Strong Normalization as Transfinite Induction on Reduction*

Robert Harper

September 3, 2025

1 Introduction

A formal type system is inductively defined by a collection of rules for deriving judgments of the form $\Gamma \vdash M : A$ and $\Gamma \vdash M \equiv M' : A$ expressing, respectively, that a term M is structurally well-typed at type A, relative to typing assumptions for variables given by the context Γ, and that well-typed terms M and M' of type A are *convertible*, or *definitionally equivalent*, at type A, relative to context Γ.

The question arises, is conversion decidable for well-typed terms? By Scott's Theorem conversion between untyped terms is undecidable, so the only hope for proving decidability is to take advantage of typing. Many such methods have been developed. *Reduction methods* work by checking whether or not two terms of the same type can be reduced to a common term by applying simplifications. *Normalization methods* are reduction methods that demand that the common term be itself irreducible, a *normal form*.

The use of reduction and normalization-based methods for deciding conversion may be justified by the principle of *transfinite induction on reduction (TIR)*. Writing $M \to N$ for reduction, a property $\mathcal{P}$ of terms is said to be $\to$*-inductive* iff

$$\forall M.(\forall N.M \to N \supset \mathcal{P}(N)) \supset \mathcal{P}(M).$$

The principle of transfinite $\to$-induction states that any $\to$-inductive property holds of every term:

$$\text{if } \mathcal{P} \text{ is } \to\text{-inductive, then } \forall M.\mathcal{P}(M).$$

The validity of transfinite $\to$-induction may be taken as the definition of the well-foundedness of the reduction relation. One use of this principal is to show that local confluence of a reduction relation suffices for its confluence, which leads to a decision procedure for definitional equivalence that reduces both sides to normal form.

In the literature it is common to take a more classical (as distinct from constructive) viewpoint. *Strong normalization (SN)* states that there are no infinite reduction sequences from a well-typed term. As the terminology suggests, this property is stronger than mere normalization, for which a particular strategy is either specified explicitly or is implicit in the constructive content of the proof that every term reduces to a normal form. Strong normalization may be derived from transfinite induction on reduction, for if all β-reducts of a term are SN, then so is that term. However, it is more usual to prove strong normalization directly—as Tait did—and to derive TIR from it by a "least counterexample" argument. The proof is thus indirect, and lacking in computational content.

This note develops a set of conditions on an otherwise abstract property $\mathcal{P}$ such that every well-typed term has that property using a Kripke-style formulation of Tait's method. The property of being strongly normalizing satisfies these conditions, providing a direct proof of SN for well-typed terms. The SN property suffices to validate transfinite induction on reduction, which is used in the proof of correctness of an algorithm for checking equivalence of typed terms.

2 Validity of Transfinite β-Reduction for Well-Typed Terms

The simply typed λ-calculus is as defined in Harper (2025b). The β-reduction relation, $\to_\beta$, for it is defined in Harper (2025a). Fix a family of properties ${\mathcal{P}_A}^\Gamma(M)$ governing well-typed terms $\Gamma \vdash M : A$.[1] Under what conditions does every well-typed term have property $\mathcal{P}$?

Conjecture 1. *If $\Gamma \vdash M : A$, then ${\mathcal{P}_A}^\Gamma(M)$.*

The proof makes use of Tait computability, as would be expected, yielding a set of conditions on $\mathcal{P}$ that suffice for the proof.

Define the family of predicates $\mathsf{h}{\mathcal{P}_A}^\Delta(M)$, called *hereditarily* $\mathcal{P}$, by induction on A as follows:

$$
\begin{aligned}
\mathsf{h}{\mathcal{P}_1}^\Delta(M) &\text{ iff } M \longmapsto^*_\beta \langle\rangle \text{ or } M \longmapsto^*_\beta U \text{ and } \mathsf{n}{\mathcal{P}_1}^\Delta(U)\\
\mathsf{h}{\mathcal{P}_{A_1\times A_2}}^\Delta(M) &\text{ iff } \mathsf{h}{\mathcal{P}_{A_1}}^\Delta(M\cdot 1) \text{ and } \mathsf{h}{\mathcal{P}_{A_2}}^\Delta(M\cdot 2)\\
\mathsf{h}{\mathcal{P}_{A_1\to A_2}}^\Delta(M) &\text{ iff } \forall \Delta' \leq \Delta \text{ if } \mathsf{h}{\mathcal{P}_{A_1}}^{\Delta'}(M_1) \text{ then } \mathsf{h}{\mathcal{P}_{A_2}}^{\Delta'}(\mathsf{ap}(M\,;M_2))
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{n}{\mathcal{P}_A}^{\Delta,x:A}(x) &\text{ iff (always)}\\
\mathsf{n}{\mathcal{P}_{A_1}}^\Delta(U\cdot 1) &\text{ iff } \mathsf{n}{\mathcal{P}_{A_1\times A_2}}^\Delta(U)\\
\mathsf{n}{\mathcal{P}_{A_2}}^\Delta(U\cdot 2) &\text{ iff } \mathsf{n}{\mathcal{P}_{A_1\times A_2}}^\Delta(U)\\
\mathsf{n}{\mathcal{P}_{A_2}}^\Delta(\mathsf{ap}(U\,;M_1)) &\text{ iff } \mathsf{n}{\mathcal{P}_{A_1\to A_2}}^\Delta(U) \text{ and } \mathsf{h}{\mathcal{P}_{A_1}}^\Delta(M_1)
\end{aligned}
$$

The definition of $\mathsf{h}{\mathcal{P}_A}^\Delta(M)$ makes use of the auxiliary property, $\mathsf{n}{\mathcal{P}_A}^\Delta(U)$, meaning that U is *neutrally* $\mathcal{P}$, which is defined by induction on the structure of U to require that the argument terms within U have property $\mathsf{h}\mathcal{P}$.

Lemma 1 (Pas-de-deux)**.** *1. If $\mathsf{n}{\mathcal{P}_A}^\Delta(U)$, then $\mathsf{h}{\mathcal{P}_A}^\Delta(U)$.*

2. If $\mathsf{h}{\mathcal{P}_A}^\Delta(M)$, then ${\mathcal{P}_A}^\Delta(M)$.

Proof. Simultaneously, by induction on A.

($A = 1$) Consider each case in turn:

1. By definition.
2. Suppose that $\mathsf{h}{\mathcal{P}_1}^\Delta(M)$. Then ${\mathcal{P}_1}^\Delta(M)$ by conditions (1) and (2) on $\mathcal{P}$.

($A = A_1 \times A_2$) Consider each case:

[1]The "staggered" sub- and super-scripts indicate the order of quantification.

1. Suppose that $\mathsf{n}\mathcal{P}_A{}^\Delta(U)$, with the intent to show $\mathsf{h}\mathcal{P}_A{}^\Delta(U)$. It suffices to show $\mathsf{h}\mathcal{P}_{A_1}{}^\Delta(U \cdot 1)$ and $\mathsf{h}\mathcal{P}_{A_2}{}^\Delta(U \cdot 2)$. The definition of $\mathsf{n}\mathcal{P}$ ensures that $\mathsf{n}\mathcal{P}_{A_1}{}^\Delta(U \cdot 1)$ and $\mathsf{n}\mathcal{P}_{A_2}{}^\Delta(U \cdot 1)$. But then the required follow by two applications of part (1) of the inductive hypthesis.
2. Suppose that $\mathsf{h}\mathcal{P}_A{}^\Delta(M)$, with the intent to show $\mathcal{P}_A{}^\Delta(M)$. The definition of $\mathsf{h}\mathcal{P}$ implies that $\mathsf{h}\mathcal{P}_{A_1}{}^\Delta(M \cdot 1)$ and $\mathsf{h}\mathcal{P}_{A_2}{}^\Delta(M \cdot 2)$, so by part (2) of the inductive hypothesis $\mathcal{P}_{A_1}{}^\Delta(M \cdot 1)$ and $\mathcal{P}_{A_2}{}^\Delta(M \cdot 2)$. But then the result follows by condition (3) on $\mathcal{P}$.

($A = A_1 \to A_2$) Consider each case:

1. Suppose that $\mathsf{n}\mathcal{P}_A{}^\Delta(U)$, with the intent to show $\mathsf{h}\mathcal{P}_A{}^\Delta(U)$. To this end suppose that $\Delta' \leq \Delta$ and $\mathsf{h}\mathcal{P}_{A_1}{}^{\Delta'}(M_1)$. But then $\mathsf{n}\mathcal{P}_{A_2}{}^{\Delta'}(\mathsf{ap}(U\,;M_1))$ by definition of $\mathsf{n}\mathcal{P}$, so by part (1) of the inductive hypothesis $\mathsf{h}\mathcal{P}_{A_2}{}^{\Delta'}(\mathsf{ap}(U\,;M_1))$, as required.
2. Suppose that $\mathsf{h}\mathcal{P}_A{}^\Delta(M)$, so as to show $\mathcal{P}_A{}^\Delta(M)$. Let $\Delta' = \Delta, x : A_1$, and note that by definition $\mathsf{n}\mathcal{P}_{A_1}{}^{\Delta'}(x)$. But then by inductive hypothesis part (1) it follows that $\mathsf{h}\mathcal{P}_{A_1}{}^{\Delta'}(x)$, and so by definition of $\mathsf{h}\mathcal{P}$ at function type, $\mathsf{h}\mathcal{P}_{A_2}{}^{\Delta'}(\mathsf{ap}(M\,;x))$. But then inductive hypothesis part (2) gives $\mathcal{P}_{A_2}{}^{\Delta'}(\mathsf{ap}(M\,;x))$, so that by condition (4) on $\mathcal{P}$, $\mathcal{P}_A{}^\Delta(M)$, as required.

□

The lemma is so-named because of the argument at function type, which swaps back and forth between the two parts of the inductive hypothesis.

Corollary 2. *Variables are computable:* $\mathsf{h}\mathcal{P}_\Gamma{}^\Gamma(id_\Gamma)$.

The proof of Lemma 1 relies on these conditions on the property $\mathcal{P}$:

1. $\mathcal{P}_1{}^\Delta(\langle\rangle)$.
2. If $\mathsf{n}\mathcal{P}_1{}^\Delta(U)$, then $\mathcal{P}_1{}^\Delta(U)$.
3. If $\mathcal{P}_{A_1}{}^\Delta(M \cdot 1)$ and $\mathcal{P}_{A_2}{}^\Delta(M \cdot 2)$, then $\mathcal{P}_{A_1 \times A_2}{}^\Delta(M)$.
4. If $\mathcal{P}_{A_2}{}^{\Delta, x : A_1}(\mathsf{ap}(M\,;x))$, then $\mathcal{P}_{A_1 \to A_2}{}^\Delta(M)$.

Any property of typed terms satisfying these conditions is called a *reduction property*.

Theorem 3. *Assuming that $\mathcal{P}$ is a reduction property, if $\Gamma \vdash M : A$, then $\mathsf{h}\mathcal{P}_\Gamma{}^\Delta(\gamma)$ implies $\mathsf{h}\mathcal{P}_A{}^\Delta(M)$.*

Proof. By induction on typing derivations. □

Exercise 1. *Prove theorem 3. Your proof will require two additional conditions on the property $\mathcal{P}$.*

Corollary 4. *If $\Gamma \vdash M : A$, then $\mathcal{P}_A{}^\Gamma(M)$.*

Proof. Choose γ to be the identity substitution on Γ. □

Exercise 2. *Prove that strong normalization is a reduction property. Conclude that no well-typed term has an infinite reduction sequence starting from it.*

Exercise 3. *Show that strong normalization implies transfinite induction on reduction for any property $\mathcal{P}$ of well-typed terms.* Hint: *Use a "least counterexample" argument with respect to reduction.*

3 Decidability of β-Equivalence

The best-known application of transfinite induction on reduction is to the proof of correctness of a reduction-based decision method for β-equivalence of terms.[2] Define the relation $M \downarrow_\beta N$ to mean that M and N have a common reduct, a term P such that $M \to_\beta^* P$ and $N \to_\beta^* P$. Clearly, if $\Gamma \vdash M, N : A$ and $M \downarrow_\beta N$, then $\Gamma \vdash M \equiv N : A$ via a chain of equivalences from M to N via their common reduct. Conversely, does $\Gamma \vdash M \equiv N : A$ imply that $M \downarrow_\beta N$? The evident strategy is to proceed by induction on the derivation of the equivalence. All cases go swimmingly, but for transitivity. For suppose that $\Gamma \vdash M \equiv N \equiv P : A$ with the intent to show that $M \downarrow_\beta N$. By induction $M \downarrow_\beta N$ and $N \downarrow_\beta P$, but can we conclude that $M \downarrow_\beta P$? Let Q be the common reduct of M and N and let Q' be the common reduct of N and P. To complete the proof it suffices to show that $Q \downarrow_\beta Q'$ (draw the picture!).

This suggests the following conjecture:

Lemma 5 (Confluence). *If $N \to_\beta^* Q$ and $N \to_\beta^* Q'$, then $Q \downarrow_\beta Q'$.*

In words any two multi-step paths from N "flow together" to a common reduct.

How might this be proved? A direct combinatorial proof seems difficult because of the multi-step split from N—there's no telling what Q and Q' might look like, much less how they may be reconciled. However, a multi-step reduction has a finite length, so it makes sense to consider $M \to_\beta{}^m Q$ and $M \to_\beta{}^{m'} Q'$, and then proceed by induction on m and m'. If either or both are zero, the answer is immediate, otherwise $m = n+1$ and $m' = n'+1$, and each path from N starts with a single β-reduction step. Aha! This seems more tractible, because it is not out of the question that all one-step splits are reconcilable; this property is called *local confluence*.

Lemma 6 (Local Confluence). *If $N \to_\beta Q$ and $N \to_\beta Q'$, then $Q \downarrow_\beta Q'$.*

It is not difficult to prove local confluence directly by considering, for each term N, all possible one-way splits, and showing them to be reconcilable.

It is tempting to conclude, by the heuristic argument just given, that $\to_\beta$ is confluent, and therefore equivalence for well-typed terms is equivalent to their having a common reduct. To complete the proof it suffices to prove (weak) normalization, so that the common reduct can be taken to be a normal form.

Exercise 4. *Try to prove that local confluence implies confluence by a simultaneous induction on the lengths of the paths in the multi-way split, as suggested above. Watch yourself fail. Explain, informally, why the proof does not go through.*

The difficulty, as you will have noticed, is that the proof attempt quickly breaks down with no applicable inductive hypothesis, and there is no way to bound the length of the offending multi-step reductions.[3]

Exercise 5. *Prove by transfinite induction on reduction that local confluence implies confluence. The heuristic argument survives to some extent, using the much stronger inductive hypothesis afforded by transfinite induction on reduction.*

Thus, in the presence of transfinite induction on reduction, local confluence suffices to justify the suggested decision method for equivalence by reduction to a common, irreducible, reduct.

Exercise 6 (Extra Credit). *Prove local confluence of β-reduction for the typed λ-calculus.*

[2] See Huet (1980) for an abstract account in any rewriting system.

[3] The cryptic nature of this comment is necessary to avoid giving away the solution to the exercise!

References

Robert Harper. Kripke-style logical relations for normalization. Unpublished lecture note, January 2025a. URL `https://www.cs.cmu.edu/~rwh/courses/atpl/pdfs/kripke.pdf`.

Robert Harper. How to (re)invent Tait's method. Unpublished lecture note, January 2025b. URL `https://www.cs.cmu.edu/~rwh/courses/atpl/pdfs/tait.pdf`.

Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems: Abstract properties and applications to term rewriting systems. *Journal of the ACM (JACM)*, 27(4): 797–821, 1980.