Weak Equality Reflection in MLTT with Propositional Truncation

Felix Bradley and Zhaohui Luo

Royal Holloway, University of London, Egham, U.K.
felix.bradley@rhul.ac.uk
zhaohui.luo@hotmail.co.uk

Introduction In type theory, there are two primary kinds of equality: judgemental equality, and propositional equality. The former is definitional and takes the form of equality judgements $\Gamma \vdash a = b : A$, where as the latter inhabits the theory's internal logic wherein you can have proof objects to associate an identity or equality between two objects, e.g. of the form $p : \mathrm{Id}_A(a, b)$. Equality reflection is a statement about the relationship between these two notions of equality: strong equality reflection allows for judgemental equality to be derived from propositional equality within a system, whereas weak equality reflection simply asks whether these two different notions of equality coincide. A theory with strong equality reflection, such as an extensional type theory, may have an inference rule such as

$$\frac{\Gamma \vdash p : \mathrm{Eq}_A(x, y)}{\Gamma \vdash x = y : A}$$

whereas a system with weak equality reflection is defined to be one such that at least one of the following rules[1]

$$\frac{\langle\rangle \vdash p : \mathrm{Id}_A(x, y)}{\langle\rangle \vdash x = y : A} \qquad \frac{\langle\rangle \vdash p : x =_A y}{\langle\rangle \vdash x = y : A}$$

is admissible within the system for the empty context, but not necessarily derivable. Weak equality reflection holds in systems such as MLTT [6, 8] and UTT [3], but fails to hold in systems such as traditional homotopy type theory[2] [2].

Some developed applications of type theory, such as program specification/analysis and modern type theory semantics for natural language [9, 1, 5], have been developed with the use of weak equality reflection in mind. For example, weak equality reflection in program specification is critically used to define what is expected from definitional and computational equality. Some applications allow for various system to serve as a sufficient foundation - however, MLTT presents issues for some of applications in natural language semantics due to the counting problem [11]. This work analyses a proposed solution in $\mathrm{MLTT_h}$ - the type theory MLTT extended with propositional truncation - and examines weak equality reflection within this system.

The Counting Problem For a sentence such as 'the number of black cats in the garden is one', one may provide type-theoretic semantics for the adjectival modification as a dependent pair type, as first proposed and studied by Mönnich [7] and Sundholm [10]. as

[1] Id is the identity type as found within MLTT [8], and $=_A$ is Leibniz equality as found within Luo's UTT [3].

[2] Consider the mere proposition $\Sigma(x : S^1).\, \mathrm{Id}(S^1, \mathrm{base}, x)$, where S^1 is the higher inductive type of the circle as typically defined.

$|\Sigma(x : \mathrm{Cat}).black(x)| = 1$. However, as theories such as MLTT use a propositions-as-types logic and lack proof irrelevance, it may be the case that there exist multiple distinct proofs that a cat is black, and so the above semantics would be incorrect. As such, MLTT alone is not adequate as a foundational system for these purposes - but MLTT extended with the proper features can be. It was proposed that $\mathrm{MLTT_h}$ - the extension of MLTT with propositional truncation [2] - could serve as an adequate system for formalising modern type theory semantics for natural language [4].

Propositional Truncation $\mathrm{MLTT_h}$ is defined[3] as MLTT extended with propositional truncation - a type-level operator $\|-\|$ such that

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash |a| : \|A\|} \qquad \frac{\Gamma \vdash A\,\mathrm{type}}{\Gamma \vdash p : \mathrm{isProp}(\|A\|)} \qquad \frac{\Gamma \vdash \mathrm{isProp}(B)\,\mathrm{true} \quad \Gamma \vdash f : A \to B}{\Gamma \vdash \kappa_A(f) : \|A\| \to B}$$

where the elimination operator κ_A satisfies the definitional equality $\kappa_A(f, |a|) = f(a)$. Importantly, this introduces a kind of higher inductive type to the theory by mandating that any two objects of $\|A\|$ must be propositionally equal.

This allows us to define the traditional logical operators for this system's internal logic as follows:

- $true = \mathbf{1}$
- $false = \mathbf{1}$
- $P \mathbin{\dot\wedge} Q = P \times Q$
- $P \mathbin{\dot\vee} Q = \|P + Q\|$
- $P \mathbin{\dot\supset} Q = P \to Q$
- $\dot\neg\, P = P \to \mathbf{0}$
- $\dot\forall(x : P).Q = \Pi(x : P).Q$
- $\dot\exists(x : P).Q = \|\Sigma(x : P).Q\|$

Weak Equality Reflection This new internal logic replaces MLTT's propositions-as-types logic with a system built on proof irrelevance, allowing for $\mathrm{MLTT_h}$ to serve as a foundational system for e.g. natural language semantics. However, it is easy to show that this system no longer has weak equality reflection. For two terms $a : A$ and $b : B$, we can consider the example of $\|A + B\|$; we can conclude that $|\mathrm{inl}(a)|$ and $|\mathrm{inr}(b)|$ are propositionally equal as objects of a mere proposition, yet judgementally distinct due to their constructors.

However, because of how $\mathrm{MLTT_h}$ is constructed, there is a subtheory of it that resembles MLTT. Weak equality reflection holds for MLTT, so this subtheory of $\mathrm{MLTT_h}$ should preserve weak equality reflection, even if it does not hold for $\mathrm{MLTT_h}$ as a whole. In this case, $\mathrm{MLTT_h}$ could still be suitable for the applications discussed earlier.

One method for proving this being considered is through showing that $\mathrm{MLTT_h}$ is a conservative extension of MLTT. In particular, one would want to show that, for every context Γ and type A obtained in MLTT, if there exists some term a in $\mathrm{MLTT_h}$ such that $\Gamma \vdash a : A$, then there exists some term a' in MLTT such that $\Gamma \vdash a' : A$. If one is able to prove this then, because weak equality reflection holds for MLTT, it follows that the MLTT-like subtheory of $\mathrm{MLTT_h}$ must also preserve weak equality reflection.

Conclusion Our work on extending MLTT with propositional truncation and analysing its properties shows that $\mathrm{MLTT_h}$ is able to function as a foundational language for applications

[3] Some other works by the second author also refer to this system as MLTT *with h-logic* [4].

such as in natural language semantics, even if the theory as a whole does not necessarily preserve weak equality reflection. It also shows that some subsets of homotopy type theory are not able to fully preserve weak equality reflection, which suggests that it may not be possible for a notion of higher inductive types and weak equality reflection to exist within the same system. Our currently in-progress work aims to refine these results, and to further explore expanding MLTT_h with features which are useful but may not have been fully developed yet for homotopy type theory in general, such as with coercive subtyping.

References

[1] Stergios Chatzikyriakidis and Zhaohui Luo. *Formal Semantics in Modern Type Theories*. Wiley/ISTE, 2020.

[2] HoTT. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, The Univalent Foundations Program, Institute for Advanced Study, 2013.

[3] Zhaohui Luo. *Computation and Reasoning: A Type Theory for Computer Science*. Oxford University Press, London, March 1994.

[4] Zhaohui Luo. Proof irrelevance in type-theoretical semantics. *Logic and Algorithms in Computational Linguistics 2018 (LACompLing2018), Studies in Computational Intelligence 860*, pages 1–15, 2019.

[5] Zhaohui Luo. *Modern Type Theories: Their Development and Applications*. Tsinghua University Press, 2024. (In Chinese).

[6] Per Martin-Löf. About models for intuitionistic type theories and the notion of definitional equality. *Studies in Logic and the Foundations of Math*, 82, 1975.

[7] U. Mönnich. *Untersuchungen zu einer konstruktiven Semantik fur ein Fragment des Englischen*. Habilitation. University of Tübingen, 1985.

[8] Bengt Nordström, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löf's Type Theory: An Introduction*. International Series of Monographs in Computer Science. Oxford University Press, London, June 1990.

[9] Aarne Ranta. *Type-Theoretical Grammar*. Oxford University Press, Oxford, 1994.

[10] Göran Sundholm. Proof theory and meaning. In *Handbook of philosophical logic*, pages 471–506. Springer, 1986.

[11] Göran Sundholm. Constructive generalized quantifiers. *Synthese*, 79(1):1–12, 1989.