)` and `fold inr,ω(m)` resp. We wish to define two mutually recursive functions `even` and `odd`, using the informal inductive definition:

$$\begin{array}{ll} \text{even}(0) & = \text{true} \\ \text{even}(s(x)) & = \text{odd}(x) \\ \\ \text{odd}(0) & = \text{false} \\ \text{odd}(s(x)) & = \text{even}(x) \end{array}$$

Task 9. Write down a term m of type $(\omega \rightarrow \text{bool}) \times (\omega \rightarrow \text{bool})$, such that `fst(m)` implements the function `even`, and `snd(m)` implements the function `odd`, as per the above definition. You are allowed to use `true` and `false` as constant terms since they are definable in the above language.

A Appendix

$$\begin{array}{c}
\frac{}{\Gamma, x:A \vdash x : A} \text{(hyp)} \qquad \frac{\Gamma, x:A \vdash m : A'}{\Gamma \vdash \lambda x:A. m : A \rightarrow A'} (\rightarrow\text{-I}) \\
\frac{\Gamma \vdash m_2 : A_1 \rightarrow A_2 \quad \Gamma \vdash m_1 : A_1}{\Gamma \vdash \text{ap}(m_2; m_1) : A_2} (\rightarrow\text{-E}) \qquad \frac{\Gamma \vdash m_1 : A_1 \quad \Gamma \vdash m_2 : A_2}{\Gamma \vdash \langle m_1; m_2 \rangle : A_1 \times A_2} (\times\text{-I}) \\
\frac{\Gamma \vdash m : A_1 \times A_2}{\Gamma \vdash \text{fst}(m) : A_1} (\times\text{-E}_1) \qquad \frac{\Gamma \vdash m : A_1 \times A_2}{\Gamma \vdash \text{snd}(m) : A_2} (\times\text{-E}_2)
\end{array}$$

Figure 1: Type system for $\lambda^{\rightarrow \times}$

$$\begin{array}{c}
\frac{}{\text{ap}(\lambda x:A. m; n) \rightsquigarrow [n/x]m} (\rightarrow \rightsquigarrow) \qquad \frac{}{\text{fst}(\langle m_1; m_2 \rangle) \rightsquigarrow m_1} (\times \rightsquigarrow_1) \\
\frac{}{\text{snd}(\langle m_1; m_2 \rangle) \rightsquigarrow m_2} (\times \rightsquigarrow_2)
\end{array}$$

$$\begin{array}{c}
\frac{m \rightsquigarrow n}{m \rightarrow_{\text{wh}} n} \text{(step)} \qquad \frac{m \rightarrow_{\text{wh}} m'}{\text{ap}(m; n) \rightarrow_{\text{wh}} \text{ap}(m'; n)} \text{(app} \rightarrow_{\text{wh}}) \\
\frac{m \rightarrow_{\text{wh}} m'}{\text{fst}(m) \rightarrow_{\text{wh}} \text{fst}(m')} \text{(fst} \rightarrow_{\text{wh}}) \qquad \frac{m \rightarrow_{\text{wh}} m'}{\text{snd}(m) \rightarrow_{\text{wh}} \text{snd}(m')} \text{(snd} \rightarrow_{\text{wh}})
\end{array}$$

Figure 2: Weak-head reduction for $\lambda^{\rightarrow \times}$ -calculus