# Deciding $\beta\eta$-Equivalence for Product and Function Types

Robert Harper

April 12, 2021

## 1 Introduction

Formal equality for product and function types is expressed by the inductively defined judgment $\Gamma \vdash M \equiv M' : A$, called *definitional equivalence*, which is defined for terms $\Gamma \vdash M, M' : A$. Because products and functions have negative polarity, it is natural to postulate their full universal properties, which are expressed by including $\beta$ rules to express computation and $\eta$ rules to express the unicity conditions required to fully determine the elements of a type. To avoid degeneracy, a type of binary answers ("accept/reject") is included. In a fuller type system this requirement would not be necessary, but would introduce complications that are avoided here. The rules defining formal equality are given in Figure 1.

The question considered here is whether definitional equivalence for product and function types is decidable.[1] The algorithm, see Figure 2, is given as a collection of rules defining the main judgment, $M \Leftrightarrow M' \downarrow A\ [\Delta]$, and its auxiliary, $U \leftrightarrow U' \uparrow A\ [\Delta]$. All arguments to the main judgment are regarded as inputs; the terms $M$ and $M'$ are deemed equivalent at type $A$ relative to context $\Delta$ exactly when it is derivable. The context and neutral terms $U$ and $U'$ are regarded as inputs to the auxiliary judgment, and the type $A$ as its output; the neutral terms are deemed equivalent relative to context $\Delta$ when there is a type $A$ for which this judgment is derivable.

These rules constitute an algorithm in that it is apparent that, for the given inputs, it is straightforward to determine whether or not the required derivation exists. In the case of the main judgment termination is proved by induction on the structure of the type $A$, and in the auxiliary case it is proved by induction on the neutral terms. The correctness of the algorithm is stated by two theorems, often called the *soundness* and *completeness* properties.

**Theorem 1** (Correctness)**.** *1. If $M \Leftrightarrow M' \downarrow A\ [\Gamma]$, then $\Gamma \vdash M \equiv M' : A$.*

*2. If $\Gamma \vdash M \equiv M' : A$, then $M \Leftrightarrow M' \downarrow A\ [\Gamma]$.*

The first of these, soundness, is proved by induction on the derivation of the algorithmic judgment, with a similar property of the auxiliary, with which it is recursive, being proved simultaneously. It is only necessary to show that definitional equivalence is closed under well-typed head expansion, the rest being a straightforward induction on derivations.

**Exercise 1.** *State and prove the soundness of the equivalence checking algorithm defined in Figure 2 relative to definitional equivalence as defined in 1.*

[1] The interest in this question arises in dependent type systems for which decidability of type checking is reducible to decidability of equivalence, albeit for a richer class of types.

$$\frac{}{\Gamma \vdash M \equiv M : A}\ \text{REFL} \qquad \frac{\Gamma \vdash M \equiv M' : A}{\Gamma \vdash M' \equiv M : A}\ \text{SYM} \qquad \frac{\Gamma \vdash M \equiv M' : A \quad \Gamma \vdash M' \equiv M'' : A}{\Gamma \vdash M \equiv M'' : A}\ \text{TRANS} \qquad \frac{}{\Gamma \vdash M \equiv \ : \mathbf{1}}\ \mathbf{1}\text{-}\eta$$

$$\frac{\Gamma \vdash M_1 \equiv M_1' : A_1 \quad \Gamma \vdash M_2 \equiv M_2' : A_2}{\Gamma \vdash \langle M_1 \,, M_2 \rangle \equiv \langle M_1' \,, M_2' \rangle : A_1 \times A_2}\ \times\text{-I} \qquad \frac{\Gamma \vdash M \equiv M' : A_1 \times A_2}{\Gamma \vdash M \cdot 1 \equiv M' \cdot 1 : A_1}\ \times\text{-E-L} \qquad \frac{\Gamma \vdash M \equiv M' : A_1 \times A_2}{\Gamma \vdash M \cdot 2 \equiv M' \cdot 2 : A_1}\ \times\text{-E-R}$$

$$\frac{}{\Gamma \vdash \langle M_1 \,, M_2 \rangle \cdot 1 \equiv M_1 : A_1}\ \times\text{-}\beta\text{-L} \qquad \frac{}{\Gamma \vdash \langle M_1 \,, M_2 \rangle \cdot 2 \equiv M_2 : A_2}\ \times\text{-}\beta\text{-R} \qquad \frac{}{\Gamma \vdash M \equiv \langle M \cdot 1 \,, M \cdot 2 \rangle : A_1 \times A_2}\ \times\text{-}\eta$$

$$\frac{\Gamma, x : A_1 \vdash M_2 \equiv M_2' : A_2}{\Gamma \vdash \lambda_{A_1}(x \,.\, M_2) \equiv \lambda_{A_1}(x \,.\, M_2') : A_1 \to A_2}\ \to\text{-I} \qquad \frac{\Gamma \vdash M \equiv M' : A_1 \to A_2 \quad \Gamma \vdash M_1 \equiv M_1' : A_1}{\Gamma \vdash \mathsf{ap}(M, M_1) \equiv \mathsf{ap}(M', M_1') : A_2}\ \to\text{-E}$$

$$\frac{}{\Gamma \vdash \mathsf{ap}(\lambda_{A_1}(x \,.\, M_2), M_1) \equiv [M_1/x]M_2 : A_2}\ \to\text{-}\beta \qquad \frac{}{\Gamma \vdash M \equiv \lambda_{A_1}(x \,.\, \mathsf{ap}(M, x)) : A_1 \to A_2}\ \to\text{-}\eta$$

Figure 1: Definitional Equivalence

The second, completeness, requires a more substantial argument based on Tait's method, which is the subject of the main body of this note.

## 2 Completeness Proof

As might be surmised from Harper (2021), the proof of completeness breaks down into two parts, mediated by *logical equivalence* as defined in Figure 3.

**Lemma 2** (Pas-de-deux)**.** *1. If $U \leftrightarrow U' \uparrow A\ [\Delta]$, then $U = U' \in A\ [\Delta]$.*

*2. If $M = M' \in A\ [\Delta]$, then $M \Leftrightarrow M' \downarrow A\ [\Delta]$.*

*Proof.* Simultaneously, by induction on $A$.

1. $A = \mathbf{1}$:
   (a) Immediate, by definition of logical equivalence.
   (b) Immediate, by definition of algorithmic equivalence.
2. $A = \mathbf{2}$:
   (a) By definition of algorithmic equivalence.
   (b) By definition of logical equivalence.

$$\frac{}{M \Leftrightarrow M' \downarrow \mathbf{1}\ [\Delta]}\ \textbf{1}\text{-EQ} \qquad \frac{M \mapsto_\beta^* U \qquad M' \mapsto_\beta^* U' \qquad U \leftrightarrow U' \uparrow \mathbf{2}\ [\Delta]}{M \Leftrightarrow M' \downarrow \mathbf{2}\ [\Delta]}\ \textbf{2}\text{-EQ}$$

$$\frac{M \cdot 1 \Leftrightarrow M' \cdot 1 \downarrow A_1\ [\Delta] \qquad M \cdot 2 \Leftrightarrow M' \cdot 2 \downarrow A_2\ [\Delta]}{M \Leftrightarrow M' \downarrow A_1 \times A_2\ [\Delta]}\ \times\text{-EQ} \qquad \frac{\mathsf{ap}(M, x) \Leftrightarrow \mathsf{ap}(M', x) \downarrow A_2\ [\Delta, x : A_1]}{M \Leftrightarrow M' \downarrow A_1 \to A_2\ [\Delta]}\ \to\text{-EQ}$$

$$\frac{}{x \leftrightarrow x \uparrow A\ [\Delta, x : A]}\ \text{VAR} \qquad \frac{}{\mathsf{yes} \leftrightarrow \mathsf{yes} \uparrow \mathbf{2}\ [\Delta]}\ \text{YES} \qquad \frac{}{\mathsf{no} \leftrightarrow \mathsf{no} \uparrow \mathbf{2}\ [\Delta]}\ \text{NO}$$

$$\frac{U \leftrightarrow U' \uparrow A_1 \times A_2\ [\Delta]}{U \cdot 1 \leftrightarrow U' \cdot 1 \uparrow A_1\ [\Delta]}\ \text{LFT} \qquad \frac{U \leftrightarrow U' \uparrow A_1 \times A_2\ [\Delta]}{U \cdot 2 \leftrightarrow U' \cdot 2 \uparrow A_1\ [\Delta]}\ \text{RHT}$$

$$\frac{U \leftrightarrow U' \uparrow A_1 \to A_2\ [\Delta] \qquad M_1 \Leftrightarrow M_1' \downarrow A_1\ [\Delta]}{\mathsf{ap}(U, M_1) \leftrightarrow \mathsf{ap}(U', M_1') \uparrow A_2\ [\Delta]}\ \text{APP}$$

Figure 2: Algorithmic Equality

$$\begin{aligned}
M = M' \in \mathbf{1}\ [\Delta] &\iff \text{(true)} \\
M = M' \in \mathbf{2}\ [\Delta] &\iff M \Leftrightarrow M' \downarrow \mathbf{2}\ [\Delta] \\
M = M' \in A_1 \times A_2\ [\Delta] &\iff M \cdot 1 = M' \cdot 1 \in A_1\ [\Delta] \text{ and } M \cdot 2 = M' \cdot 2 \in A_1\ [\Delta] \\
M = M' \in A_1 \to A_2\ [\Delta] &\iff \forall \Delta' \leq \Delta \text{ if } M_1 = M_1' \in A_1\ [\Delta'] \text{ then } \mathsf{ap}(M, M_1) = \mathsf{ap}(M', M_1') \in A_2\ [\Delta']
\end{aligned}$$

Figure 3: Logical Equivalence

3. $A = A_1 \times A_2$:

   (a) If $U' \leftrightarrow A \uparrow U$ $[\Delta]$, then $U' \cdot \mathsf{1} \leftrightarrow A_1 \uparrow U \cdot \mathsf{1}$ $[\Delta]$ and $U' \cdot \mathsf{2} \leftrightarrow A_2 \uparrow U \cdot \mathsf{2}$ $[\Delta]$. By induction $U' \cdot \mathsf{1} = A_1 \in U \cdot \mathsf{1}$ $[\Delta]$ and $U' \cdot \mathsf{2} = A_2 \in U \cdot \mathsf{2}$ $[\Delta]$, and hence $U' = A \in U$ $[\Delta]$, as desired.

   (b) Similarly, $M' \cdot \mathsf{1} = A_1 \in M \cdot \mathsf{1}$ $[\Delta]$ and $M' \cdot \mathsf{2} = A_2 \in M \cdot \mathsf{2}$ $[\Delta]$, and hence by induction $M' \cdot \mathsf{1} \Leftrightarrow A_1 \downarrow M \cdot \mathsf{1}$ $[\Delta]$ and $M' \cdot \mathsf{2} \Leftrightarrow A_2 \downarrow M \cdot \mathsf{2}$ $[\Delta]$, from which the result follows directly.

4. $A = A_1 \to A_2$:

   (a) Suppose that $U' \leftrightarrow A \uparrow U$ $[\Delta]$, that $\Delta' \leq \Delta$, and that $M_1' = A_1 \in M_1$ $[\Delta']$. By inductive hypothesis $M_1' \Leftrightarrow A_1 \downarrow M_1$ $[\Delta']$, and thus $\mathsf{ap}(U', M') \leftrightarrow A_2 \uparrow \mathsf{ap}(U, M)$ $[\Delta']$. But then by induction $\mathsf{ap}(U', M') = A_2 \in \mathsf{ap}(U, M)$ $[\Delta']$, as desired.

   (b) Let $\Delta' \triangleq \Delta, x : A_1$, noting that $\Delta' \leq \Delta$, and that $x \leftrightarrow A_1 \uparrow x$ $[\Delta']$. But then by induction $x = A_1 \in x$ $[\Delta']$, and so $\mathsf{ap}(M', x) = A_2 \in \mathsf{ap}(M, x)$ $[\Delta']$. By induction $\mathsf{ap}(M', x) \Leftrightarrow A_2 \downarrow \mathsf{ap}(M, x)$ $[\Delta']$, and so $M' \Leftrightarrow A_1 \to A_2 \downarrow M$ $[\Delta]$.

□

**Corollary 3.** *For id the identity substitution,* $id = id \in \Gamma$ $[\Gamma]$.

**Lemma 4** (Fundamental Lemma)**.**

1. *If* $\Gamma \vdash M : A$ *and* $\gamma = \gamma' \in \Gamma$ $[\Delta]$*, then* $\gamma^*(M) = \gamma'^*(M) \in A$ $[\Delta]$.

2. *If* $\Gamma \vdash M \equiv M' : A$ *and* $\gamma = \gamma' \in \Gamma$ $[\Delta]$*, then* $\gamma^*(M) = \gamma'^*(M') \in A$ $[\Delta]$.

*Proof.* By induction on the structure of derivations of definitional equivalence. □

**Exercise 2.** *Prove Lemma 4.*

**Corollary 5.** *If* $\Gamma \vdash M \equiv M' : A$*, then* $M \Leftrightarrow M' \downarrow A$ $[\Gamma]$.

*Proof.* By Lemma 4 and Corollary 3. □

## References

Robert Harper. Kripke-style logical relations for termination. Unpublished lecture note, Spring 2021. URL https://www.cs.cmu.edu/~rwh/courses/chtt/pdfs/kripke.pdf.