Semantic Equality for Typed λ-Calculus*

Robert Harper

Spring 2024

1 Introduction

The unary logical relations developed in Harper (2024) may be extended from unary predicates to binary relations. In binary form these relations define *exact equality* at each type. Unlike axiomatic accounts (those given by rules) exact equality defines when two terms of a type are *semantically equal*. For example, the "add-to-self" and "doubling" functions on the natural numbers are exactly equal, because they have the same I/O behavior. This formulation—the standard one for equality of functions—is called *extensional equality*.

This note defines exact equality for terms of each type, and establishes some basic properties of it, in particular that it is an equivalence relation that is compatible with the term-forming operations, and that it respects—and thus contains—evaluation.

2 Exact Equality

The definition of exact equality is very similar to the definition of hereditary termination given in Harper (2024).

$$M \doteq M' \in \mathsf{ans} \text{ iff } M, M' \longmapsto^* \mathsf{yes} \text{ or } M, M' \longmapsto^* \mathsf{no}$$

$$M \doteq M' \in \mathsf{unit} \text{ iff } M, M' \longmapsto^* \langle\rangle$$

$$M \doteq M' \in A_1 \times A_2 \text{ iff } M \longmapsto^* \langle M_1, M_2\rangle, M' \longmapsto^* \langle M_1', M_2'\rangle, \text{ and}$$
$$M_1 \doteq M_1' \in A_1 \text{ and } M_2 \doteq M_2' \in A_2$$

$$M \doteq M' \in A_1 \to A_2 \text{ iff } M \longmapsto^* \lambda(x.N), M' \longmapsto^* \lambda(x.N'), \text{ and}$$
$$\text{if } M_1 \doteq M_1' \in A_1 \text{ then } [M_1/x]N \doteq [M_1'/x]N' \in A_2$$

The judgment $M \in A$ is defined to mean $M \doteq M \in A$.

If exact equality is to be so-called, it ought to be symmetric and transitive.

Lemma 1. *1. If $M \doteq M' \in A$ then $M' \doteq M \in A$.*

2. *If* $M \doteq M' \in A$ *and* $M' \doteq M'' \in A$, *then* $M \doteq M'' \in A$.

Proof. By induction on A. Consider the case $A = A_1 \to A_2$.

1. Suppose that $M \doteq M' \in A$ with the goal to show that $M' \doteq M \in A$. By assumption $M \longmapsto^* \lambda(x.N)$ and $M' \longmapsto^* \lambda(x.N')$. Assume that $M_1' \doteq M_1 \in A_1$, with the intent to show that $[M_1'/x]N' \doteq [M_1/x]N \in A_2$. A direct application of the outer assumption yields $[M_1/x]N' \doteq [M_1'/x]N \in A_2$, which is not what is required. However, exact equality at both A_1 and A_2 is symmetric. First, appealing to symmetry at A_1, from the assumption $M_1' \doteq M_1 \in A_1$ it follows that $M_1 \doteq M_1' \in A_1$, and hence by the outer assumption $[M_1/x]N \doteq [M_1'/x]N' \in A_2$. Then, applying symmetry at A_2, the desired result follows.

2. Suppose that $M \doteq M' \in A$ and $M' \doteq M'' \in A$ with the goal to show that $M \doteq M'' \in A$. By the definition of exact equality at function type, the two assumptions imply that $M \longmapsto^* \lambda(x.N)$, $M' \longmapsto^* \lambda(x.N')$, and $M'' \longmapsto^* \lambda(x.N'')$. Now suppose that $M_1 \doteq M_1'' \in A_1$ with the intent to show that $[M_1/x]N \doteq [M_1''/x]N'' \in A_2$. Here again a direct application of the outer assumptions does not seem to help, obtaining

   (a) $[M_1/x]N \doteq [M_1''/x]N' \in A_2$, and

   (b) $[M_1/x]N' \doteq [M_1''/x]N'' \in A_2$.

   Note that a symmetric and transitive relation is reflexive on related elements: if $R(M, M')$ then $R(M', M)$, and so $R(M, M)$ and $R(M', M')$. By the inductive assumptions equality at type A_1 is symmetric and transitive, and so $M \doteq M \in A_1$ follows from the inner assumption. Then, by the first outer assumption, $[M/x]N \doteq [M/x]N' \in A_2$. Applying the second displayed equation above, and the transitivity of equality at A_2, the result follows.

□

Symmetric and transitive relations are called *partial equivalence relations*, or *p.e.r.'s*. The remark in the proof about deriving reflexivity for related elements, called the "p.e.r. trick," will be of further use below.

Exercise 1. *Check the remaining cases of symmetry and transitivity.*

The analogue of the fundamental theorem in Harper (2024) is the reflexivity of exact equality. Define $\gamma \doteq \gamma' \in \Gamma$ variable-by-variable and define $\Gamma \gg M \doteq M' \in A$ to mean if $\gamma \doteq \gamma' \in \Gamma$, then $\hat{\gamma}(M) \doteq \hat{\gamma}'(M') \in A$.

Lemma 2 (Head Expansion)**.** *If* $M \doteq M' \in A$ *and* $N \longmapsto M$, *then* $N \doteq M' \in A$.

The analogous propery for the right-hand side of the equation follows from symmetry, or may be proved separately by an analogous argument.

Theorem 3 (Reflexivity)**.** *If* $\Gamma \vdash M : A$, *then* $\Gamma \gg M \in A$.

Proof. By induction on typing derivations, proceeding analogously to the proof given in Harper (2024). □

Observe that the full meaning of reflexivity of open terms involves disparate substitution instances of them. This is necessitated by the definition of computability at function types.

The fundamental theorem tells us that well-typed terms are exactly equal to themselves. At first this may sound trivial, but because exact equality is a *behavioral* condition on evaluation, it requires proof, and can even fail when a type system is not properly designed. By Lemma 1 exact equality for *closed* terms is symmetric and transitive. However, this does not immediately imply that the same is true for *open* terms!

Lemma 4 (Symmetry and Transitivity)**.** *1. If* $\Gamma \gg M \doteq M' \in A$*, then* $\Gamma \gg M' \doteq M \in A$*.*

2. If $\Gamma \gg M \doteq M' \in A$ *and* $\Gamma \gg M' \doteq M'' \in A$*, then* $\Gamma \gg M \doteq M'' \in A$*.*

Proof. 1. Assume that $\Gamma \gg M \doteq M' \in A$, and suppose that $\gamma' \doteq \gamma \in \Gamma$, with the intent to show that $\widehat{\gamma'}(M') \doteq \hat{\gamma}(M) \in A$. Simply instantiating the assumption yields $\widehat{\gamma'}(M) \doteq \hat{\gamma}(M') \in A$, which is neither the intended result, nor its symmetric form. Instead, by the symmetry of closed exact equality, $\gamma \doteq \gamma' \in \Gamma$ holds as well, so that instantiating the assumption yields the desired result.

2. Assume the two premises, and suppose that $\gamma \doteq \gamma'' \in \Gamma$, with the intent to show $\hat{\gamma}(M) \doteq \widehat{\gamma''}(M'') \in A$. Instantiating the two premises directly yields

   (a) $\hat{\gamma}(M) \doteq \widehat{\gamma''}(M') \in A$, and
   (b) $\hat{\gamma}(M') \doteq \widehat{\gamma''}(M'') \in A$.

   These do not combine to yield the desired result. Instead, using again that the supposition governing the substitutions implies that $\gamma \doteq \gamma \in \Gamma$, we obtain $\hat{\gamma}(M) \doteq \hat{\gamma}(M') \in A$, which combines with the second equation above by transitivity to yield the desired conclusion.

□

The rules in Figure 1 may be validated as expressing true exact equations.

Theorem 5 (Equational Validity)**.** *If* $\Gamma \vdash M \equiv N : A$*, then* $\hat{\gamma}(M) \doteq \widehat{\gamma'}(N) \in A$ *for all* $\gamma \doteq \gamma' \in \Gamma$*.*

Proof. The proof is by induction on the derivation of the equation, making use of the lemmas given above, including head expansion and reflexivity lemmas. The rules are formulated with typing premises that are essential to the argument. In particular the rule for β-equivalence for function types relies on the combination of the two typing premises to obtain an equation between two instances of the right-hand side of the equation, with the result then following by head expansion. □

Exercise 2. *Complete the proof of Theorem 5 in the indicated manner. Which typing premises, if any, are needed to complete the proof?*

Semantic equality may be extended to the empty type, sum types, and natural numbers type as follows.

$$M \doteq M' \in \mathsf{void} \text{ iff } \textit{(never)}$$

$$M \doteq M' \in A_1 + A_2 \text{ iff } M \longmapsto^* 1 \cdot M_1,\ M' \longmapsto^* 1 \cdot M_1' \text{ and } M_1 \doteq M_1' \in A_1 \text{ or}$$
$$M \longmapsto^* 2 \cdot M_2,\ M' \longmapsto^* 2 \cdot M_2' \text{ and } M_2 \doteq M_2' \in A_1$$

$$M \doteq M' \in \mathsf{nat} \text{ iff } \mathcal{N}(M, M'), \text{ where } \mathcal{N} \text{ is the strongest relation such that } \mathcal{N}(M, M') \text{ if}$$
$$M, M' \longmapsto^* \mathsf{zero}, \text{ or } M \longmapsto^* \mathsf{succ}(N),\ M' \longmapsto^* \mathsf{succ}(N') \text{ and } \mathcal{N}(N, N').$$

$$\text{REFL}\ \frac{\Gamma \vdash M : A}{\Gamma \vdash M \equiv M : A} \qquad \text{SYM}\ \frac{\Gamma \vdash M \equiv N : A}{\Gamma \vdash N \equiv M : A} \qquad \text{TRANS}\ \frac{\Gamma \vdash M \equiv N : A \qquad \Gamma \vdash N : A \qquad \Gamma \vdash N \equiv P : A}{\Gamma \vdash M \equiv P : A}$$

$$1\text{-}\eta\ \frac{\Gamma \vdash M : 1}{\Gamma \vdash M \equiv \langle\rangle : 1} \qquad \times\text{-I}\ \frac{\Gamma \vdash M_1 \equiv N_1 : A_1 \qquad \Gamma \vdash M_2 \equiv N_2 : A_2}{\Gamma \vdash \langle M_1, M_2\rangle \equiv \langle N_1, N_2\rangle : A_1 \times A_2} \qquad \times\text{-E-L}\ \frac{\Gamma \vdash M \equiv N : A_1 \times A_2}{\Gamma \vdash M \cdot 1 \equiv N \cdot 1 : A_1}$$

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

$$\times\text{-}\eta\ \frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash M \equiv \langle M \cdot 1, M \cdot 2\rangle : A_1 \times A_2} \qquad \rightarrow\text{-I}\ \frac{\Gamma, x : A_1 \vdash M_2 \equiv N_2 : A_2}{\Gamma \vdash \lambda(x.M_2) \equiv \lambda(x.N_2) : A_1 \rightarrow A_2}$$

$$\rightarrow\text{-E}\ \frac{\Gamma \vdash M \equiv N : A_1 \rightarrow A_2 \qquad \Gamma \vdash M_1 \equiv N_1 : A_1}{\Gamma \vdash \mathsf{ap}(M;M_1) \equiv \mathsf{ap}(N;N_1) : A_2} \qquad \rightarrow\text{-}\beta\ \frac{\Gamma, x : A_1 \vdash M_2 : A_2 \qquad \Gamma \vdash M_1 : A_1}{\Gamma \vdash \mathsf{ap}(\lambda(x.M_2);M_1) \equiv [M_1/x]M_2 : A_2}$$

$$\rightarrow\text{-}\eta\ \frac{\Gamma \vdash M : A_1 \rightarrow A_2}{\Gamma \vdash M \equiv \lambda(x.\,\mathsf{ap}(M;x)) : A_1 \rightarrow A_2}$$

Figure 1: Definitional Equivalence

Exact equality of natural numbers is inductively defined by the stated (and expected) conditions. To prove that some property holds of exactly equal natural numbers, it suffices to exhibit a relation R that satisfies the stated conditions. This is not exactly mathematical induction, but is obviously and intentionally closed related to it.

Exercise 3. *What equations should be added to the rules in Figure 1 to account for sums? State these, and prove that they are semantically valid. Pay special attention to η-like principles, which characterize all of the elements of a type.*

Exercise 4. *Extend the rules in Figure 1 to account for natural numbers, and verify that they are valid with respect to semantic equality. What η-like principles are feasible? What limitations do you encounter?*

Exercise 5. *Define exact equality for co-natural numbers, and prove reflexivity for the corresponding extension of the statics. What reasoning principle is validated by your definition of semantic equality for this type? What are appropriate equations for these constructs, and why are they valid?*

3 Zig-Zag Closure

Proving Lemma 1 at the outset makes use of, and makes available, the "p.e.r. trick" in the proof of Lemma 4. Another line of argumentation in the proof of Lemma 4 leads to an interesting variation on the foregoing development that will be essential in more general settings (see Harper (2020).) In this formulation reflexivity is taken as a *presupposition* of the exact equality judgment, which is to say that when speaking of $\Gamma \gg M \doteq M' \in A$, it is *pre-supposed* that M and M' are semantically sensible.

Let us then revisit the proof of Lemma 4 with this in mind. In both the symmetric and transitive cases the presuppositions are instantiated with the given substitutions, yielding the horizontal lines, oriented left-to-right, as depicted in Figure 3. The (solid) diagonal lines are provided by similarly instantiating the assumptions. The desired conclusions are then indicated by the dashed lines. In each case the diagonal is the *completion* of a *zig-zag* as depicted abstractly in Figure 2. Given Lemma 1 the desired completions may be obtained using symmetry to reverse the orientation of a line and transitivity to compose lines, and hence to complete these zig-zags, finishing the proofs.

This suggests another perspective. A binary relation R is *zig-zag complete* iff $R \circ R^{\mathrm{op}} \circ R \subseteq R$; that is, if $R(M, M')$ and $R(N, M')$ and $R(N, N')$, then $R(M, N')$. A useful visualization is given in Figure 2 in which the premises are indicated by solid lines and the conclusion by the dashed line, bearing in mind that these lines are oriented from left to right. Noting that the only use of symmetry and transivity in the foregoing development is in the proof of Lemma 4, it is sufficient to prove that the relations are, instead, zig-zag complete, and then to appeal to this property directly to complete the (reformulated) proof of symmetry and transitivity for open terms.

Exercise 6. *Prove that closed exact equality is zig-zag complete by induction on type.*

The reflexivity of exact equality is the statement that well-typed terms satisfy the behavioral conditions dictated by their types.

Theorem 6 (Reflexivity, Revisited)**.** *If $\Gamma \vdash M : A$, then $\Gamma \gg M \in A$.*

Exercise 7. *Prove the reflexivity theorem by induction on typing.*

The soundness theorem for the derivable equations is reformulated as follows. Then, the validity theorem is formulated as follows:

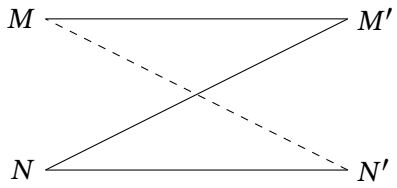

Figure 2: Zig-Zag Completeness

Theorem 7 (Equational Validity, Revisited)**.** *If* $\Gamma \vdash M \equiv N : A$*, then* $\Gamma \gg M \in A$ *and* $\Gamma \gg N \in A$ *imply* $\Gamma \gg M \doteq N \in A$*.*

The two membership conditions in Theorem 7 express the presuppositions required for the truth of an equation.

Figure 3 illustrates the proof of symmetry and transitivity in terms of zig-zag diagrams.

Exercise 8. *Prove the validity of the symmetry and transitivity rules given in Figure 1 in the sense of Theorem 7. What makes the typing premise on the transitivity rule necessary?*

Exercise 9. *Prove the validity of the* β *and* η *equality rules given in Figure 1 in the sense of Theorem 7. Given the presuppositions, which, if any, of the typing premises are necessary?*

Observe that the reflexity rule is trivially validated by the presupposition, which may be discharged by applying Theorem 6.

References

Robert Harper. Reynolds's parametricity theorem, directly. Unpublished lecture note, Spring 2020. URL `https://www.cs.cmu.edu/~rwh/courses/atpl/pdfs/reynolds.pdf`.

Robert Harper. How to (re)invent Tait's method. Unpublished lecture note, Spring 2024. URL `https://www.cs.cmu.edu/~rwh/courses/atpl/pdfs/tait.pdf`.

Neelakantan R. Krishnaswami and Derek Dreyer. Internalizing Relational Parametricity in the Extensional Calculus of Constructions. pages 20 pages, 591837 bytes, 2013. ISSN 1868-8969. doi: 10.4230/LIPICS.CSL.2013.432. URL `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CSL.2013.432`. Artwork Size: 20 pages, 591837 bytes ISBN: 9783939897606 Medium: application/pdf Publisher: Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Per Martin-Löf. Constructive mathematics and computer programming. In L. Jonathan Cohen, Jerzy Łoś, Helmut Pfeiffer, and Klaus-Peter Podewski, editors, *Logic, Methodology and Philosophy of Science VI, Proceedings of the Sixth International Congress of Logic, Methodology and Philosophy of Science, Hannover 1979*, volume 104 of *Studies in Logic and the Foundations of Mathematics*, pages 153–175. North-Holland, 1982. doi: 10.1016/S0049-237X(09)70189-2. URL `http://dx.doi.org/10.1016/S0049-237X(09)70189-2`.

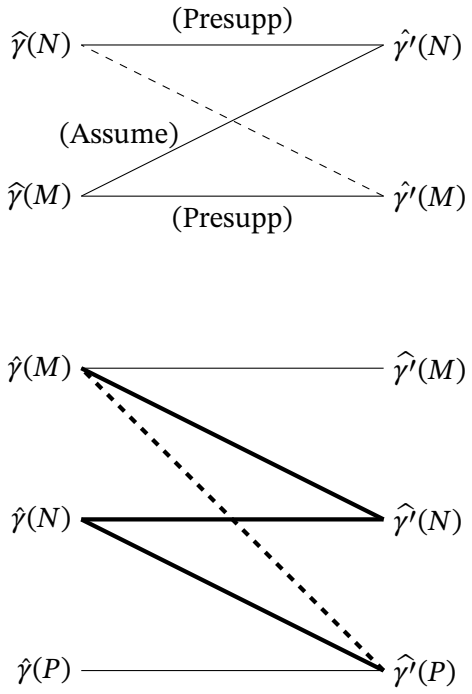

Figure 3: Symmetry and Transitivity via Zig-Zag Completeness