# Towards a Proof Theory of Rewriting: The Simply-Typed 2λ-Calculus

Barnaby P. Hilken*

November 26, 1999

**Abstract**

This paper describes the simply-typed 2λ-calculus, a language with three levels: types, terms and rewrites. The types and terms are those of the simply-typed λ-calculus, and the rewrites are expressions denoting sequences of $\beta$-reductions and $\eta$-expansions. An equational theory is imposed on the rewrites, based on 2-categorical justifications, and the word problem for this theory is solved by finding a canonical expression in each equivalence class.

The canonical form of rewrites allows us to prove several properties of the calculus, including a strong form of confluence and a classification of the long-$\beta\eta$-normal forms in terms of their rewrites. Finally we use these properties as the basic definitions of a theory of categorical rewriting, and find that the expected relationships between confluence, strong normalisation and normal forms hold.

## 1 Introduction

In the theoretical computer science community recently there has been much interest in *proof theory*: the study of logics not in terms of their consequence relations, but in terms of their proofs. The point of interest is not just *whether* propositions are provable, but *how* they are proved, and what mathematical structure can be given to proofs. This raises the question of which proofs should be considered equivalent, and which are distinct. Traditional proof theory answers this with the notions of cut elimination (for sequent calculus)

*This work was carried out as part of a SERC-funded project while the author was at the University of Manchester. The paper was completed while the author was a SERC postdoctoral fellow at the University of Cambridge.

and proof normalisation (for natural deduction), which identify proofs by certain syntactic rules. Typically, the introduction followed by immediate elimination of a connective is equated with a trivial proof. This kind of syntactic rule is really justified by the fact that it works: we are left with the suspicion that there might be another way to do it.

Categorical logic provides an alternative, more mathematical, approach to the same problem, at least for intuitionistic logic. Here the propositions and proofs of a logic are taken to be the objects and arrows of a category respectively, and two proofs are equal if, and only if, the corresponding arrows are forced to be equal by the axioms of category theory. In other words, the logic is identified with a free category of a certain form, depending on the connectives of the logic. The connectives are given universal properties: conjunction as product, disjunction as coproduct and implication as exponential, for example. Since universal properties characterise objects up to isomorphism, this gives a more convincing reason for identifying proofs. Categorical proof theory arises from the observation that the identifications justified by the category theory are the same as the traditional syntactic ones.

The aim of this work is to develop a proof theory for rewriting. Our analogy is this: the elements (terms, strings, etc.) of a rewrite system correspond to the propositions of a logic, and the rewrite relation $t \to_* s$ ($t$ rewrites in zero or more steps to $s$) corresponds to the consequence relation. The analogue of a proof we call a *rewrite*, and we write $\alpha\colon t \Rightarrow s$ when $\alpha$ is a rewrite whose effect is to transform $t$ into $s$. Just as proofs say how propositions are proved, so the rewrite $\alpha$ says how $t$ is rewritten to get $s$. We can think of $\alpha$ as an algorithm—perhaps as simple as a sequence of instances of rewrite rules—which expresses the necessary computational information. The questions we wish to study are: what form do such algorithms take, what mathematical structure do they have, and when are two of them equal?

The reflexivity and transitivity of the relation $\to_*$ suggest that we push our analogy further, and try to develop categorical rewriting. We take the elements and rewrites of a rewrite system to be the objects and arrows of a category respectively. Composition of arrows is sequential composition of rewrites, corresponding to the transitivity of $\to_*$, and identity arrows are "zero-step" rewrites, corresponding to reflexivity. We can then look for categorical justification of identifications between rewrites. In particular, we would hope that Seely's description of $\beta$-reduction and $\eta$-expansion as unit and counit of an adjunction [16] would fit this framework.

In this paper we study one particular rewrite system, the simply-typed $\lambda$-calculus, in some detail. We define the types and terms in the usual way, and give a language for rewrites generated from $\beta$-reduction and $\eta$-expansion by sequential and parallel composition. We then introduce equations between rewrites which are motivated by categorical considerations similar to those of Seely. These equations lead to a simple canonical form for rewrites, which solves the word problem, and allows us to prove several results about our system.

Generalising from this example, we then define categorical rewriting by which we mean a theory of rewriting which concerns not just the relation $\to_*$ but the rewrites themselves. Our intention is to axiomatise those categories which behave like the theory of rewriting considered earlier, independently of the particular syntax of the $\lambda$-calculus. Of course we do not want merely to recapture the usual notions of normal forms, confluence and normalisation, but rather to say when these notions interact well with the proof theory of the rewrites. We show that our main example has extremely good proof-theoretic properties, and characterise Huet's long-$\beta\eta$-normal forms [11] purely categorically.

## 1.1 Related Work

This work brings together several strands of recent research in theoretical computer science. The author's own introduction was via "2-categorical rewriting" [15, 16] which studies the relationship between rewriting and term structure. By concentrating on one particular system (the $\lambda$-calculus) this paper shows that this technique can lead to greater understanding of important languages.

The main content of the paper is the notion of equivalence between rewrites. Previous work on "strongly equivalent reductions" [14], summarised in [2, chapter 12], introduced an equivalence between $\beta$-reduction sequences with rather different motivation. In fact our equivalence agrees with the definition of strong equivalence on $\beta$-reductions, and extends it to $\eta$-expansions as well.

Recently there has been much interest in typed $\lambda$-calculi with $\eta$-expansion [1, 5, 6, 7, 12, 13], which seems to have much better confluence properties than $\eta$-contraction. [7] provides a valuable historical survey including older references. Of course, the problem with $\eta$-expansion is how to avoid the infinite sequences which arise. Our method is most closely related to that of [13], which advocates that certain "loops" should be "cut". Our work can be interpreted as giving an algebraic method for determining precisely which loops should be cut, and which represent genuine nontermination.

## 1.2 The Paper

In section 2 we define the simply-typed 2$\lambda$-calculus: its types, terms and rewrites. There are several forms of substitution, and we prove the basic syntactic properties. In section 3 we introduce the equations on rewrites of the theory 2-$\lambda$. We give an axiomatisation and a categorical semantics, and use the syntax to construct the free model.

In section 4 we solve the word problem for the theory 2-$\lambda$ by choosing a unique canonical rewrite from each equivalence class. This is the fundamental result of this paper; unfortunately the proof is long and intricate with a great many cases. The high points are a form of "cut-elimination" for rewrites, and the associativity of this "cut" operation.

In section 5 we investigate the relationship between the equational theory 2-$\lambda$ and the properties of the $\lambda$-calculus as a rewrite system. We show that the theory 2-$\lambda$ agrees with the notion of "strongly equivalent reductions" on $\beta$-rewrites, and extend the "strong Church-Rosser" theorem to our system. Next we prove a new property we call "mellifluence", and we characterise the long-$\beta\eta$-normal forms by a property of their rewrites.

Finally in section 6 we take the properties we have proved of the 2$\lambda$-calculus as the basic definitions of a categorical theory of rewriting. We show that in any mellifluent category, we can define notions closely related to (strong) confluence, (long) normal forms, weak and strong normalisation, such that weak normalisation implies existence of normal forms, and confluence implies their uniqueness up to isomorphism.

# 2 The Simply-Typed 2$\lambda$-Calculus

The simply-typed 2$\lambda$-calculus is a language of three syntactic classes, called **types**, **terms** and **rewrites**. Each term has a **context** which gives the types of free variables which might appear in the term, and a type. Each rewrite has a source term and a target term, which share a common context and type. These well-formedness conditions are expressed by two judgements:

- $\Gamma \vdash t\colon X$ means that $t$ is a well-formed term of type $X$ in context $\Gamma$.
- $\Gamma \vdash \gamma\colon t \Rightarrow u\colon X$ means that $\gamma$ is a well-formed rewrite with source $t$ and target $u$, where $t$ and $u$ are well-formed terms of type $X$ in context $\Gamma$.

The intended interpretation of the language is that the types and terms are those of the simply-typed $\lambda$-calculus, and the rewrites are algorithms which describe a sequence of $\beta$-reductions and $\eta$-expansions which can be applied to a term.

## 2.1 Syntax

Let $\mathbb{B}$ be a set of "basic types", with typical element $B$. The language is defined inductively as follows, where (in order to simplify several points) DeBruijn notation (see [4] or [2, Appendix C]) is used for variables.

### Types

$$X ::= B \mid X \to X$$

Since this is a simply-typed calculus, a type is built up from basic types using the $\to$ (function space) constructor.

A context $\Gamma$ is just a list of types $X_1, \ldots, X_n$.

### Terms

$$X_1, \ldots, X_n \vdash j \colon X_j \qquad (1 \leq j \leq n) \qquad \text{(variable)}$$

$$\frac{X, \Gamma \vdash t \colon Y}{\Gamma \vdash \lambda t \colon X \to Y} \qquad \text{(lambda)}$$

$$\frac{\Gamma \vdash t \colon X \to Y \quad \Gamma \vdash u \colon X}{\Gamma \vdash t\, u \colon Y} \qquad \text{(apply)}$$

A term $t$ is a term of the simply-typed $\lambda$-calculus, in DeBruijn notation.

**Rewrites**

$$X_1, \ldots, X_n \vdash j \colon j \Rightarrow j \colon X_j \qquad (1 \le j \le n) \qquad \text{(id)}$$

$$\frac{X, \Gamma \vdash \gamma \colon t \Rightarrow t' \colon Y}{\Gamma \vdash \lambda\gamma \colon \lambda t \Rightarrow \lambda t' \colon X \to Y} \qquad \text{(lambda)}$$

$$\frac{\Gamma \vdash \gamma \colon t \Rightarrow t' \colon X \to Y \quad \Gamma \vdash \delta \colon u \Rightarrow u' \colon X}{\Gamma \vdash \gamma\,\delta \colon t\,u \Rightarrow t'\,u' \colon Y} \qquad \text{(apply)}$$

$$\frac{\Gamma \vdash \gamma \colon t \Rightarrow t' \colon X \quad \Gamma \vdash \delta \colon t' \Rightarrow t'' \colon X}{\Gamma \vdash \gamma;\delta \colon t \Rightarrow t'' \colon X} \qquad \text{(compose)}$$

$$\frac{\Gamma \vdash t \colon X \to Y}{\Gamma \vdash \eta_t \colon t \Rightarrow \lambda(t^1\,1) \colon X \to Y} \qquad \text{(eta)}$$

$$\frac{X, \Gamma \vdash t \colon Y \quad \Gamma \vdash u \colon X}{\Gamma \vdash \beta_{t,u} \colon (\lambda t)\,u \Rightarrow t[u] \colon Y} \qquad \text{(beta)}$$

Rewrites are built up from $\beta$-reduction and $\eta$-expansion by sequential and parallel composition. By a simple induction, for any term $\Gamma \vdash t \colon X$ there is a rewrite $\Gamma \vdash t \colon t \Rightarrow t \colon X$, which we call an **identity** rewrite. The notations $t^1$ and $t[u]$ are defined below.

## 2.2 Substitution

For definiteness, we give our notation for substitution in some detail. The reader who is unfamiliar with DeBruijn notation should read this carefully, noting how variable capture and other problems are dealt with.

**Terms**

Firstly, $t^n$ is $t$ with all free variables greater than or equal to $n$ incremented by one:

$$j^n = \begin{cases} j+1 & \text{if } j \ge n, \\ j & \text{otherwise} \end{cases}$$
$$(t\,u)^n = t^n\,u^n$$
$$(\lambda t)^n = \lambda t^{n+1}$$

The effect of this operation on terms in context shows that it is a form of weakening: If $X_1, \ldots, X_m \vdash t \colon Y$, $1 \le n \le m+1$ and $X$ is a type, then $X_1, \ldots, X_{n-1}, X, X_n, \ldots, X_m \vdash t^n \colon Y$.

Next, $t[v_1, v_2, \ldots]$ is $t$ with $v_j$ substituted for $j$:

$$\begin{aligned} j[v_1, v_2, \ldots] &= v_j \\ (t\,u)[v_1, v_2, \ldots] &= t[v_1, v_2, \ldots]\,u[v_1, v_2, \ldots] \\ (\lambda t)[v_1, v_2, \ldots] &= \lambda t[1, v_1^1, v_2^1, \ldots] \end{aligned}$$

for brevity we write $t[u]$ for $t[u, 1, 2, \ldots]$. The effect of this on terms in context shows that it is a form of cut: If $X_1, \ldots, X_n \vdash t\colon Y$ and $\Gamma \vdash u_j\colon X_j$ for $j = 1, \ldots, n$, then $\Gamma \vdash t[u_1, \ldots, u_n]\colon Y$.

**Lemma 1.** Some basic properties of substitution:

1. The identity substitution: $t[1, 2, \ldots] = t$.
2. Associativity of term substitution:

$$t[u_1, u_2, \ldots][v_1, v_2, \ldots] = t[u_1[v_1, v_2, \ldots], u_2[v_1, v_2, \ldots], \ldots]$$

3. $t^2[1] = t$.
4. $(t[u_1, u_2, \ldots])^1 = t^1[1, u_1^1, u_2^1, \ldots]$.

*Proof.* These results (and the weakening and cut properties above) are all straightforward structural inductions. □

**Rewrites**

The operation of incrementing variables extends to rewrites in a straightforward way:

$$\begin{aligned} j^n &= \begin{cases} j+1 & \text{if } j \geq n, \\ j & \text{otherwise} \end{cases} \\ (\gamma\,\delta)^n &= \gamma^n\,\delta^n \\ (\lambda\gamma)^n &= \lambda\gamma^{n+1} \\ (\gamma;\delta)^n &= \gamma^n;\delta^n \\ \eta_t^n &= \eta_{t^n} \\ \beta_{t,u}^n &= \beta_{t^{n+1},u^n} \end{aligned}$$

Its effect on rewrites in context is as expected: If $X_1, \ldots, X_m \vdash \gamma\colon t \Rightarrow u\colon Y$, $1 \leq n \leq m{+}1$ and $X$ is a type, then $X_1, \ldots, X_{n-1}, X, X_n, \ldots, X_m \vdash \gamma^n\colon t^n \Rightarrow u^n\colon Y$.

Substitution extends to rewrites as two distinct operations. Substitution of rewrites into terms is defined by:

$$\begin{aligned} j[\gamma_1, \gamma_2, \dots] &= \gamma_j \\ (t\, u)[\gamma_1, \gamma_2, \dots] &= t[\gamma_1, \gamma_2, \dots]\, u[\gamma_1, \gamma_2, \dots] \\ (\lambda t)[\gamma_1, \gamma_2, \dots] &= \lambda t[1, \gamma_1^1, \gamma_2^1, \dots] \end{aligned}$$

giving a cut rule: If $X_1, \dots, X_n \vdash t \colon Y$ and $\Gamma \vdash \gamma_j \colon u_j \Rightarrow u'_j \colon X_j$ for $j = 1, \dots, n$, then $\Gamma \vdash t[\gamma_1, \dots, \gamma_n] \colon t[u_1, \dots, u_n] \Rightarrow t[u'_1, \dots, u'_n] \colon Y$.

Substitution of terms into rewrites is defined by

$$\begin{aligned} j[v_1, v_2, \dots] &= v_j \\ (\gamma\, \delta)[v_1, v_2, \dots] &= \gamma[v_1, v_2, \dots]\, \delta[v_1, v_2, \dots] \\ (\lambda \gamma)[v_1, v_2, \dots] &= \lambda \gamma[1, v_1^1, v_2^1, \dots] \\ (\gamma; \delta)[v_1, v_2, \dots] &= \gamma[v_1, v_2, \dots]; \delta[v_1, v_2, \dots] \\ \eta_t[v_1, v_2, \dots] &= \eta_{t[v_1, v_2, \dots]} \\ \beta_{t,u}[v_1, v_2, \dots] &= \beta_{t[1, v_1^1, v_2^1, \dots], u[v_1, v_2, \dots]} \end{aligned}$$

giving a cut rule: If $X_1, \dots, X_n \vdash \gamma \colon t \Rightarrow t' \colon Y$ and $\Gamma \vdash u_j \colon X_j$ for $j = 1, \dots, n$, then $\Gamma \vdash \gamma[u_1, \dots, u_n] \colon t[u_1, \dots, u_n] \Rightarrow t'[u_1, \dots, u_n] \colon Y$.

Note that there are three interpretations of $t[u_1, u_2, \dots]$ as a rewrite: the identity on $t[u_1, u_2, \dots]$, the substitution of $[u_1, u_2, \dots]$ into the identity on $t$ and substitution of identities on $[u_1, u_2, \dots]$ into $t$. A simple induction shows that these three are equal, so there is no ambiguity.

**Lemma 2.** Basic properties of substitution of rewrites:

1. Identity substitution: $\gamma[1, 2, \dots] = \gamma$.
2. Associativity of term-term-rewrite substitution:
$$t[u_1, u_2, \dots][\gamma_1, \gamma_2, \dots] = t[u_1[\gamma_1, \gamma_2, \dots], u_2[\gamma_1, \gamma_2, \dots], \dots]$$
3. Associativity of term-rewrite-term substitution:
$$t[\gamma_1, \gamma_2, \dots][v_1, v_2, \dots] = t[\gamma_1[v_1, v_2, \dots], \gamma_2[v_1, v_2, \dots], \dots]$$
4. Associativity of rewrite-term-term substitution:
$$\gamma[u_1, u_2, \dots][v_1, v_2, \dots] = \gamma[u_1[v_1, v_2, \dots], u_2[v_1, v_2, \dots], \dots]$$
5. $(t[\gamma_1, \gamma_2, \dots])^1 = t^1[1, \gamma_1^1, \gamma_2^1, \dots]$.
6. $\gamma^2[1] = \gamma$.
7. $(\gamma[u_1, u_2, \dots])^1 = \gamma^1[1, u_1^1, u_2^1, \dots]$.

*Proof.* Again, these are straightforward structural inductions. □

# 3 The Theory 2-$\lambda$

The theory 2-$\lambda$ is an equational theory on the *rewrites* of the 2$\lambda$-calculus. We write it as a judgement:

- $\Gamma \vdash \gamma = \delta \colon t \Rightarrow u \colon X$ means that $\gamma$ and $\delta$ are equivalent in the theory 2-$\lambda$, where $\gamma$ and $\delta$ are well-formed rewrites with source $t$ and target $u$, of type $X$ in context $\Gamma$.

The intention is to axiomatise not when two rewrites have the same effect—after all, we are only considering equations between rewrites with common source and target—but when two rewrites might be implemented identically; for example, a parallel rewrite might be implemented on a sequential machine in either order. This is an attempt to say when two rewrites represent the same algorithm.

## 3.1 The Axiomatisation of 2-$\lambda$.

The first axioms need no explanation, they merely formalise what might be called a 2$\lambda$-theory: an equivalence which respects the syntactic structure.

$$\frac{\Gamma \vdash \gamma \colon t \Rightarrow t' \colon X}{\Gamma \vdash \gamma = \gamma \colon t \Rightarrow t' \colon X} \qquad \text{(reflexivity)}$$

$$\frac{\Gamma \vdash \gamma = \delta \colon t \Rightarrow t' \colon X}{\Gamma \vdash \delta = \gamma \colon t \Rightarrow t' \colon X} \qquad \text{(symmetry)}$$

$$\frac{\Gamma \vdash \gamma = \delta \colon t \Rightarrow t' \colon X \quad \Gamma \vdash \delta = \epsilon \colon t \Rightarrow t' \colon X}{\Gamma \vdash \gamma = \epsilon \colon t \Rightarrow t' \colon X} \qquad \text{(transitivity)}$$

$$\frac{X, \Gamma \vdash \gamma = \gamma' \colon t \Rightarrow t' \colon Y}{\Gamma \vdash \lambda\gamma = \lambda\gamma' \colon \lambda t \Rightarrow \lambda t' \colon X \to Y} \qquad \text{($\lambda$-subst)}$$

$$\frac{\Gamma \vdash \gamma = \gamma' \colon t \Rightarrow t' \colon X \to Y \quad \Gamma \vdash \delta = \delta' \colon u \Rightarrow u' \colon X}{\Gamma \vdash \gamma\,\delta = \gamma'\,\delta' \colon t\,u \Rightarrow t'\,u' \colon Y} \qquad \text{(app-subst)}$$

$$\frac{\Gamma \vdash \gamma = \gamma' \colon t \Rightarrow t' \colon X \quad \Gamma \vdash \delta = \delta' \colon t' \Rightarrow t'' \colon X}{\Gamma \vdash \gamma; \delta = \gamma'; \delta' \colon t \Rightarrow t'' \colon X} \qquad \text{(;-subst)}$$

The particular axioms which define the theory 2-$\lambda$ are as follows. Rewrites $j$ act as left and right identities of composition:

$$\frac{\Gamma \vdash \gamma \colon j \Rightarrow t \colon X}{\Gamma \vdash j; \gamma = \gamma \colon j \Rightarrow t \colon X} \qquad \text{(id-l)}$$

$$\frac{\Gamma \vdash \gamma \colon t \Rightarrow j \colon X}{\Gamma \vdash \gamma; j = \gamma \colon t \Rightarrow j \colon X} \qquad \text{(id-r)}$$

Abstraction and application distribute over composition:

$$\frac{X,\Gamma \vdash \gamma\colon t \Rightarrow t'\colon Y \quad X,\Gamma \vdash \delta\colon t' \Rightarrow t''\colon Y}{\Gamma \vdash \lambda\gamma;\lambda\delta = \lambda(\gamma;\delta)\colon \lambda t \Rightarrow \lambda t''\colon X \to Y} \qquad (\lambda\text{-dist})$$

$$\frac{\begin{array}{c}\Gamma \vdash \gamma\colon t \Rightarrow t'\colon X \to Y \quad \Gamma \vdash \gamma'\colon t' \Rightarrow t''\colon X \to Y \\ \Gamma \vdash \delta\colon u \Rightarrow u'\colon X \quad \Gamma \vdash \delta'\colon u' \Rightarrow u''\colon X\end{array}}{\Gamma \vdash (\gamma\,\delta);(\gamma'\,\delta') = (\gamma;\gamma')\,(\delta;\delta')\colon t\,u \Rightarrow t''\,u''\colon Y} \qquad (\text{app-dist})$$

Composition is associative:

$$\frac{\Gamma \vdash \gamma\colon t \Rightarrow t'\colon X \quad \Gamma \vdash \delta\colon t' \Rightarrow t''\colon X \quad \Gamma \vdash \epsilon\colon t'' \Rightarrow t'''\colon X}{\Gamma \vdash \gamma;(\delta;\epsilon) = (\gamma;\delta);\epsilon\colon t \Rightarrow t'''\colon X} \qquad (\text{assoc})$$

$\eta$ and $\beta$ commute with rewrites of their subscripts:

$$\frac{\Gamma \vdash \gamma\colon t \Rightarrow t'\colon X \to Y}{\Gamma \vdash \eta_t;\lambda(\gamma^1\,1) = \gamma;\eta_{t'}\colon t \Rightarrow \lambda(t'^1\,1)\colon X \to Y} \qquad (\eta\text{-nat})$$

$$\frac{X,\Gamma \vdash \gamma\colon t \Rightarrow t'\colon Y \quad \Gamma \vdash \delta\colon u \Rightarrow u'\colon X}{\Gamma \vdash (\lambda\gamma)\,\delta;\beta_{t',u'} = \beta_{t,u};\gamma[u];t'[\delta]\colon (\lambda t)\,u \Rightarrow t'[u']\colon Y} \qquad (\beta\text{-nat})$$

$\eta$-expansion followed by $\beta$-reduction cancels out:

$$\frac{X,\Gamma \vdash t\colon Y}{\Gamma \vdash \eta_{\lambda t};\lambda\beta_{t^2,1} = \lambda t\colon \lambda t \Rightarrow \lambda t\colon X \to Y} \qquad (\text{triangle1})$$

$$\frac{\Gamma \vdash t\colon X \to Y \quad \Gamma \vdash u\colon X}{\Gamma \vdash \eta_t\,u;\beta_{t^1\,1,u} = t\,u\colon t\,u \Rightarrow t\,u\colon Y} \qquad (\text{triangle2})$$

**Lemma 3.** Basic properties of the theory, relating the equations to substitution:

1. Substitution into equations: If $X_1,\ldots,X_n \vdash \gamma = \delta\colon t \Rightarrow t'\colon Y$ and $\Gamma \vdash u_j\colon X_j$ for $j = 1,\ldots,n$, then

$$\Gamma \vdash \gamma[u_1,\ldots,u_n] = \delta[u_1,\ldots,u_n]\colon t[u_1,\ldots,u_n] \Rightarrow t'[u_1,\ldots,u_n]\colon Y$$

2. Substitution of equals for equals: If $X_1,\ldots,X_n \vdash t\colon Y$ and $\Gamma \vdash \gamma_j = \delta_j\colon u_j \Rightarrow u'_j\colon X_j$ for $j = 1,\ldots,n$, then

$$\Gamma \vdash t[\gamma_1,\ldots,\gamma_n] = t[\delta_1,\ldots,\delta_n]\colon t[u_1,\ldots,u_n] \Rightarrow t[u'_1,\ldots,u'_n]\colon Y$$

3. Composition with identities: If $\Gamma \vdash \gamma\colon t \Rightarrow u\colon X$ then $\Gamma \vdash t;\gamma = \gamma\colon t \Rightarrow u\colon X$ and $\Gamma \vdash \gamma;u = \gamma\colon t \Rightarrow u\colon X$.

4. Distributivity of substitution: If $X_1, \dots, X_n \vdash t \colon Y$, $\Gamma \vdash \gamma_j \colon u_j \Rightarrow u_j' \colon X_j$ and $\Gamma \vdash \delta_j \colon u_j' \Rightarrow u_j'' \colon X_j$ for $j = 1, \dots, n$, then

$$\Gamma \vdash t[\gamma_1; \delta_1, \dots, \gamma_n; \delta_n] = t[\gamma_1, \dots, \gamma_n]; t[\delta_1, \dots, \delta_n] \\ \colon t[u_1, \dots, u_n] \Rightarrow t[u_1'', \dots, u_n''] \colon Y$$

5. The interchange law: If $X_1, \dots, X_n \vdash \gamma \colon t \Rightarrow t' \colon Y$ and $\Gamma \vdash \delta_j \colon u_j \Rightarrow u_j' \colon X_j$ for $j = 1, \dots, n$, then

$$\Gamma \vdash \gamma[u_1, \dots, u_n]; t'[\delta_1, \dots, \delta_n] = t[\delta_1, \dots, \delta_n]; \gamma[u_1', \dots, u_n'] \\ \colon t[u_1, \dots, u_n] \Rightarrow t'[u_1', \dots, u_n'] \colon Y$$

*Proof.* Yet more structural inductions. □

## 3.2 The Categorical Description of 2-$\lambda$

In this paragraph we present the author's original motivation for the theory 2-$\lambda$, which justifies the equations of section 3.1. It is based on Seely's description of the $\lambda$-calculus as a 2-category [16]. This motivation uses some fairly delicate notions from the theory of 2-categories. The reader who is unfamiliar with this material can safely skip the rest of this section, as neither the results nor the methods will be used in the rest of the paper.

The 2-categorical objects which occur here are either *strict* or *lax*, but not *pseudo*. We will therefore stick to the "Australian" terminology, where everything is preserved on the nose unless otherwise qualified, and the word strict is used only for emphasis.

The theory of 2-categories has several notions of adjunction (see, for example, [10]) of which we shall need the following:

**Definition.** A **2-natural adjunction** consists of the following data:

- Two 2-categories $C$ and $D$,
- Two (strict) 2-functors $F, G \colon C \to D$,
- Two 2-natural transformations $\sigma \colon F \Rightarrow G$ and $\tau \colon G \Rightarrow F$, and
- Two modifications $\eta \colon 1_F \to \tau\sigma$ and $\epsilon \colon \sigma\tau \to 1_G$,

satisfying the triangle laws:

$$\epsilon\sigma \circ \sigma\eta = 1_\sigma$$
$$\tau\epsilon \circ \eta\tau = 1_\tau$$

In this case we say that $\sigma$ is **naturally left adjoint** to $\tau$.

**Definition.** Let $F: C \to D$ be a 2-functor. A **lax right adjoint** to $F$ assigns to each object $Y$ of $D$ the following:

- An object $G(Y)$ of $C$ and
- Two 2-natural transformations $\sigma(Y): C(\_, G(Y)) \Rightarrow D(F(\_), Y)$ and $\tau(Y): D(F(\_), Y) \Rightarrow C(\_, G(Y))$,

such that $\sigma(Y)$ is naturally left adjoint to $\tau(Y)$.

**Definition.** Let $C$ be a 2-category with finite products (in the enriched sense). We say $C$ has **lax exponentials** if for each object $X$ the 2-functor $X \times \_: C \to C$ has a lax right adjoint.

**Lemma 4.** Let $C$ be a 2-category with finite products, and $X, X'$ be objects of $C$. If $X \times \_$ and $X' \times \_$ have lax right adjoints, then so does $X \times X' \times \_$.

*Proof.* Let $X \times \_$ have lax right adjoint $G^X, \sigma^X, \tau^X$ etc. Then

$$\begin{aligned} G^{X \times X'}(Y) &= G^{X'}(G^X(Y)) \\ \sigma^{X \times X'}(Y)_Z &= \sigma^X(Y)_{X' \times Z} \sigma^{X'}(G^X(Y))_Z \\ \tau^{X \times X'}(Y)_Z &= \tau^{X'}(G^X(Y))_Z \tau^X(Y)_{X' \times Z} \end{aligned}$$

defines a lax right adjoint to $X \times X' \times \_$. □

This 2-category theory is related to the theory 2-$\lambda$ by a 2-categorical version of the Lambek-Lawvere correspondence. We associate a 2-category with 2-$\lambda$ as follows:

**Definition.** The 2-category $\Lambda$ is defined by

- The objects are contexts $\Gamma$.
- The arrows are lists of terms $[t_1, \ldots, t_n]: \Gamma \to X_1, \ldots, X_n$, where $\Gamma \vdash t_j: X_j$.
- The 2-cells are lists of equivalence classes of rewrites

  $$[\gamma_1, \ldots, \gamma_n]: [t_1, \ldots, t_n] \Rightarrow [u_1, \ldots, u_n]: \Gamma \to X_1, \ldots, X_n$$

  where $\Gamma \vdash \gamma_j: t_j \Rightarrow u_j: X_j$;
- Two rewrites $\gamma$ and $\delta: t \Rightarrow u: \Gamma \to X$ are equivalent iff $\Gamma \vdash \gamma = \delta: t \Rightarrow u: X$.

- Horizontal composition of $[t_1, \dots, t_n] \colon \Delta \to E$ and $[u_1, \dots, u_m] \colon \Gamma \to \Delta$ is

$$[t_1[u_1, \dots, u_m], \dots, t_n[u_1, \dots, u_m]] \colon \Gamma \to E$$

- Vertical composition of $[\gamma_1, \dots, \gamma_n] \colon [t_1, \dots, t_n] \Rightarrow [u_1, \dots, u_n]$ and $[\delta_1, \dots, \delta_n] \colon [u_1, \dots, u_n] \Rightarrow [v_1, \dots, v_n]$ is

$$[\gamma_1; \delta_1, \dots, \gamma_n; \delta_n] \colon [t_1, \dots, t_n] \Rightarrow [v_1, \dots, v_n]$$

**Proposition 5.** $\Lambda$ is a 2-category with finite products and lax exponentials.

*Proof.* That $\Lambda$ is a 2-category amounts to checking various axioms, all of which are either immediate or appear in lemmas 1–3.

Products are defined by concatenation of contexts, projections are variables and universal arrows are given by concatenations of lists.

In view of lemma 4, it is enough to give a lax right adjoint to $X \times \_$. This is defined by

$$\begin{aligned} G^X(Y_1, \dots, Y_m) &= X \to Y_1, \dots, X \to Y_m \\ \sigma^X(Y_1, \dots, Y_m)_\Gamma([t_1, \dots, t_m]) &= [t_1^1\, 1, \dots, t_m^1\, 1] \\ \tau^X(Y_1, \dots, Y_m)_\Gamma([t_1, \dots, t_m]) &= [\lambda t_1, \dots, \lambda t_m] \\ \eta^X(Y_1, \dots, Y_m)_\Gamma([t_1, \dots, t_m]) &= [\eta_{t_1}, \dots, \eta_{t_m}] \\ \epsilon^X(Y_1, \dots, Y_m)_\Gamma([t_1, \dots, t_m]) &= [\beta_{t_1^2,1}, \dots, \beta_{t_m^2,1}] \end{aligned}$$

Again, all the work has been done in the lemmas. □

**Theorem 6.** $\Lambda$ is the universal (free) 2-category with finite products and lax exponentials on the set $\mathbb{B}$ of basic types.

*Proof.* Let $\mathcal{C}$ be a 2-category with finite products and lax exponentials, the lax right adjoint to $X \times \_$ being given by $G^X$, $\sigma^X$, $\tau^X$, $\eta^X$ and $\epsilon^X$. For each $B \in \mathbb{B}$ let $B_C$ be an object of $\mathcal{C}$. We construct a 2-functor $\mathcal{F} \colon \Lambda \to \mathcal{C}$ which preserves finite products and lax exponentials as follows:

$$\begin{aligned} \mathcal{F}(X_1, \dots, X_n) &= \mathcal{F}(X_1) \times \cdots \times \mathcal{F}(X_n) \\ \mathcal{F}(X \to Y) &= G^{\mathcal{F}(X)}(\mathcal{F}(Y)) \\ \mathcal{F}(B) &= B_C \end{aligned}$$

$$\begin{aligned} \mathcal{F}[t_1, \dots, t_n] &= \langle \mathcal{F}(t_1), \dots, \mathcal{F}(t_n) \rangle \\ \mathcal{F}(j) &= \pi_j \\ \mathcal{F}(\lambda t) &= \tau^{\mathcal{F}(X)}(\mathcal{F}(Y))_{\mathcal{F}(\Gamma)}(\mathcal{F}(t)) \quad \text{where } X, \Gamma \vdash t \colon Y \\ \mathcal{F}(t\, u) &= \sigma^{\mathcal{F}(X)}(\mathcal{F}(Y))_{\mathcal{F}(\Gamma)}(\mathcal{F}(t)) \circ \langle \mathcal{F}(u), 1 \rangle \quad \text{where } \Gamma \vdash t \colon X \to Y \end{aligned}$$

$$
\begin{aligned}
\mathcal{F}[\gamma_1, \ldots, \gamma_n] &= \langle \mathcal{F}(\gamma_1), \ldots, \mathcal{F}(\gamma_n) \rangle \\
\mathcal{F}(j) &= 1_{\pi_j} \\
\mathcal{F}(\lambda\gamma) &= \tau^{\mathcal{F}(X)}(\mathcal{F}(Y))_{\mathcal{F}(\Gamma)}(\mathcal{F}(\gamma)) \quad \text{where } X, \Gamma \vdash \gamma \colon t \Rightarrow t' \colon Y \\
\mathcal{F}(\gamma\,\delta) &= \sigma^{\mathcal{F}(X)}(\mathcal{F}(Y))_{\mathcal{F}(\Gamma)}(\mathcal{F}(\gamma)) \circ \langle \mathcal{F}(\delta), 1 \rangle \\
&\quad \text{where } \Gamma \vdash \gamma \colon t \Rightarrow t' \colon X \to Y \\
\mathcal{F}(\gamma; \delta) &= \mathcal{F}(\gamma); \mathcal{F}(\delta) \\
\mathcal{F}(\eta_t) &= \eta^{\mathcal{F}(X)}(\mathcal{F}(Y))_{\mathcal{F}(\Gamma)}(\mathcal{F}(t)) \quad \text{where } \Gamma \vdash t \colon X \to Y \\
\mathcal{F}(\beta_{t,u}) &= \epsilon^{\mathcal{F}(X)}(\mathcal{F}(Y))_{\mathcal{F}(\Gamma)}(\mathcal{F}(t)) \circ \langle \mathcal{F}(u), 1 \rangle \quad \text{where } \Gamma \vdash t \colon X \to Y
\end{aligned}
$$

It is straightforward to check that this is well-defined. Uniqueness is immediate because each of the clauses above must be true if $\mathcal{F}$ is to preserve finite products and lax exponentials. □

# 4 The Canonical Form of Rewrites

This section is the heart of the paper. By analogy with proof theory, we view the composition (;) of rewrites as a cut rule, and prove a kind of ;-elimination theorem. This operation of ;-elimination preserves the equations of the theory 2-$\lambda$; we show that conversely, it maps equated rewrites to identical ones.

Although this process of ;-elimination does not remove all ;s, it does map all rewrites to a syntactically simple kind we call **canonical form**. This solves the word problem for 2-$\lambda$: two rewrites are equated in 2-$\lambda$ if, and only if, they have the same canonical form. This result is particularly useful when proving properties of the theory.

**Definition (Canonical Form).** In order to define the canonical form, we identify three special classes of rewrites. Let $\mathcal{A}$ be defined inductively as follows:

- Every identity rewrite is in $\mathcal{A}$.
- If $\Gamma \vdash \alpha_1 \colon t_1 \Rightarrow \lambda t \colon X \to Y$ and $\Gamma \vdash \alpha_2 \colon t[u] \Rightarrow t_2 \colon Y$ are in $\mathcal{A}$, then $\Gamma \vdash \alpha_1\, u; \beta_{t,u}; \alpha_2 \colon t_1\, u \to t_2 \colon Y$ is in $\mathcal{A}$.

let $\mathcal{E}$ be defined inductively by:

- Every identity rewrite is in $\mathcal{E}$.
- If $\Gamma \vdash \epsilon_1 \colon t_1 \Rightarrow t \colon X \to Y$ and $X, \Gamma \vdash \epsilon_2 \colon 1 \Rightarrow t_2 \colon X$ and $X, \Gamma \vdash \epsilon_3 \colon t^1\, t_2 \Rightarrow t_3 \colon Y$ are in $\mathcal{E}$, then $\Gamma \vdash \epsilon_1; \eta_t; \lambda(t^1\, \epsilon_2; \epsilon_3) \colon t_1 \Rightarrow \lambda t_3 \colon X \to Y$ is in $\mathcal{E}$.

and let $\mathcal{G}$ be defined inductively by:

- If $\Gamma \vdash \alpha\colon t \Rightarrow j\colon X$ is in $\mathcal{A}$ and $\Gamma \vdash \epsilon\colon j \Rightarrow u\colon X$ is in $\mathcal{E}$ then $\Gamma \vdash \alpha; j; \epsilon\colon t \Rightarrow u\colon X$ is in $\mathcal{G}$.
- If $\Gamma \vdash \alpha\colon t \Rightarrow \lambda t_1\colon X \to Y$ is in $\mathcal{A}$, $\Gamma \vdash \epsilon\colon \lambda u_1 \Rightarrow u\colon X \to Y$ is in $\mathcal{E}$ and $X, \Gamma \vdash \gamma\colon t_1 \Rightarrow u_1\colon Y$ is in $\mathcal{G}$ then $\Gamma \vdash \alpha; \lambda\gamma; \epsilon\colon t \Rightarrow u\colon X \to Y$ is in $\mathcal{G}$.
- If $\Gamma \vdash \alpha\colon t \Rightarrow t_1\, u_1\colon Y$ is in $\mathcal{A}$, $\Gamma \vdash \epsilon\colon t_2\, u_2 \Rightarrow u\colon Y$ is in $\mathcal{E}$, and $\Gamma \vdash \gamma\colon t_1 \Rightarrow t_2\colon X \to Y$ and $\Gamma \vdash \delta\colon u_1 \Rightarrow u_2\colon X$ are in $\mathcal{G}$, then $\Gamma \vdash \alpha; \gamma\, \delta; \epsilon\colon t \Rightarrow u\colon Y$ is in $\mathcal{G}$.

A rewrite is in **canonical form** when it is in $\mathcal{G}$.

The notation $\gamma_1; \gamma_2; \gamma_3$ is shorthand for $(\gamma_1; \gamma_2); \gamma_3$. (Of course, the choice of left rather than right bracketing is arbitrary, as long as we are consistent.)

This canonical form is not meant to have anything to do with efficiency of implementation, nor should it be understood as an evaluation strategy: it is simply a formal device for studying the theory 2-$\lambda$. Nonetheless, we can make some attempt to describe it informally.

- A rewrite in $\mathcal{A}$ is a sequence of $\beta$-reductions at the top level. However, the first term of a $\beta$-redex must be a $\lambda$-abstraction: a sequence of top-level $\beta$-reductions to this term will achieve this.
- A rewrite in $\mathcal{E}$ is a sequence of $\eta$-expansions at the top level. However, $\eta$-expansions create new subterms, which can themselves be expanded.
- A rewrite in $\mathcal{G}$ takes the form: first all the top-level $\beta$-reductions, last all the top-level $\eta$-expansions, and in between, all the rewrites of the subterms.

**Lemma 7.** The following results are no more than observations; they are recorded here so that we can use them without further comment.

1. Every rewrite in $\mathcal{G}$ is of the form $\alpha; \delta; \epsilon$, where $\alpha \in \mathcal{A}$, $\epsilon \in \mathcal{E}$ and $\delta$ is either $j$, $\lambda\gamma_1$ or $\gamma_1\, \gamma_2$.
2. If $\Gamma \vdash \alpha\colon t \Rightarrow t'\colon X$ in $\mathcal{A}$ is not an identity rewrite, then $t = t_1\, t_2$ for some $t_1$, $t_2$.
3. If $\Gamma \vdash \epsilon\colon t \Rightarrow t'\colon X$ in $\mathcal{E}$ is not an identity rewrite, then $t' = \lambda t_1$ for some $t_1$.

4. If $\Gamma \vdash \gamma \colon 1 \Rightarrow t \colon X$ is in $\mathcal{G}$ then $\gamma = 1; 1; \epsilon$ for some $\Gamma \vdash \epsilon \colon 1 \Rightarrow t \colon X$ in $\mathcal{E}$.

In general, identity rewrites are not members of $\mathcal{G}$. The following definition describes those canonical rewrites which play the role of identities.

**Definition.** For each term $t$ we define $\mathcal{I}(t)$ inductively as follows:

$$\begin{aligned} \mathcal{I}(j) &= j; j; j \\ \mathcal{I}(\lambda t) &= \lambda t; \lambda \mathcal{I}(t); \lambda t \\ \mathcal{I}(t\, u) &= t\, u; \mathcal{I}(t)\, \mathcal{I}(u); t\, u \end{aligned}$$

This definition is justified by the following lemma.

**Lemma 8.** If $\Gamma \vdash t \colon X$ then

1. $\Gamma \vdash \mathcal{I}(t) \colon t \Rightarrow t \colon X$ is in $\mathcal{G}$
2. $\Gamma \vdash \mathcal{I}(t) = t \colon t \Rightarrow t \colon X$.

*Proof.* Structural induction. □

Substitution of rewrites in $\mathcal{G}$ takes a different form from either of the substitutions of rewrites defined so far.

**Definition.** If $\Gamma \vdash \delta_j \colon u_j \Rightarrow u'_j \colon X_j$ for $j = 1 \ldots n$, then:

$$\begin{aligned} (\alpha; j; \epsilon)[\delta_1, \ldots, \delta_n] &= \alpha[u_1, \ldots, u_n]; ; \delta_j; ; \epsilon[u'_1, \ldots, u'_n] \\ (\alpha; \lambda\gamma; \epsilon)[\delta_1, \ldots, \delta_n] &= \alpha[u_1, \ldots, u_n]; \lambda(\gamma[1, \delta^1_1, \ldots, \delta^1_n]); \epsilon[u'_1, \ldots, u'_n] \\ (\alpha; \gamma_1\, \gamma_2; \epsilon)[\delta_1, \ldots, \delta_n] &= \alpha[u_1, \ldots, u_n]; \gamma_1[\delta_1, \ldots, \delta_n]\, \gamma_2[\delta_1, \ldots, \delta_n]; \epsilon[u'_1, \ldots, u'_n] \end{aligned}$$

The relationship of this form with the other two is given by the following lemma.

**Lemma 9.** Let $X_1, \ldots, X_n \vdash \gamma \colon t \Rightarrow t' \colon X$ and $\Gamma \vdash \delta_j \colon u_j \Rightarrow u'_j \colon X_j$ for $j = 1 \ldots n$ be rewrites in $\mathcal{G}$. Then

1. $\Gamma \vdash \gamma[\delta_1, \ldots, \delta_n] \colon t[u_1, \ldots, u_n] \Rightarrow t'[u'_1, \ldots, u'_n] \colon X$ is in $\mathcal{G}$
2. $\Gamma \vdash \gamma[\delta_1, \ldots, \delta_n] = \gamma[u_1, \ldots, u_n]; t'[\delta_1, \ldots, \delta_n]$
   $\colon t[u_1, \ldots, u_n] \Rightarrow t'[u'_1, \ldots, u'_n] \colon X$
3. $\Gamma \vdash \gamma[\delta_1, \ldots, \delta_n] = t[\delta_1, \ldots, \delta_n]; \gamma[u'_1, \ldots, u'_n]$
   $\colon t[u_1, \ldots, u_n] \Rightarrow t'[u'_1, \ldots, u'_n] \colon X$

*Proof.* Structural induction. □

The heart of the proof of the canonical form theorem is the definition of sequential composition of rewrites as a binary operation on $\mathcal{G}$. The composition of rewrites in $\mathcal{E}$ and $\mathcal{A}$ is straightforward; we use the symbol '$;;$' defined as follows, with the convention that $\alpha \in \mathcal{A}$, $\epsilon \in \mathcal{E}$ and $\alpha;\delta;\epsilon \in \mathcal{G}$.

$$\begin{aligned} t;;\alpha &= \alpha \quad \text{where } t \text{ is an identity} \\ (\alpha_1 u; \beta_{t,u}; \alpha_2);;\alpha &= \alpha_1 u; \beta_{t,u}; (\alpha_2;;\alpha) \end{aligned}$$

$$\begin{aligned} \epsilon;;t &= \epsilon \quad \text{where } t \text{ is an identity} \\ \epsilon;;(\epsilon_1;\eta_t;\lambda(t^1\,\epsilon_2;\epsilon_3)) &= (\epsilon;;\epsilon_1);\eta_t;\lambda(t^1\,\epsilon_2;\epsilon_3) \end{aligned}$$

$$\begin{aligned} \alpha;;(\alpha';\delta;\epsilon) &= (\alpha;;\alpha');\delta;\epsilon \\ (\alpha;\delta;\epsilon');;\epsilon &= \alpha;\delta;(\epsilon';;\epsilon) \end{aligned}$$

Note that $;;$ is associative (in every possible way) and that identities in $\mathcal{A}$ and $\mathcal{E}$ are identities of $;;$.

**Definition.** If $\gamma_1$ and $\gamma_2$ are rewrites in $\mathcal{G}$, their **composition** $\gamma_1 \dagger \gamma_2$ is defined by syntactic cases:

$$\begin{aligned} \alpha;j;j \dagger j;j;\epsilon &= \alpha;j;\epsilon \\ \alpha;\lambda\gamma_1;\lambda t \dagger \lambda t;\lambda\gamma_2;\epsilon &= \alpha;\lambda(\gamma_1 \dagger \gamma_2);\epsilon \\ \alpha;\gamma_{11}\,\gamma_{12};t\,u \dagger t\,u;\gamma_{21}\,\gamma_{22};\epsilon &= \alpha;(\gamma_{11} \dagger \gamma_{21})\,(\gamma_{12} \dagger \gamma_{22});\epsilon \end{aligned}$$

$$\gamma_1;\eta_{t_1};\lambda(t_1^1\,\epsilon_{11};\epsilon_{12}) \dagger \lambda t_2;\lambda\gamma_2;\epsilon_2 = \begin{cases} \gamma_1 \dagger \gamma_3;\eta_{t_3};\lambda(t_3^1\,\epsilon_{31};\epsilon_{32});;\epsilon_2 \\ \qquad \text{if } t_1^1\,1;\mathcal{I}(t_1^1)\,(1;1;\epsilon_{11});\epsilon_{12} \dagger \gamma_2 = t_1^1\,1;\gamma_3^1\,(1;1;\epsilon_{31});\epsilon_{32} \\ \gamma_1 \dagger \alpha_3;\lambda\gamma_3;\epsilon_2 \\ \qquad \text{if } t_1^1\,1;\mathcal{I}(t_1^1)\,(1;1;\epsilon_{11});\epsilon_{12} \dagger \gamma_2 = \alpha_3^1\,1;\beta_{t_3,1};\gamma_3 \end{cases}$$

$$\alpha_1;\gamma_{11}\,\gamma_{12};t_{21}\,t_{22} \dagger \alpha_{21}\,t_{22};\beta_{t_{23},t_{22}};\gamma_2 = \begin{cases} \alpha_1;;\alpha_3\,t_{12};\beta_{t_3,t_{12}};\gamma_3[\gamma_{12}] \dagger \gamma_2 \\ \qquad \text{if } \gamma_{11} \dagger \alpha_{21};\lambda\mathcal{I}(t_{23});\lambda t_{23} = \alpha_3;\lambda\gamma_3;\lambda t_{23} \\ \alpha_1;\gamma_3\,(\gamma_{12};;\epsilon_{31}[t_{22}]);\epsilon_{32}[t_{22}] \dagger \gamma_2 \\ \qquad \text{if } \gamma_{11} \dagger \alpha_{21};\lambda\mathcal{I}(t_{23});\lambda t_{23} = \gamma_3;\eta_{t_3};\lambda(t_3^1\,\epsilon_{31};\epsilon_{32}) \end{cases}$$

This definition is not easy to motivate intuitively, although we will show formally that it is correct. The following lemma shows that † is the operation which finds the canonical form of the sequential composition $\gamma_1; \gamma_2$ of two canonical rewrites $\gamma_1$ and $\gamma_2$. By analogy with proof theory, this should be thought of as the 'elimination' of the ; (which is analogous to a cut).

**Lemma 10.** If $\Gamma \vdash \gamma_1 \colon t_1 \Rightarrow t_2 \colon X$ and $\Gamma \vdash \gamma_2 \colon t_2 \Rightarrow t_3 \colon X$ are in $\mathcal{G}$, then

1. $\gamma_1 \dagger \gamma_2$ is well-defined and $\Gamma \vdash \gamma_1 \dagger \gamma_2 \colon t_1 \Rightarrow t_3 \colon X$ is in $\mathcal{G}$,

2. $\Gamma \vdash \gamma_1 \dagger \gamma_2 = \gamma_1; \gamma_2 \colon t_1 \Rightarrow t_3 \colon X$.

*Proof.* (1) That the clauses defining † are exhaustive follows from lemma 7. The well-foundedness of the recursion is slightly more complicated than the simple structural inductions considered so far; we define a measure $|\gamma|_{\mathcal{G}}$ on $\mathcal{G}$ and $|\alpha|_{\mathcal{A}}$ on $\mathcal{A}$ as follows:

$$\begin{aligned}
|\alpha; j; \epsilon|_{\mathcal{G}} &= |\alpha|_{\mathcal{A}} + 1 \\
|\alpha; \lambda\gamma; \epsilon|_{\mathcal{G}} &= |\alpha|_{\mathcal{A}} + |\gamma|_{\mathcal{G}} + 1 \\
|\alpha; \gamma\,\delta; \epsilon|_{\mathcal{G}} &= |\alpha|_{\mathcal{A}} + |\gamma|_{\mathcal{G}} + |\delta|_{\mathcal{G}} + 1 \\
|t|_{\mathcal{A}} &= 0 \\
|\alpha_1\, u; \beta_{t,u}; \alpha_2|_{\mathcal{A}} &= |\alpha_1|_{\mathcal{A}} + |\mathcal{I}(t)|_{\mathcal{G}} + |\alpha_2|_{\mathcal{A}} + 1
\end{aligned}$$

and use the inductive hypothesis on $n$ that:

- $\gamma_1 \dagger \gamma_2$ is well defined for all composable $\gamma_1, \gamma_2 \in \mathcal{G}$ such that $|\gamma_2|_{\mathcal{G}} < n$, and

- If $|\gamma_2|_{\mathcal{G}} < n$ then $|(t\,1; \mathcal{I}(t)\,(1; 1; \epsilon_1); \epsilon_2) \dagger \gamma_2|_{\mathcal{G}} < n$ for all $\epsilon_1, \epsilon_2 \in \mathcal{E}$ which make the composition defined.

The proof is then straightforward.

(2) This is a fairly straightforward induction, which amounts to justifying the clauses in the definition of † using the rules (id-l–triangle2) of section 3.1. □

**Lemma 11.** Basic facts relating the various operations on $\mathcal{G}$.

1. The compositions ;; and † agree: If $\Gamma \vdash \alpha \colon t_1 \Rightarrow t_2 \colon X$ is in $\mathcal{A}$, and $\Gamma \vdash \gamma_1 \colon t_2 \Rightarrow t_3 \colon X$ and $\Gamma \vdash \gamma_2 \colon t_3 \Rightarrow t_4 \colon X$ are in $\mathcal{G}$, then

$$\alpha;;(\gamma_1 \dagger \gamma_2) = (\alpha;;\gamma_1) \dagger \gamma_2$$

2. Similarly: If $\Gamma \vdash \gamma_1 \colon t_1 \Rightarrow t_2 \colon X$ and $\Gamma \vdash \gamma_2 \colon t_2 \Rightarrow t_3 \colon X$ are in $\mathcal{G}$, and $\Gamma \vdash \epsilon \colon t_3 \Rightarrow t_4 \colon X$ is in $\mathcal{E}$, then

$$(\gamma_1 \dagger \gamma_2);;\epsilon = \gamma_1 \dagger (\gamma_2;;\epsilon)$$

3. Rewrites $\mathcal{I}(t)$ are identities for $\dagger$: If $\Gamma \vdash \gamma \colon t \Rightarrow u \colon X$ is in $\mathcal{G}$ then

$$\mathcal{I}(t) \dagger \gamma = \gamma = \gamma \dagger \mathcal{I}(u)$$

4. The interchange law for $\dagger$: If $X_1, \ldots, X_n \vdash \gamma \colon t \Rightarrow t' \colon X$ and $X_1, \ldots, X_n \vdash \gamma' \colon t' \Rightarrow t'' \colon X$ and $\Gamma \vdash \delta_j \colon u_j \Rightarrow u'_j \colon X_j$ and $\Gamma \vdash \delta'_j \colon u'_j \Rightarrow u''_j \colon X_j$ are in $\mathcal{G}$ for $j = 1 \ldots n$, then

$$(\gamma \dagger \gamma')[\delta_1 \dagger \delta'_1, \ldots, \delta_n \dagger \delta'_n] = \gamma[\delta_1, \ldots, \delta_n] \dagger \gamma'[\delta'_1, \ldots, \delta'_n]$$

*Proof.* Straightforward inductions, using the complexity measure $|\gamma|_{\mathcal{G}}$ of lemma 10. □

The final result we need before proving the canonical form theorem is the associativity of $\dagger$:

**Proposition 12.** If $\Gamma \vdash \gamma_1 \colon t_1 \Rightarrow t_2 \colon X$ and $\Gamma \vdash \gamma_2 \colon t_2 \Rightarrow t_3 \colon X$ and $\Gamma \vdash \gamma_3 \colon t_3 \Rightarrow t_4 \colon X$ are in $\mathcal{G}$, then $(\gamma_1 \dagger \gamma_2) \dagger \gamma_3 = \gamma_1 \dagger (\gamma_2 \dagger \gamma_3)$.

The proof of this proposition is long and technical, with a large number of cases. It has been moved to an appendix.

Using $\dagger$, we now formally define the canonical form of a general rewrite.

**Definition.** If $\Gamma \vdash \gamma \colon t \Rightarrow u \colon X$ is any rewrite, we define $\Gamma \vdash \mathcal{C}(\gamma) \colon t \Rightarrow u \colon X$ in $\mathcal{G}$ as follows:

$$\begin{aligned}
\mathcal{C}(j) &= j; j; j \\
\mathcal{C}(\lambda\gamma) &= \lambda t; \lambda\mathcal{C}(\gamma); \lambda u \quad \text{where } X, \Gamma \vdash \gamma \colon t \Rightarrow u \colon Y \\
\mathcal{C}(\gamma_1\, \gamma_2) &= t_1\, t_2; \mathcal{C}(\gamma_1)\, \mathcal{C}(\gamma_2); u_1\, u_2 \quad \text{where } \Gamma \vdash \gamma_j \colon t_j \Rightarrow u_j \colon X_j \\
\mathcal{C}(\gamma_1; \gamma_2) &= \mathcal{C}(\gamma_1) \dagger \mathcal{C}(\gamma_2) \\
\mathcal{C}(\eta_t) &= \mathcal{I}(t);; t; \eta_t; \lambda(t^1\, 1; t^1\, 1) \\
\mathcal{C}(\beta_{t,u}) &= (\lambda t)\, u; \beta_{t,u}; t[u];; \mathcal{I}(t[u])
\end{aligned}$$

This definition is justified by the following proposition.

**Proposition 13.** If $\Gamma \vdash \gamma \colon t \Rightarrow u \colon X$ then

1. $\Gamma \vdash \mathcal{C}(\gamma) \colon t \Rightarrow u \colon X$ is in $\mathcal{G}$

2. $\Gamma \vdash \mathcal{C}(\gamma) = \gamma \colon t \Rightarrow u \colon X$.

*Proof.* Straightforward induction. The work has already been done in lemmas 8–10. □

Finally, we show that $\mathcal{C}$ preserves the equalities of the theory 2-$\lambda$.

**Proposition 14.** If $\Gamma \vdash \gamma = \delta \colon t \Rightarrow u \colon X$ then $\mathcal{C}(\gamma) = \mathcal{C}(\delta)$.

*Proof.* By induction on the length of the derivation of $\Gamma \vdash \gamma = \delta \colon t \Rightarrow u \colon X$. The hard cases have already been done in lemmas 11–12. □

**Theorem 15 (The Canonical Form Theorem).** The set $\mathcal{G}$ contains exactly one member of each equivalence class of the rewrites quotiented by the theory 2-$\lambda$.

*Proof.* This follows immediately from the last two results. □

**Corollary 16.** The theory 2-$\lambda$ is consistent, in the sense that it does not identify everything possible.

*Proof.* By the theorem, it suffices to give two distinct elements of $\mathcal{G}$ with the same source and target. We give two different examples:

- If $X, \Gamma \vdash t \colon Y$ then

  $$\Gamma \vdash \lambda(\lambda t^2\, 1); \lambda\mathcal{C}(\beta_{t^2,1}); \lambda t; \eta_{\lambda t}; \lambda(\lambda t^2\, 1; \lambda t^2\, 1) \\ \colon \lambda(\lambda t^2\, 1) \Rightarrow \lambda(\lambda t^2\, 1) \colon X \to Y$$

  is in $\mathcal{G}$, but it is not equal to $\mathcal{I}(\lambda(\lambda t^2\, 1))$.

- Let $I = \lambda 1$. Then

  $$X \vdash I\,(I\,1); \mathcal{I}(I)\,\mathcal{C}(\beta_{1,1}); I\,1 \colon I\,(I\,1) \Rightarrow I\,1 \colon X$$

  and

  $$X \vdash \mathcal{C}(\beta_{1,I\,1}) \colon I\,(I\,1) \Rightarrow I\,1 \colon X$$

  are both in $\mathcal{G}$, but they are not equal.

□

# 5 The 2$\lambda$-Calculus as a Rewrite System

In this section we use the canonical form of rewrites to investigate the 2$\lambda$-calculus as a rewrite system. First we investigate the relationship between equality of rewrites in 2-$\lambda$ and 'strongly equivalent' reductions of the $\lambda$-calculus. Next we generalise the 'strong Church-Rosser' theorem to our setting, and prove a related property we call 'mellifluence'. Finally we characterise the long-$\beta\eta$-normal forms in terms of the properties of the rewrites starting from them. These results provide evidence for the claim that the 2$\lambda$-calculus is not just a formalism for talking about rewrites, but also has something to say about the process of reduction.

From this point on, we consider rewrites up to equivalence. The formalism of the first part of the paper has done its job, and we no longer need the notion of syntactic equality. We can assume that any rewrite is in $\mathcal{G}$ even though we will use the rules of 2-$\lambda$ to reason about them, and write $\beta_{t,u}$ instead of $\lambda t\, u; \beta_{t,u}; t[u]; ; \mathcal{I}(t[u])$ and $\gamma; \delta$ instead of $\gamma \dagger \delta$. The more pedantic reader can insert $\mathcal{C}$ at every appropriate point.

## 5.1 Strongly Equivalent Reductions

The theory 2-$\lambda$ is closely related to Lévy's notion of strongly equivalent reductions, defined as follows (see [14] or [2, chapter 12]).

> Two sequences of $\beta$-reductions $\sigma$ and $\rho$ in the $\lambda$-calculus from the same term are **strongly equivalent** if the residuals $\sigma/\rho$ and $\rho/\sigma$ are both empty.

This definition does not extend to an equivalence relation in the presence of $\eta$-expansion, since

$$\eta_t/\eta_t = \lambda(\eta_{t^1}\, 1)$$

so the relation is not reflexive. However, for $\beta$-reductions, the two theories agree, as we now demonstrate.

**Proposition 17.** The $\beta$-reductions of the simply-typed $\lambda$-calculus, quotiented by strong equivalence, are in bijective correspondence with those rewrites in $\mathcal{G}$ which contain no $\eta$s.

*Proof.* In [2, chapter 12] it is shown that every finite $\beta$-reduction is strongly equivalent to a unique *standard* reduction. Therefore we need only show a bijection between standard reductions and canonical forms.

A standard reduction is one in 'leftmost-outermost' order. We define a map $\mathcal{S}$ from canonical forms with no $\eta$s to standard reductions (using the notation of [2]) as follows:

$$\begin{aligned}
\mathcal{S}(\alpha; j; j) &= \mathcal{S}(\alpha) \\
\mathcal{S}(\alpha; \lambda\gamma; \lambda t) &= \mathcal{S}(\alpha) + \lambda\mathcal{S}(\gamma) \\
\mathcal{S}(\alpha; \gamma_1\, \gamma_2; t'\, u') &= \mathcal{S}(\alpha) + (\mathcal{S}(\gamma_1)\, u) + (t'\, \mathcal{S}(\gamma_2)) \\
\mathcal{S}(t) &= \emptyset \\
\mathcal{S}(\alpha_1\, u; \beta_{t,u}; \alpha_2) &= (\mathcal{S}(\alpha_1)\, u) + \beta_{t,u} + \mathcal{S}(\alpha_2)
\end{aligned}$$

This is clearly a bijection: it just sequentialises the reductions in a standard way. □

The theory 2-$\lambda$ can therefore be seen as a way of extending the notion of strong equivalence to the $\lambda$-calculus with $\eta$-expansion. The triangle laws (triangle1) and (triangle2) mean that it is not just a simple-minded adaptation of Lévy's definition, but has rather more structure.

## 5.2 Strong Confluence

The most important concept of rewriting theory is that of confluence/the Church-Rosser theorem/the diamond property: that if $\gamma_1\colon t \Rightarrow u_1$ and $\gamma_2\colon t \Rightarrow u_2$ are two rewrites with a common source then there exist $\delta_1\colon u_1 \Rightarrow v$ and $\delta_2\colon u_2 \Rightarrow v$ with common target. This is proved for the $\lambda$-calculus with $\eta$-expansion in [13]. The definition of strongly-equivalent reductions in the $\lambda$-calculus gives rise to the *strong* Church-Rosser theorem/*commuting* diamond property: that $\delta_1$ and $\delta_2$ can be chosen so that $\gamma_1; \delta_1$ and $\gamma_2; \delta_2$ are strongly equivalent (see [2, chapter 12]). In this section we extend this result to the 2$\lambda$-calculus.

The first lemma we prove says that if two rewrites in $\mathcal{A}$ have a common source, one is a prefix of the other:

**Lemma 18.** If $\alpha_1\colon t \Rightarrow u_1$ and $\alpha_2\colon t \Rightarrow u_2$ are in $\mathcal{A}$ then either there exists $\alpha_3\colon u_1 \Rightarrow u_2$ in $\mathcal{A}$ such that $\vdash \alpha_1; \alpha_3 = \alpha_2$ or vice versa.

*Proof.* By induction on the structure of $\alpha_1$ and $\alpha_2$.

case 1: $\alpha_1 = t$. Then $u_1 = t$, take $\alpha_3 = \alpha_2$.

case 2: $\alpha_2 = t$. Then $u_2 = t$, take $\alpha_3 = \alpha_1$.

case 3: $\alpha_j = \alpha_{j1}\, t_2; \beta_{v_j,t_2}; \alpha_{j2}$ where $t = t_1\, t_2$, $\alpha_{j1}\colon t_1 \Rightarrow \lambda v_j$.

Apply the inductive hypothesis to $\alpha_{11}, \alpha_{21}$ to get (without loss of generality) $\alpha_{31}\colon \lambda v_1 \Rightarrow \lambda v_2$. But any rewrite in $\mathcal{A}$ with source a $\lambda$-term is identity, so $v_1 = v_2$ and $\alpha_{11} = \alpha_{21}$. The result then follows by applying the inductive hypothesis to $\alpha_{12}\colon v_1[t_2] \Rightarrow u_1$ and $\alpha_{22}\colon v_2[t_2] \Rightarrow u_2$. □

Next we turn our attention to rewrites in $\mathcal{E}$: they commute with any other rewrite:

**Lemma 19.** If $\epsilon\colon t \Rightarrow u_1$ is in $\mathcal{E}$ and $\gamma\colon t \Rightarrow u_2$ then there exist $\gamma'\colon u_1 \Rightarrow v$ and $\epsilon'\colon u_2 \Rightarrow v$ in $\mathcal{E}$ such that $\vdash \epsilon; \gamma' = \gamma; \epsilon'$.

*Proof.* By induction on the structure of $\epsilon$.

case 1: $\epsilon = t$. Then $u_1 = t$, take $\gamma' = \gamma$ and $\epsilon' = u_2$.

case 2: $\epsilon = \epsilon_1; \eta_w; \lambda(w^1\, \epsilon_2; \epsilon_3)$, where $\epsilon_1\colon t \Rightarrow w$, $\epsilon_2\colon 1 \Rightarrow x$ and $\epsilon_3\colon w^1\, x \Rightarrow y$.

Apply the inductive hypothesis to $\epsilon_1$ and $\gamma$ to get $\epsilon_1'$ and $\gamma_1$, then apply it to $\epsilon_3$ and $\gamma_1^1\, x$ to get $\gamma_2$ and $\epsilon_3'$. We then have

$$\begin{aligned} \epsilon; \lambda\gamma_2 &= \epsilon_1; \eta_w; \lambda(w^1\, \epsilon_2; \epsilon_3); \lambda\gamma_2 \\ &= \epsilon_1; \eta_w; \lambda(w^1\, \epsilon_2; \epsilon_3; \gamma_2) \\ &= \epsilon_1; \eta_w; \lambda(w^1\, \epsilon_2; \gamma_1^1\, x; \epsilon_3') \\ &= \epsilon_1; \gamma_1; \eta_{w'}; \lambda(w'^1\, \epsilon_2; \epsilon_3') \\ &= \gamma; \epsilon_1'; \eta_{w'}; \lambda(w'^1\, \epsilon_2; \epsilon_3') \end{aligned}$$

so take $\gamma' = \lambda\gamma_2$ and $\epsilon' = \epsilon_1'; \eta_{w'}; \lambda(w'^1\, \epsilon_2; \epsilon_3')$. □

A final lemma to say how rewrites in $\mathcal{E}$ interact with $\beta$-reductions:

**Lemma 20.** If $\epsilon\colon \lambda t \Rightarrow u$ and $\epsilon'\colon u\, s \Rightarrow v$ are in $\mathcal{E}$ then there exist $w$, $\gamma\colon v \Rightarrow w$ and $\gamma'\colon t[s] \Rightarrow w$ such that $\vdash \epsilon\, s; \epsilon'; \gamma = \beta_{t,s}; \gamma'$.

*Proof.* By induction on the structure of $\epsilon$.

case 1: $\epsilon = \lambda t$. Then $u = \lambda t$; apply lemma 19 to $\epsilon'$ and $\beta_{t,s}$.

case 2: $\epsilon = \epsilon_1; \eta_x; \lambda(x^1\, \epsilon_2; \epsilon_3)$, where $\epsilon_1\colon \lambda t \Rightarrow x$, $\epsilon_2\colon 1 \Rightarrow r$ and $\epsilon_3\colon x^1\, r \Rightarrow y$.

Apply lemma 19 to $\epsilon'$ and $\beta_{y,s}$ to get $\gamma_1\colon v \Rightarrow w_1$ and $\epsilon''\colon y[s] \Rightarrow w_1$, then apply the inductive hypothesis to $\epsilon_1$ and $\epsilon_3[s]; \epsilon''$ to get $\gamma_2\colon w_1 \Rightarrow w$

and $\gamma_3 \colon t[r[s]] \Rightarrow w$. We then have

$$\begin{aligned}\epsilon\, s; \epsilon'; \gamma_1; \gamma_2 &= \epsilon_1\, s; \eta_x\, s; \lambda(x^1\, \epsilon_2; \epsilon_3)\, s; \beta_{y,s}; \epsilon''; \gamma_2 \\ &= \epsilon_1\, s; x\, \epsilon_2[s]; \epsilon_3[s]; \epsilon''; \gamma_2 \\ &= \lambda t\, \epsilon_2[s]; \epsilon_1\, r[s]; \epsilon_3[s]; \epsilon''; \gamma_2 \\ &= \lambda t\, \epsilon_2[s]; \beta_{t,r[s]}; \gamma_3 \\ &= \beta_{t,s}; t[\epsilon_2[s]]; \gamma_3\end{aligned}$$

so take $\gamma = \gamma_1; \gamma_2$ and $\gamma' = t[\epsilon_2[s]]; \gamma_3$. □

We are now in a position to prove confluence. For this (and another) proof, some more sophisticated well-foundedness is needed: the usual proof of confluence of the $\lambda$-calculus depends upon 'finiteness of developments' [2, chapter 11]. Rather than set up all that machinery here, we use the fact that the simply-typed $\lambda$-calculus (with $\eta$-*reduction*) is strongly normalising. We write $||t||$ for the length of the longest $\beta\eta$-reduction path starting from $t$.

**Proposition 21.** If $\gamma_1 \colon t \Rightarrow u_1$ and $\gamma_2 \colon t \Rightarrow u_2$ then there exist $\gamma_1' \colon u_1 \Rightarrow v$ and $\gamma_2' \colon u_2 \Rightarrow v$ such that $\vdash \gamma_1; \gamma_1' = \gamma_2; \gamma_2'$.

*Proof.* By induction on $||t||$, the complexity of their common source term.

Let $(\alpha_j; \delta_j; \epsilon_j) = \gamma_j$ for $j = 1, 2$. Then $\alpha_1$ and $\alpha_2$ have the same domain; by lemma 18 there exists $\alpha_3$ such that (without loss of generality) $\alpha_2 = \alpha_1; \alpha_3$.

case 1: $\delta_1 = j$. Apply lemma 19 to $\epsilon_1$ and $(\alpha_3; \delta_2; \epsilon_2)$.

case 2: $\delta_1 = \lambda\gamma_{11}$, $\alpha_3 = \lambda t_1$ and $\delta_2 = \lambda\gamma_{21}$. Apply the inductive hypothesis to $\gamma_{11}$ and $\gamma_{21}$ to get $\gamma_{12}$ and $\gamma_{22}$; two applications of lemma 19 then give the answer.

case 3: $\delta_1 = \gamma_{11}\, \gamma_{12}$ and $\alpha_3 = t_1\, t_2$. Similar to case 2.

case 4: $\delta_1 = \gamma_{11}\, \gamma_{12}$ and $\alpha_3 = \alpha_{31}\, t_2; \beta_{t_3,t_2}; \alpha_{32}$.

Apply the inductive hypothesis to $\gamma_{11} \colon t_1 \Rightarrow v_1$ and $\alpha_{31} \colon t_1 \Rightarrow \lambda t_3$ to get $\gamma_3 \colon v_1 \Rightarrow w_1$ and $\gamma_4 \colon \lambda t_3 \Rightarrow w_1$, and lemma 19 to $\epsilon_1$ and $\gamma_3\, v_2$ to get $\gamma_5$ and $\epsilon_3$. The domain of $\gamma_4$ is a lambda term, so it is of the form $(\lambda t_3; \lambda\gamma_{41}; \epsilon_4)$. Apply lemma 20 to $\epsilon_4$ and $\epsilon_3$ to get $\gamma_6$ and $\gamma_7$.

Now consider the terms $\gamma_{41}[\gamma_{12}]; \gamma_7$ and $\alpha_{32}; \delta_2; \epsilon_2$. They have common source $t_3[t_2]$, which is strictly simpler than the source of $\alpha_3$, so we can apply the inductive hypothesis to get $\gamma_8$ and $\gamma_9$. We then have:

$$\begin{aligned}\gamma_2;\gamma_9 &= \alpha_1;\alpha_{31}\,t_2;\beta_{t_3,t_2};\alpha_{32};\delta_2;\epsilon_2;\gamma_9\\ &= \alpha_1;\alpha_{31}\,t_2;\lambda\gamma_{41}\,\gamma_{12};\beta_{t_4,v_2};\gamma_7;\gamma_8\\ &= \alpha_1;\alpha_{31}\,t_2;\lambda\gamma_{41}\,\gamma_{12};\epsilon_4\,v_2;\epsilon_3;\gamma_6;\gamma_8\\ &= \alpha_1;\gamma_{11}\,\gamma_{12};\epsilon_1;\gamma_5;\gamma_6;\gamma_8\\ &= \gamma_1;\gamma_5;\gamma_6;\gamma_8\end{aligned}$$

so take $\gamma_1' = \gamma_5;\gamma_6;\gamma_8$ and $\gamma_2' = \gamma_9$. □

## 5.3 Mellifluence

The $2\lambda$-calculus has another property, related to strong confluence, which has not been investigated in the $\lambda$-calculus. In section 6 we use this property to relate confluence, strong normalisation and normal forms. In this section we show that every rewrite is mellifluent, where:

**Definition.** A rewrite $\gamma : t \Rightarrow u$ is **mellifluent** if whenever $\delta_1, \delta_2 : u \Rightarrow v$ satisfy $\gamma;\delta_1 = \gamma;\delta_2$, there exists $\gamma' : v \Rightarrow w$ such that $\delta_1;\gamma' = \delta_2;\gamma'$.

**Lemma 22.**

1. Any rewrite in $\mathcal{A}$ is mellifluent.
2. If $\gamma_1$ and $\gamma_2$ are mellifluent, then $\gamma_1;\gamma_2$ is mellifluent.
3. If $\gamma_1;\gamma_2$ is mellifluent, then $\gamma_2$ is mellifluent.

*Proof.* Straightforward. □

**Lemma 23.** If $\gamma\colon t \Rightarrow u$ is mellifluent, then $\lambda\gamma\colon \lambda t \Rightarrow \lambda u$ is mellifluent.

*Proof.* Let $\gamma_1, \gamma_2\colon \lambda u \Rightarrow v$ satisfy $\lambda\gamma;\gamma_1 = \lambda\gamma;\gamma_2$. Then $\gamma_j = (\lambda u;\lambda\gamma_{j1};\epsilon_j)$ so $\lambda\gamma;\gamma_j = (\lambda t;\lambda(\gamma;\gamma_{j1});\epsilon_j)$ which is in canonical form, so $\epsilon_1 = \epsilon_2$ and $\gamma;\gamma_{11} = \gamma;\gamma_{21}$. But $\gamma$ is mellifluent, so there exists $\gamma_3$ satisfying $\gamma_{11};\gamma_3 = \gamma_{21};\gamma_3$. Now by lemma 19 there exist $\gamma_4$ and $\epsilon_4$ satisfying $\lambda\gamma_3;\epsilon_4 = \epsilon_j;\gamma_4$, and $\gamma_1;\gamma_4 = \gamma_2;\gamma_4$ so $\lambda\gamma$ is mellifluent. □

The next lemma describes a property of rewrites in $\mathcal{E}$:

**Lemma 24.** If $\epsilon\colon \lambda t \Rightarrow u$ in $\mathcal{E}$ then $u = \lambda u'$ and there exist $\gamma_1\colon t \Rightarrow v$ and $\gamma_2\colon u' \Rightarrow v$ such that $\epsilon;\lambda\gamma_2 = \lambda\gamma_1$.

*Proof.* By induction on the structure of $\epsilon$.

case 1: $\epsilon = \lambda t$. Then $u' = t$; take $\gamma_1 = \gamma_2 = t$.

case 2: $\epsilon = \epsilon_1; \eta_{u_1}; \lambda(u_1^1\, \epsilon_2; \epsilon_3)$

Apply the inductive hypothesis to $\epsilon_1$ to get $u_1 = \lambda u_2$, $\gamma_3 \colon t \Rightarrow v_1$ and $\gamma_4 \colon u_2 \Rightarrow v_1$. We now have $u_1^1\, \epsilon_2; \epsilon_3 \colon u_1^1\, 1 \Rightarrow u'$ and $\beta_{u_2^2,1}; \gamma_4 \colon u_1^1\, 1 \Rightarrow v_1$; apply confluence to get $\gamma_5$ and $\gamma_6$. We then have:

$$\begin{aligned}\epsilon; \lambda\gamma_6 &= \epsilon_1; \eta_{u_1}; \lambda(u_1^1\, \epsilon_2; \epsilon_3); \lambda\gamma_6\\ &= \epsilon_1; \eta_{u_1}; \lambda(\beta_{u_2^1,1}; \gamma_4); \lambda\gamma_5\\ &= \epsilon_1; \lambda\gamma_4; \lambda\gamma_5\\ &= \lambda(\gamma_3; \gamma_5)\end{aligned}$$

so take $\gamma_1 = \gamma_3; \gamma_5$ and $\gamma_2 = \gamma_6$. □

From this it follows that $\eta$-expansion is mellifluent:

**Lemma 25.** The rewrite $\eta_t \colon t \Rightarrow \lambda(t^1\, 1)$ is mellifluent.

*Proof.* Let $\gamma_1, \gamma_2 \colon \lambda(t^1\, 1) \Rightarrow v$ satisfy $\eta_t; \gamma_1 = \eta_t; \gamma_2$. Then

$$\gamma_j = (\lambda(t^1\, 1); \lambda\gamma_{j1}; \epsilon_j)$$

where $\epsilon_j \colon \lambda u_j \Rightarrow v$. By lemma 24, there exist $\gamma_{j2}, \gamma_{j3}$ such that $\epsilon_j; \lambda\gamma_{j3} = \lambda\gamma_{j2}$; apply confluence to $\gamma_{13}$ and $\gamma_{23}$ to get $\gamma_{14}$ and $\gamma_{24}$. We will take $\gamma_3 = \lambda(\gamma_{13}; \gamma_{14}) = \lambda(\gamma_{23}; \gamma_{24})$, and prove that $\gamma_1; \gamma_3 = \gamma_2; \gamma_3$.

If $\gamma_{j5} = \gamma_{j1}; \gamma_{j2}; \gamma_{j4}$, then $\eta_t; \lambda\gamma_{15} = \eta_t; \lambda\gamma_{25}$ and $\gamma_j; \gamma_3 = \lambda\gamma_{j5}$. Let $\gamma_{j5} = (\alpha_j; \delta_j; \epsilon_{j1})$, and proceed by cases of $\alpha_j$:

case 1: $\alpha_1 = \alpha_2 = t^1\, 1$. Then $\delta_j = \gamma_{j6}^1\, (1; 1; \epsilon_{j2})$, and

$$\eta_t; \lambda\gamma_{j5} = \gamma_{j6}; \eta_w; \lambda(w^1\, \epsilon_{j2}; \epsilon_{j1})$$

which is in canonical form. Therefore, $\gamma_{16} = \gamma_{26}$, $\epsilon_{12} = \epsilon_{22}$ and $\epsilon_{11} = \epsilon_{21}$, so $\gamma_{15} = \gamma_{25}$.

case 2: $\alpha_j = \alpha_{j1}^1\, 1; \beta_{w,1}; \alpha_{j2}$. Then $\eta_t; \lambda\gamma_{j5} = (\alpha_{j1}; \lambda(\alpha_{j2}; \delta_j; \epsilon_{j1}); \lambda z)$ which is in canonical form; matching up as before gives $\gamma_{15} = \gamma_{25}$.

case 3: One of each. This case is impossible, since the two canonical forms cannot match. □

Finally:

**Proposition 26.** All the rewrites of the $2\lambda$-calculus are mellifluent.

*Proof.* By induction on $||u||$, where $\gamma : t \Rightarrow u$.

Let $\gamma = (\alpha; \delta; \epsilon)$. By lemma 22, it is sufficient to prove that $\delta\colon t' \Rightarrow u'$ and $\epsilon\colon u' \Rightarrow u$ are mellifluent; note that $||u'|| \leq ||u||$. First consider $\epsilon$:

case 1: $\epsilon = t$. This is identity, therefore mellifluent.

case 2: $\epsilon = \epsilon_1; \eta_v; \lambda(v^1\,\epsilon_2; \epsilon_3)$. Then $v^1\,\epsilon_2; \epsilon_3\colon v^1\,1 \Rightarrow u_2$ where $\lambda u_2 = u$ so $||u_2|| < ||u||$ so by the inductive hypothesis, $v^1\,\epsilon_2; \epsilon_3\colon v^1\,1 \Rightarrow u_2$ is mellifluent, and by lemma 23, $\lambda(v^1\,\epsilon_2; \epsilon_3)$ is mellifluent. Also, $\epsilon_1\colon u' \Rightarrow v$, and $||v|| < ||\lambda(v^1\,1)|| \leq ||u||$ so by the inductive hypothesis, $\epsilon_1$ is mellifluent. Finally, $\eta_v$ is mellifluent by lemma 25, so $\epsilon$ is mellifluent.

Next we consider $\delta\colon t' \Rightarrow u'$:

case 1: $\delta = j$. This is identity, therefore mellifluent.

case 2: $\delta = \lambda\gamma_1$. This is mellifluent by the inductive hypothesis and lemma 23.

case 3: $\delta = \gamma_1\,\gamma_2$. Then $t' = t_1\,t_2$ and $u' = u_1\,u_2$. Let $\gamma_3, \gamma_4\colon u' \Rightarrow v$ satisfy $\delta; \gamma_3 = \delta; \gamma_4$ and proceed by cases of $\gamma_3$ and $\gamma_4$:

case 3.1: $\gamma_j = (u_1\,u_2; \gamma_{j1}\,\gamma_{j2}; \epsilon_j)$ for $j = 3, 4$. Then

$$\delta; \gamma_j = (t'; (\gamma_1; \gamma_{j1})\,(\gamma_2; \gamma_{j2}); \epsilon_j)$$

which is in canonical form, so $\epsilon_3 = \epsilon_4$ and $\gamma_k; \gamma_{3k} = \gamma_k; \gamma_{4k}$ for $k = 1, 2$. By the inductive hypothesis, there exist $\gamma_{5k}$ satisfying $\gamma_{3k}; \gamma_{5k} = \gamma_{4k}; \gamma_{5k}$, and by confluence there exist $\gamma_6, \epsilon_6$ satisfying $\gamma_{51}\,\gamma_{52}; \epsilon_6 = \epsilon_j; \gamma_6$. Then $\gamma_3; \gamma_6 = \gamma_4; \gamma_6$, so $\delta$ is mellifluent.

case 3.2: $\gamma_j = (\alpha_j\,u_2; \beta_{v_j,u_2}; \gamma_{j1})$ for $j = 3, 4$. Then $\alpha_3 = \alpha_4$ and $v_3 = v_4$ by lemma 18, so $\delta; \gamma_j = (\delta; \alpha_j\,u_2; \beta_{v_j,u_2}); \gamma_{j1}$. But $\delta; \alpha_j\,u_2; \beta_{v_j,u_2}\colon t_1 \Rightarrow v_j[u_2]$ and $||v_j[u_2]|| < ||\lambda v_j\,u_2|| \leq ||u'||$, so by the inductive hypothesis it is mellifluent, and there exists $\gamma_5$ satisfying $\gamma_{31}; \gamma_5 = \gamma_{41}; \gamma_5$. Then $\gamma_3; \gamma_5 = \gamma_4; \gamma_5$, so $\delta$ is mellifluent.

case 3.3: $\gamma_3 = (u'; \gamma_{31}\,\gamma_{32}; \epsilon_3)$ and $\gamma_4 = (\alpha_4\,u_2; \beta_{v_4,u_2}; \gamma_{41})$, or vice versa. This case cannot arise since $\delta; \gamma_4$ and $\delta; \gamma_3$ then have different canonical forms. □

## 5.4 Long Normal Forms

The $2\lambda$-calculus (thought of naïvely as a rewrite system) has unrestricted $\eta$-expansion, which is clearly non-terminating. Nonetheless, it does have certain terms, the 'long-$\beta\eta$-normal forms,' which play the rôle of normal forms in several treatments [1, 5, 6, 7, 11, 12, 13]. These long normal forms avoid infinite expansion paths by stopping $\eta$-expansion when the structure of the term matches its type.

In this section we investigate these long-$\beta\eta$-normal forms, and show that any rewrite whose source is of this form has a very special property. Not only is such a rewrite reversible (in the sense that there is a rewrite back the other way) but the composition of the rewrite with its reversal is equal to the identity in the theory 2-$\lambda$. Furthermore, we show that this property precisely characterises the long-$\beta\eta$-normal forms.

First we define the necessary concepts.

**Definition.**

- A rewrite $\gamma\colon t \Rightarrow u$ is **split monic** if there exists $\delta\colon u \Rightarrow t$ such that $\gamma;\delta = t$.
- A term $t$ is **essentially normal** if every rewrite $\gamma\colon t \Rightarrow u$ is split monic.
- Recall that **long-$\beta\eta$-normal forms** and **reduced** forms are defined inductively by
  - $t\colon X \to Y$ is in long-$\beta\eta$-normal form iff $t = \lambda t'$ where $t'\colon Y$ is in long-$\beta\eta$-normal form.
  - $t\colon B$ is in long-$\beta\eta$-normal form iff $t$ is in reduced form.
  - $j$ is in reduced form.
  - $t_1\, t_2$ is in reduced form iff $t_1$ is in reduced form and $t_2$ is in long-$\beta\eta$-normal form.
  - $\lambda t$ is not in reduced form.

We will prove that the rewrites from long-$\beta\eta$-normal forms are split monic. This involves studying the rewrites from reduced forms, which satisfy the following.

**Definition.** Let $\mathcal{P}$ be the smallest set of rewrites such that:

- All identity rewrites are in $\mathcal{P}$.
- If $\gamma \in \mathcal{P}$ then $\gamma;\epsilon \in \mathcal{P}$, for all $\epsilon$ in $\mathcal{E}$.

- If $\gamma_1; \gamma_2 \in \mathcal{P}$ then $\gamma_1 \in \mathcal{P}$.

Note that we have defined $\mathcal{P}$ for each type independently. In particular, for base types $B$, the second clause does not apply, and $\gamma \in \mathcal{P}$ iff $\gamma$ is split monic.

The following lemma describes a closure property of $\mathcal{P}$ which relates rewrites of different type:

**Lemma 27.** If $\gamma_1 \colon t_1 \Rightarrow u_1 \colon X \to Y \in \mathcal{P}$ and $t_2 \colon X$ is essentially normal, then for any $\gamma_2 \colon t_2 \Rightarrow u_2$, the application $\gamma_1\,\gamma_2$ is in $\mathcal{P}$.

*Proof.* We prove that this property is preserved by the three clauses defining $\mathcal{P}$.

- If $\gamma_1$ is identity then $\gamma_1\,\gamma_2$ is split monic, so a member of $\mathcal{P}$.
- If $\gamma_1$ has this property then we prove that $\gamma_1; \epsilon$ does by structural induction on $\epsilon$.

  case 1: $\epsilon = u_1$. Then $\gamma_1; \epsilon = \gamma_1$.

  case 2: $\epsilon = \epsilon_1; \eta_u; \lambda(u^1\,\epsilon_2; \epsilon_3)$. Then

  $$(\gamma_1; \epsilon)\,\gamma_2; \beta_{v,u_2} = (t_1; (\gamma_1; \epsilon_1)\,(\gamma_2; \epsilon_2[v]); \epsilon_3[v])$$

  which, by inductive hypothesis, is a member of $\mathcal{P}$. Therefore $\gamma_1; \epsilon \in \mathcal{P}$ as required.
- If $\gamma_1; \gamma_3 \in \mathcal{P}$ has this property, then $(\gamma_1\,\gamma_2); (\gamma_3\,u_2) = (\gamma_1; \gamma_3)\,\gamma_2$ is a member of $\mathcal{P}$, so $\gamma_1\,\gamma_2 \in \mathcal{P}$ as required. ☐

We are now ready to prove half our theorem:

**Proposition 28.**

- If $\gamma \colon t \Rightarrow u$ and $t$ is in long-$\beta\eta$-normal form, then $\gamma$ is split monic.
- If $\gamma \colon t \Rightarrow u$ and $t$ is in reduced form, then $\gamma \in \mathcal{P}$.

*Proof.* By structural induction on $t$. We proceed by cases:

case 1: $t \colon X \to Y$ is in long-$\beta\eta$-normal form. Then $t = \lambda t_1$ where $t_1$ is in long-$\beta\eta$-normal form, and $\gamma = (t; \lambda\gamma_1; \epsilon)$. By lemma 24, there exist $\gamma_2 \colon u_1 \Rightarrow v$ and $\gamma_3 \colon u_2 \Rightarrow v$ s.t. $\epsilon; \lambda\gamma_3 = \lambda\gamma_2$. Then $\gamma_1; \gamma_2 \colon t_1 \Rightarrow v$ and by inductive hypothesis, has a left inverse $\gamma_4$. Now $\lambda(\gamma_3; \gamma_4)$ is a left inverse for $\gamma$.

case 2: $t\colon B$ is in long-$\beta\eta$-normal form. Then $t$ is in reduced form, and by the second inductive hypothesis, $\gamma \in \mathcal{P}$. As remarked above, this means $\gamma$ is split monic.

case 3: $t = j$ is in reduced form. Then $\gamma = (j; j; \epsilon)$ which is certainly in $\mathcal{P}$.

case 4: $t = t_1\, t_2$ is in reduced form. Then $\gamma = (t; \gamma_1\, \gamma_2; \epsilon)$ by a simple induction. By inductive hypothesis $\gamma_1 \in \mathcal{P}$, and $t_2$ is essentially normal, so $\gamma_1\, \gamma_2 \in \mathcal{P}$ by lemma 27. Therefore $\gamma \in \mathcal{P}$ as required. □

The next lemma tells us more about the rewrites in $\mathcal{P}$:

**Lemma 29.** Every $\gamma \in \mathcal{P}$ is of the form $(t; \delta; \epsilon)$ where $\delta$ satisfies one of the following:

- $\delta = j$
- $\delta = \lambda\gamma_1$ and $\gamma_1 \in \mathcal{P}$
- $\delta = \gamma_1\, \gamma_2$ where $\gamma_1$ is in $\mathcal{P}$ and $\gamma_2$ is split monic.

*Proof.* We prove that this property is preserved by the three clauses defining $\mathcal{P}$. It is clear that all identities are of this form, and that it is preserved by composition with rewrites in $\mathcal{E}$. It remains to prove that if $\gamma_1; \gamma_2 \in \mathcal{P}$ is of one of the three forms above, then so is $\gamma_1$. The proof is by induction on $|\gamma_2|_{\mathcal{G}}$.

Let $\gamma_j = (\alpha_j; \delta_j; \epsilon_j)$; it is clear from the definition of † that $\alpha_1 = t$. We proceed by cases of $\epsilon_1$ and $\alpha_2$:

case 1: $\epsilon_1 = \alpha_2 = u$. There are three subcases, depending on the form of $\delta_j$:

case 1.1: $\delta_j = i$. Then $\gamma_1$ is of the required form.

case 1.2: $\delta_j = \lambda\gamma_{j1}$. Then $\gamma_1; \gamma_2 = (t; \lambda(\gamma_{11}; \gamma_{21}); \epsilon_2)$ and $\gamma_{11}; \gamma_{21} \in \mathcal{P}$, so $\gamma_{11} \in \mathcal{P}$ and $\gamma_1$ is of the required form.

case 1.3: $\delta_j = \gamma_{j1}\, \gamma_{j2}$. Then $\gamma_1; \gamma_2 = (t; (\gamma_{11}; \gamma_{21})\, (\gamma_{12}; \gamma_{22}); \epsilon_2)$ and $\gamma_{11}; \gamma_{21} \in \mathcal{P}$, $\gamma_{12}; \gamma_{22}$ is split monic. Then $\gamma_{11} \in \mathcal{P}$ and $\gamma_{12}$ is split monic, so $\gamma_1$ is of the required form.

case 2: $\alpha_2 = u$, $\epsilon_1 \neq u$. Then $\gamma_1; \gamma_2 = (t; \delta_1; v); (\epsilon_1; \gamma_2)$ and by inductive hypothesis, $(t; \delta_1; v)$ is of the required form. Therefore $\gamma_1$ is also.

case 3: $\epsilon_1 = u$, $\alpha_2 = \alpha_{21}\, t_2; \beta_{t_1,t_2}; \alpha_{22}$. Then $\delta_1 = \gamma_{11}\, \gamma_{12}$, and we proceed by cases of $\gamma_{11} \dagger \alpha_{21}$:

case 3.1: $\gamma_{11};\alpha_{21} = (\alpha_3;\lambda\gamma_3;\lambda t_1)$. Then $\gamma_1;\gamma_2 = (\alpha_3\, v;\beta_{w,v};\dots)$, contradicting the hypothesis that it is of the given form.

case 3.2: $\gamma_{11};\alpha_{21} = (\gamma_3;\eta_w;\lambda(w^1\,\epsilon_{31};\epsilon_{32}))$. Then

$$\gamma_1;\gamma_2 = t;\gamma_3\,(\gamma_{12};\epsilon_{31}[u]);\epsilon_{32}[u]);(\alpha_{22};\delta_2;\epsilon_2)$$

By inductive hypothesis, $\gamma_3 \in \mathcal{P}$ and $\gamma_{12};\epsilon_{31}[u]$ is split monic, so $\gamma_{11} \in P$ and $\gamma_{12}$ is split monic, as required. □

We can now prove the other half of the theorem:

**Proposition 30.** Let $t$ be a term. Then

- If every $\gamma\colon t \Rightarrow u$ is split monic, then $t$ is in long-$\beta\eta$-normal form.
- If every $\gamma\colon t \Rightarrow u$ is in $\mathcal{P}$, and $t$ is not of the form $\lambda t_1$, then $t$ is in reduced form.

*Proof.* By structural induction on $t$. We proceed by cases:

case 1: $t\colon X \to Y$ and every $\gamma\colon t \Rightarrow u$ is split monic. Then in particular, $\eta_t\colon t \Rightarrow \lambda(t^1\,1)$ is split monic, and its inverse $\gamma$ has the form $(\lambda(t^1\,1);\lambda\gamma';\epsilon)$. Then $t = \lambda t_1$. Let $\gamma_1 : t_1 \Rightarrow u_1$. Then $\lambda\gamma_1 : t \Rightarrow \lambda u_1$ is split monic, with inverse $(\lambda u_1;\lambda\gamma_2;\epsilon_2)$, say. Then $\lambda\gamma_1;(\lambda u_1;\lambda\gamma_2;\epsilon_2) = (\lambda t_1;\lambda(\gamma_1;\gamma_2);\epsilon_2)$ and $\gamma_1$ is split monic. By inductive hypothesis, therefore, $t_1$ is in long-$\beta\eta$-normal form, and so is $t$.

case 2: $t\colon B$ and every $\gamma\colon t \Rightarrow u$ is split monic. Then every such $\gamma$ is in $\mathcal{P}$, and since $t$ cannot be a lambda term, $t$ is in reduced form by the second inductive hypothesis. Therefore $t$ is in long-$\beta\eta$-normal form.

case 3: $t = j$. Then $t$ is in reduced form.

case 4: $t = t_1\,t_2$ and every $\gamma\colon t \Rightarrow u$ is in $\mathcal{P}$. If $t_1 = \lambda t_{11}$, then $\beta_{t_{11},t_2}\colon t \Rightarrow t_{11}[t_2]$, contradicting lemma 29.

Let $\gamma_1\colon t_1 \Rightarrow u_1$. Then $\gamma_1\,t_2\colon t \Rightarrow u_1\,t_2$ is in $\mathcal{P}$, and by lemma 29 $\gamma_1$ is in $\mathcal{P}$. By inductive hypothesis, therefore, $t_1$ is in reduced form.

Let $\gamma_2\colon t_2 \Rightarrow u_2$. Then $t_1\,\gamma_2\colon t \Rightarrow t_1\,u_2$ is in $\mathcal{P}$, and by lemma 29 $\gamma_2$ is split monic. By the first inductive hypothesis, therefore, $t_2$ is in long-$\beta\eta$-normal form, so $t$ is in reduced form. □

Putting this together with proposition 28, we have proved

**Corollary 31.** A term is in long-$\beta\eta$-normal form iff it is essentially normal.

We need one more property of split monic arrows, related to mellifluence:

**Lemma 32.** If $\gamma\colon t \Rightarrow u$ and $\zeta_1, \zeta_2\colon u \Rightarrow v$ satisfy $\gamma; \zeta_1 = \gamma; \zeta_2$ where $\zeta_1, \zeta_2$ are split monic, then $\zeta_1 = \zeta_2$.

*Proof.* It is convenient to define a set $\mathcal{Q}$ of rewrites whose canonical forms are built up entirely from rewrites in $\mathcal{E}$:

- If $\epsilon : j \Rightarrow t$ is in $\mathcal{E}$ then $j; j; \epsilon \in \mathcal{Q}$
- If $\epsilon : \lambda t \Rightarrow u$ is in $\mathcal{E}$ and $\zeta : s \Rightarrow t$ is in $\mathcal{Q}$, then $\lambda s; \lambda\zeta; \epsilon$ is in $\mathcal{Q}$.
- If $\epsilon : t_1\, t_2 \Rightarrow u$ is in $\mathcal{E}$ and $\zeta_1 : s_1 \Rightarrow t_1$ and $\zeta_2 : s_2 \Rightarrow t_2$ are in $\mathcal{Q}$, then $s_1\, s_2; \zeta_1\, \zeta_2; \epsilon$ is in $\mathcal{Q}$.

By lemma 29, $\mathcal{P} \subseteq \mathcal{Q}$, so every split monic rewrite is in $\mathcal{Q}$. A straightforward induction shows that the composition of two rewrites in $\mathcal{Q}$ is in $\mathcal{Q}$. We prove the stronger condition that the lemma is true for all $\zeta_1, \zeta_2$ in $\mathcal{Q}$.

The proof is by induction on $|\zeta_j|_{\mathcal{G}}$. Let $(\alpha; \delta; \epsilon)$ be the canonical form of $\gamma$, and $(t_j; \theta_j; \epsilon_j)$ that of $\zeta_j$. There are three cases of $\theta_1$ and $\theta_2$:

case 1: $\theta_1 = \theta_2 = i$. Then $\epsilon = i$ and $\delta = i$ so $\gamma; \zeta_j = (\alpha; i; \epsilon_j)$. Matching canonical forms gives $\epsilon_1 = \epsilon_2$, so $\zeta_1 = \zeta_2$ as required.

case 2: $\theta_j = \lambda\zeta_{j1}$. Proceed by cases of $\epsilon$:

case 2.1: $\epsilon = u$. Then $\delta = \lambda\gamma_1$ and $\gamma; \zeta_j = (\alpha; \lambda(\gamma_1; \zeta_{j1}); \epsilon_j)$, so $\epsilon_1 = \epsilon_2$ and $\gamma_1; \zeta_{11} = \gamma_1; \zeta_{21}$. By inductive hypothesis, $\zeta_{11} = \zeta_{21}$ so $\zeta_1 = \zeta_2$ as required.

case 2.2: $\epsilon = \epsilon_3; \eta_x; \lambda(x^1\, \epsilon_4; \epsilon_5)$. Then $(x^1\, 1; x^1\, \epsilon_4; \epsilon_5); \zeta_{j1}$ is in $\mathcal{Q}$, so equals $(x^1\, 1; \zeta_{j2}^1\, \epsilon_{j1}; \epsilon_{j2})$ for some $\zeta_{j2}$ in $\mathcal{Q}$. Therefore

$$\gamma; \zeta_j = (\alpha; \delta; \epsilon_3); (\zeta_{j2}; \eta_y; \lambda(y^1\, \epsilon_{j1}; \epsilon_{j2}); \epsilon_j)$$

and by inductive hypothesis, $\zeta_{12} = \zeta_{22}$, $\epsilon_{11} = \epsilon_{21}$, $\epsilon_{12} = \epsilon_{22}$ and $\epsilon_1 = \epsilon_2$. Therefore $(x^1\, 1; x^1\, \epsilon_4; \epsilon_5); \zeta_{11} = (x^1\, 1; x^1\, \epsilon_4; \epsilon_5); \zeta_{21}$, and by the inductive hypothesis $\zeta_{11} = \zeta_{21}$. So $\zeta_1 = \zeta_2$ as required.

case 3: $\theta_j = \zeta_{j1}\, \zeta_{j2}$. Then $\epsilon = u_1\, u_2$ and $\delta = \gamma_1\, \gamma_2$ so

$$\gamma; \zeta_j = (\alpha; (\gamma_1; \zeta_{j1})\, (\gamma_2; \zeta_{j2}); \epsilon_j)$$

Therefore $\gamma_1; \zeta_{11} = \gamma_1; \zeta_{21}$, $\gamma_2; \zeta_{12} = \gamma_2; \zeta_{22}$ and $\epsilon_1 = \epsilon_2$. By inductive hypothesis, $\zeta_{11} = \zeta_{21}$ and $\zeta_{12} = \zeta_{22}$, so $\zeta_1 = \zeta_2$ as required. $\square$

# 6 General Results

In this section we take the results we have proved about the 2$\lambda$-calculus and generalise: we aim for a theory which can be applied to many different rewrite systems. Of course it is dangerous to generalise from one example, and the definitions of this section must be taken as tentative. Nonetheless, the results do clarify the relationships between the properties proved in sections 4 and 5.

We assume a rewrite system to consist of *elements* with *rewrites* acting between them. These rewrites are multi-step: in particular, there is a zero-step rewrite on each element, and rewrites to and from any element can be composed. This immediately leads us to the idea that a rewrite system forms a category, and we take this as the basic definition.

An important example is given by any set of elements together with a set of one-step rewrites between them, i.e. a graph. The categorical rewrite system is then given by the path category of the graph: the objects are the elements, and the arrows are the sequences of one-step rewrites. This example provides important intuition, even when the category is far from this form. In particular, we always interpret the identity on an element as a zero-step operation, which is never actually performed, so takes no time. Similarly, we always interpret composition in the category as concatenation of rewrite sequences, even when the result is shorter than the sum of the parts.

The example which forms the subject of this paper we will call 'the category 2-$\lambda$'. Its objects are the terms (in context) of the 2$\lambda$-calculus, and its arrows are equivalence classes of rewrites, under the equivalence defined by the theory 2-$\lambda$. Identities are as expected, and composition is ';'.

This leads to a subtle form of conditional rewriting, the full implications of which have not been explored. For example, the $\eta$-expansion

$$\eta_t\, u \colon t\, u \to \lambda(t^1\, 1)\, u$$

can be followed by the $\beta$-reduction

$$\beta_{(t^1\, 1),u} \colon \lambda(t^1\, 1)\, u \to t\, u$$

and the composition is the zero-step rewrite. This reduction path *cannot* therefore be followed by any legitimate strategy, as the resulting composition is supposed to take no time to execute. We must discard the idea of an abstract machine which 'picks a rewrite at random' to execute, and continues until there are none left.

A full understanding of these points requires a formal definition of a 'legitimate strategy' for an abstract machine. There is no space to develop this here; indeed the author must confess to having no completely satisfactory definition. Nonetheless, this approach is related to Jay and Ghani's idea of 'cutting loops' [13], and is more algebraic than Di Cosmo and Kesner's 'simulated expansions' [7]. We hope the reader finds it stimulating.

In the following development, one property recurs in almost every proof: the *mellifluence* of section 5.3. Since it is needed so often, and its significance to rewriting is unclear, we assume it as an axiom of rewrite systems.

**Definition.** A category is **mellifluent** if the following holds for all arrows $f$, $g$ and $h$:

- If $f\colon x \to y$ and $g, h\colon y \to z$ are such that $f; g = f; h$ then there exists $k\colon z \to w$ such that $g; k = h; k$.

We proved the mellifluence of the category 2-$\lambda$ in proposition 26. The path category on any graph is clearly mellifluent, since the hypothesis of the axiom only holds when $g = h$.

An intuitive understanding of mellifluence is not easy. It is something like "If the difference between $g$ and $h$ is invisible from $x$, then it doesn't matter in the long run." The author discovered this property when trying to prove the results of this section.

If the only rewrite from an element is the identity, that element is clearly in normal form. Furthermore, if every rewrite from an element is a prefix of the identity, the informal arguments about legitimate strategies imply that this, too, is a normal form. We take this as the definition.

**Definition.**

- An object $x$ of a rewriting category is **normal** if every arrow $f\colon x \to y$ is split monic, i.e. there exists $g\colon y \to x$ such that $f; g = 1_x$.
- An object $y$ is **weakly normalising** if there exists $f\colon y \to x$ for some normal $x$. In this case we call $x$ a **normal form** of $y$.
- A rewriting category is **weakly normalising** if every object is weakly normalising.

Corollary 31 states that the normal forms of the category 2-$\lambda$ are precisely the long-$\beta\eta$-normal forms. Since every term of the simply typed $\lambda$-calculus has a long-$\beta\eta$-normal form, the category 2-$\lambda$ is weakly normalising.

In the path category of a graph, an object is normal iff there are no edges from that object in the original graph. An object is weakly normalising iff there is a path to a normal object.

**Lemma 33.**

1. Any arrow between normal objects is an isomorphism.
2. If $f\colon x \to y$ then any normal form of $y$ is a normal form of $x$.
3. Let $x$ be a normal object in a mellifluent category, and $f\colon x \to y$. Then the map $g\colon y \to x$ satisfying $f;g = 1_x$ is unique.

*Proof.* (1) Let $x$ and $y$ be normal objects, and $f\colon x \to y$. Then because $x$ is normal, there exists $g\colon y \to x$ such that $f;g = 1_x$. Similarly, because $y$ is normal, there exists $h\colon x \to y$ such that $g;h = 1_y$. Now, $f = f;(g;h) = (f;g);h = h$, so it is iso.

(2) If $g\colon y \to z$ with $z$ normal then $f;g\colon x \to z$.

(3) Let $g_1, g_2\colon y \to x$ both satisfy $f;g_j = 1_x$. Then by mellifluence, there exists $h\colon x \to z$ such that $g_1;h = g_2;h$. But $h$ must be monic because $x$ is normal, so $g_1 = g_2$. □

Perhaps the most obvious definition of confluence is the diamond property: that any span has a cospan. However, this definition completely ignores the equalities between rewrites, and relies instead on equality between objects. We reject this definition as 'uncategorical' and regard diamonds which do not commute as 'fortuitous'. The commuting diamond property has a much better theory:

**Definition.**

- An object $x$ of a rewriting category is **confluent** if for all pairs $f_1\colon x \to y_1$ and $f_2\colon x \to y_2$ there exist $z$, $g_1\colon y_1 \to z$ and $g_2\colon y_2 \to z$ such that $f_1;g_1 = f_2;g_2$.
- A rewriting category is **confluent** if every object is confluent.

Proposition 21 states that the category 2-$\lambda$ is confluent. The path category of a graph is confluent iff there is at most one edge from each node in the graph. This means that in many cases the path category is not the right category to study: equations between paths must be imposed which render the completions of critical pairs commuting.

Note that confluence and mellifluence are precisely the conditions for a calculus of fractions [8]. This means that we can calculate the free groupoid on a confluent rewriting category in a particularly simple way. This groupoid can be interpreted as the equational theory generated by the rewrite system.

**Lemma 34.** Let $x$ be an object in a mellifluent category. Then

1. If $x$ is confluent and $f\colon x \to y$ then $y$ is confluent

2. If $x$ is normal then $x$ is confluent

3. If $x$ is confluent and $f\colon x \to y$ then any normal form of $x$ is a normal form of $y$

4. If $x$ is confluent then all its normal forms are isomorphic.

*Proof.* (1) Let $g_1\colon y \to z_1$ and $g_2\colon y \to z_2$. Then $f; g_1\colon x \to z_1$ and $f; g_2\colon x \to z_2$ so by confluence of $x$ there exist $h_1\colon z_1 \to w$ and $h_2\colon z_2 \to w$ such that $f; g_1; h_1 = f; g_2; h_2$. Now by mellifluence there exists $k\colon w \to v$ such that $g_1; h_1; k = g_2; h_2; k$ so two arrows which complete the commuting diamond are $h_1; k$ and $h_2; k$.

(2) Let $f_1\colon x \to y_1$ and $f_2\colon x \to y_2$. Since $x$ is normal there exist $g_1\colon y_1 \to x$ and $g_2\colon y_2 \to x$ such that $f_1; g_1 = 1_x = f_2; g_2$. But this shows that $x$ is confluent.

(3) Let $g\colon x \to z$ where $z$ is normal. Since $x$ is confluent there exist $h_1\colon y \to w$ and $h_2\colon z \to w$ such that $f; h_1 = g; h_2$. But $z$ is normal, so there exists $k\colon w \to z$ such that $h_2; k = 1_z$. Now $h_1; k\colon y \to z$ (and $f; h_1; k = g$).

(4) By part (3), if $x$ has two normal forms, then there is an arrow between them. But by lemma 33, this arrow is iso. □

Strong normalisation is the property that every rewrite sequence from an element is finite. This is clearly false in any category, as there are always infinite sequences of identities. Nonetheless, we can capture the idea that an $\omega$-sequence is a sequence of prefixes of a fixed (finite) rewrite, by saying that there is a cocone over the corresponding $\omega$-chain.

In order to develop a good theory, we strengthen this idea in two ways. Firstly we generalise $\omega$-chains to filtered diagrams; secondly we demand that the cocone is *separating*. The first allows us to find a cocone not just over a particular $\omega$-chain, but over a class of equivalent chains. The second is a technical condition, but can be thought of as choosing a cocone at whose vertex there are no sudden ambiguities.

**Definition.**

- Let $D$ be a diagram in a category. We call a cocone $\mu\colon D \to x$ over $D$ **separating** if for any other cocone $\nu\colon D \to y$ there is *at most one* arrow $f\colon x \to y$ such that $\mu; f = \nu$.

- An object $x$ of a rewriting category is **strongly normalising** if every filtered diagram containing $x$ has a separating cocone.

- A rewriting category is **strongly normalising** if every object is strongly normalising.

Note that if we replace 'at most one' with 'exactly one' in the definition of separating cocone, it becomes the definition of colimiting cocone. The category 2-$\lambda$, however, does not have filtered colimits.

In the path category of a graph, an object is strongly normalising iff there is no infinite sequence of edges from the node. The following lemma, together with lemma 32 shows that the category 2-$\lambda$ is strongly normalising:

**Lemma 35.** Let $x$ be an object of a mellifluent category.

1. If $x$ is confluent and weakly normalising, then any filtered diagram containing it has a cocone whose apex is normal.
2. If whenever $f\colon x \to y$, $g_1, g_2 : y \to z$ are such that $f; g_1 = f; g_2$ then $g_1 = g_2$, then any cocone over a diagram containing $x$ with vertex $y$ is separating.

*Proof.* (1) Let $D$ be a filtered diagram containing $x$, and $e\colon x \to v$ for $v$ normal. Define $\mu\colon D \to v$ as follows:

- For each object $y \in D$ there exist $z_y$, $f_y\colon y \to z_y$ and $g_y\colon x \to z_y$ in $D$ (since $D$ is filtered). By lemma 34, there exists $h_y\colon z_y \to v$ s.t. $g_y; h_y = e$. Then
$$\mu_y = f_y; h_y\colon y \to v$$
- For each arrow $k\colon y \to y'$ in $D$ there exist $w$, $l\colon z_y \to w$, $l'\colon z_{y'} \to w$ s.t. $g_y; l = g_{y'}; l'$ and $f_y; l = k; f_{y'}; l'$, since $D$ is filtered. Then there exists $m\colon w \to v$ s.t. $g_y; l; m = e$, and by mellifluence, $h_y = l; m$ and $h_{y'} = l'; m$. Now
$$k; \mu_{y'} = k; f_{y'}; h_{y'} = k; f_{y'}; l'; m = f_y; l; m = f_y; h_y = \mu_y$$
so $\mu$ is a cocone.

(2) Straightforward. □

**Lemma 36.** Let $x$ be an object in a mellifluent category. Then

1. If $x$ is strongly normalising and $f\colon x \to y$ then $y$ is strongly normalising
2. If $x$ is normal then $x$ is strongly normalising

*Proof.* (1) Let $D$ be a filtered diagram containing $y$. Then there is a diagram $D'$ formed by adjoining one new object $x$ and one new arrow $f: x \to y$ to $D$. $D'$ is filtered and contains $x$, so has a separating cocone, but a separating cocone over $D'$ restricts to one over $D$.

(2) Immediate from lemma 35 □

Finally, we show that strong normalisation implies weak normalisation. This result always depends on the axiom of choice, to chose a path to a normal form. Here we use the equivalent Zorn's Lemma: that if every chain in a poset is bounded, the poset has a maximal element.

**Proposition 37.** If $x$ is a strongly normalising object in a mellifluent category $C$, then it is weakly normalising, i.e. it has a normal form.

*Proof.* Let $\preccurlyeq$ be the partial order on arrows $f: x \to y$ induced by $(x \downarrow C)$: so $[f] \preccurlyeq [f']$ iff there exists $g: y \to y'$ such that $f; g = f'$. We will prove that every chain in this poset has an upper bound.

Let $[f_j] \preccurlyeq [f_{j+1}]$ be such a chain, and choose $g_j: f_j \to f_{j+1}$ in $(x \downarrow C)$. The resulting diagram in $C$ is linear, so filtered, so has a separating cocone. The image of this cocone in the partial order is an upper bound for the chain.

So every chain is bounded and we can apply Zorn's lemma to find a maximal element $[h]$, where $h: x \to z$. Now consider the full subcategory of $(x \downarrow C)$ of arrows in the equivalence class $[h]$. This category is filtered because of mellifluence and maximality, so its image in $C$ has a separating cocone $\mu: [h] \to v$. We will show that $v$ is normal.

Let $f = h; \mu_h: x \Rightarrow v$. Now if $g: v \Rightarrow u$ then by maximality $f; g \in [h]$ so there exists $g': u \Rightarrow v$ st. $f; g; g' = f$, and by separation, $g; g' = 1$. □

The combination of lemma 34 and proposition 37 means that if $x$ is confluent and strongly normalising then it has a normal form, unique up to isomorphism. However, the proof is unnecessarily complicated and non-constructive, using the axiom of choice. The next result gives a simple construction of the normal form in the confluent case.

**Lemma 38.** Let $x$ be an object in a mellifluent category $C$, and let $P: (x \downarrow C) \to C$ be the usual projection functor. Then

1. $x$ is confluent iff $(x \downarrow C)$ is filtered.

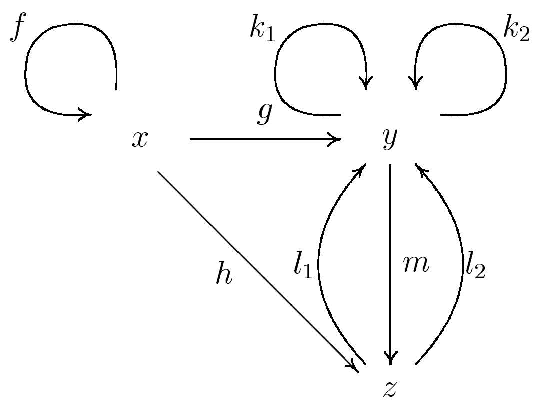
2. If $\mu: P \to y$ is separating then $y$ is normal.

*Proof.* (1) The two conditions for filteredness of the slice category are precisely mellifluence and confluence of $x$.

(2) The map $\mu_{1_x}: x \to y$ is an object of $(x \downarrow C)$, so $\mu_{\mu_1}: y \to y$, and by separation $\mu_{\mu_1} = 1_y$. If $f: y \to z$, then $\mu_1; f: x \to z$ is an object of $(x \downarrow C)$, so $\mu_{\mu_1;f}: z \to y$. Then $f; \mu_{\mu_1;f} = \mu_{\mu_1} = 1$ □

We have proved all the expected relationships between confluence, weak and strong normalisation, and even found a simple condition (lemma 35) for confluence + weak normalisation to imply strong normalisation. We now give an example to show that some such condition is necessary.

Let $C$ be the category with three objects $x$ $y$ and $z$, and eight non-identity arrows:




with composition defined by

$$
\begin{array}{ccc}
f; f = f & f; g = g & f; h = h \\
g; k_1 = g & g; k_2 = g & g; m = h \\
h; l_1 = g & h; l_2 = g & \\
k_1; k_1 = k_1 & k_1; k_2 = k_2 & k_1; m = m \\
k_2; k_1 = k_1 & k_2; k_2 = k_2 & k_2; m = m \\
l_1; k_1 = l_1 & l_1; k_2 = l_2 & l_1; m = 1_z \\
l_2; k_1 = l_1 & l_2; k_2 = l_2 & l_2; m = 1_z \\
m; l_1 = k_1 & m; l_2 = k_2 &
\end{array}
$$

Then $C$ is a mellifluent, confluent category and $z$ is normal, but $x$ is not strongly normalising because none of the three cocones over

$$x \xrightarrow{f} x \xrightarrow{f} x \xrightarrow{f} \cdots$$

is separating.

# 7 Conclusions

This treatment of the $\lambda$-calculus shows that rewriting can have a well-behaved proof theory. The analogy of terms as propositions and rewrites as proofs has lead to an interesting equational theory on rewrites with a lot of the feel of proof theory. Sequential composition acts like cut, and the triangle laws make $\eta$-expansion a sort of right rule, with $\beta$-reduction the corresponding left rule. This theory has a "cut-elimination" theorem which is associative and deterministic, and a categorical semantics which characterises $\lambda$-abstraction by an adjointness property. The author finds it hard to conceive of a neater state of affairs.

The application of these ideas to more general rewrite theory is perhaps less immediately convincing. In order to produce a good theory, we have defined normal forms which can be rewritten, confluence which puts a strong condition on the rewrites, strong normalisation which allows infinite reduction paths, and the condition "mellifluence" which has no obvious interpretation in terms of rewriting. Nonetheless, the author feels that these definitions have some justification if we understand the equations on rewrites as conditions on legitimate strategies. Only further work will decide this point.

The potential applications of this new theory are many. The $2\lambda$-calculus would generalise straightforwardly to more complex type theories such as 'system F' [9] and the 'calculus of constructions' [3]; indeed, since the proofs in this paper do not really depend on the types, we can expect the same results to hold. Many other types have "$\eta$-expansion" rules: unit types, surjective pairing, strong sums, recursive datatypes and so on. The problems here are not very different from $\eta$-expansion in the $\lambda$-calculus, and this approach is clearly worth trying.

In fact the generality of the definitions invites their application to much more varied examples. Since normal forms are defined 'up to isomorphism' we can normalise a commutative binary operation $*$ by making $x*y$ isomorphic to $y*x$. The definition of "strong normalisation" allows the possibility of infinite normal forms, which can be used to study streams and lazy datatypes. There are many other examples where some restriction on the rewrite strategy is essential. This is a fertile field for further work.

Finally, the author would like to thank the referees, whose comments lead to enormous improvements to this paper. If the result is at all readable, it is thanks to them.

# A The Proof of Proposition 12

> If $\Gamma \vdash \gamma_1 \colon t_1 \Rightarrow t_2 \colon X, \Gamma \vdash \gamma_2 \colon t_2 \Rightarrow t_3 \colon X$ and $\Gamma \vdash \gamma_3 \colon t_3 \Rightarrow t_4 \colon X$ are in $\mathcal{G}$, then $(\gamma_1 \dagger \gamma_2) \dagger \gamma_3 = \gamma_1 \dagger (\gamma_2 \dagger \gamma_3)$.

The proof is by induction on $|\gamma_3|_{\mathcal{G}}$. There are a total of eight well-formed cases of $\gamma_1, \gamma_2, \gamma_3$, with up to four subcases each. Fortunately, six of the main cases are straightforward, and can be left to the reader. The two remaining cases are as follows:

case 1: $\gamma_1 = \alpha_1; \gamma_{11}\, \gamma_{12}; t_{21}\, t_{22}$,
$\gamma_2 = t_{21}\, t_{22}; \gamma_{21}\, \gamma_{22}; t_{31}\, t_{32}$
and $\gamma_3 = \alpha_3\, t_{32}; \beta_{t_{33},t_{32}}; \gamma_{31}$.

There are several subcases, corresponding to the different cases in the definition of $\dagger$:

case 1.1: $\gamma_{21} \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} = \alpha_4; \lambda\gamma_4; \lambda t_{33}$
and $\gamma_{11} \dagger \alpha_4; \lambda\mathcal{I}(t_{23}); \lambda t_{23} = \alpha_5; \lambda\gamma_5; \lambda t_{23}$.

Then

$$\begin{aligned}(\gamma_{11} \dagger \gamma_{21}) \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} &= \gamma_{11} \dagger \alpha_4; \lambda\gamma_4; \lambda t_{33} \\ &= \alpha_5; \lambda\gamma_5; \lambda t_{23} \dagger \lambda t_{23}; \lambda\gamma_4; \lambda t_{33} \\ &= \alpha_5; \lambda(\gamma_5 \dagger \gamma_4); \lambda t_{33}\end{aligned}$$

so

$$\begin{aligned}(\gamma_1 \dagger \gamma_2) \dagger \gamma_3 &= \alpha_1; ; \alpha_5\, t_{12}; \beta_{t_{13},t_{12}}; (\gamma_5 \dagger \gamma_4)[\gamma_{12} \dagger \gamma_{22}] \dagger \gamma_{31} \\ &= (\alpha_1; ; \alpha_5\, t_{12}; \beta_{t_{13},t_{12}}; \gamma_5[\gamma_{12}] \dagger \gamma_4[\gamma_{22}]) \dagger \gamma_{31} \\ &= \gamma_1 \dagger (\gamma_2 \dagger \gamma_3)\end{aligned}$$

case 1.2: $\gamma_{21} \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} = \alpha_4; \lambda\gamma_4; \lambda t_{33}$,
$\gamma_{11} \dagger \alpha_4; \lambda\mathcal{I}(t_{23}); \lambda t_{23} = \gamma_5; \eta_{t_5}; \lambda(t_5^1\, \epsilon_{51}; \epsilon_{52})$
and $t_5^1\, 1; \mathcal{I}(t_5^1)\, (1; 1; \epsilon_{51}); \epsilon_{52} \dagger \gamma_4 = t_5^1\, 1; \gamma_6^1\, (1; 1; \epsilon_{61}); \epsilon_{62}$.

Then

$$\begin{aligned}(\gamma_{11} \dagger \gamma_{21}) \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} &= \gamma_{11} \dagger \alpha_4; \lambda\gamma_4; \lambda t_{33} \\ &= \gamma_5; \eta_{t_5}; \lambda(t_5^1\, \epsilon_{51}; \epsilon_{52}) \dagger \lambda t_{23}; \lambda\gamma_4; \lambda t_{33} \\ &= \gamma_5 \dagger \gamma_6; \eta_{t_6}; \lambda(t_6^1\, \epsilon_{61}; \epsilon_{62})\end{aligned}$$

so

$$
\begin{aligned}
(\gamma_1 \dagger \gamma_2) \dagger \gamma_3 &= \alpha_1; (\gamma_5 \dagger \gamma_6)\, (\gamma_{12} \dagger \gamma_{22};\, ; \epsilon_{61}[t_{32}]); \epsilon_{62}[t_{32}] \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, \gamma_{12}; t_5\, t_{22} \dagger (t_5^1\, 1; \gamma_6^1\, (1; 1; \epsilon_{61}); \epsilon_{62})[\gamma_{22}] \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, \gamma_{12}; t_5\, t_{22} \dagger (t_5^1\, 1; \mathcal{I}(t_5^1)\, (1; 1; \epsilon_{51}); \epsilon_{52} \dagger \gamma_4)[\gamma_{22}] \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, (\gamma_{12};\, ; \epsilon_{51}[t_{22}]); \epsilon_{52}[t_{22}] \dagger \gamma_4[\gamma_{22}] \dagger \gamma_{31} \\
&= \gamma_1 \dagger (\gamma_2 \dagger \gamma_3)
\end{aligned}
$$

case 1.3: $\gamma_{21} \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} = \alpha_4; \lambda\gamma_4; \lambda t_{33}$,
$\gamma_{11} \dagger \alpha_4; \lambda\mathcal{I}(t_{23}); \lambda t_{23} = \gamma_5; \eta_{t_5}; \lambda(t_5^1\, \epsilon_{51}; \epsilon_{52})$,
$t_5^1\, 1; \mathcal{I}(t_5^1)\, (1; 1; \epsilon_{51}); \epsilon_{52} \dagger \gamma_4 = \alpha_6^1\, 1; \beta_{t_6,1}; \gamma_6$
and $\gamma_5 \dagger \alpha_6; \lambda\gamma_6; \lambda t_{33} = \alpha_7; \lambda\gamma_7; \lambda t_{33}$.

Then

$$
\begin{aligned}
(\gamma_{11} \dagger \gamma_{21}) \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} &= \gamma_{11} \dagger \alpha_4; \lambda\gamma_4; \lambda t_{33} \\
&= \gamma_5; \eta_{t_5}; \lambda(t_5^1\, \epsilon_{51}; \epsilon_{52}) \dagger \lambda t_{23}; \lambda\gamma_4; \lambda t_{33} \\
&= \gamma_5 \dagger \alpha_6; \lambda\gamma_6; \lambda t_{33} \\
&= \alpha_7; \lambda\gamma_7; \lambda t_{33}
\end{aligned}
$$

so

$$
\begin{aligned}
(\gamma_1 \dagger \gamma_2) \dagger \gamma_3 &= \alpha_1;\, ; \alpha_7\, t_{12}; \beta_{t_{13},t_{12}}; \gamma_7[\gamma_{12} \dagger \gamma_{22}] \dagger \gamma_{31} \\
&= \alpha_1; (\alpha_7; \lambda\gamma_7; \lambda t_{33})\, (\gamma_{12} \dagger \gamma_{22}); \lambda t_{33}\, t_{32} \\
&\qquad \dagger (\lambda t_{13}\, t_{12}; \beta_{t_{13},t_{12}}; \mathcal{I}(t_{13}[t_{12}])) \dagger \gamma_{31} \\
&= \alpha_1; (\gamma_5 \dagger \alpha_6; \lambda\gamma_6; \lambda t_{33})\, (\gamma_{12} \dagger \gamma_{22}); \lambda t_{33}\, t_{32} \\
&\qquad \dagger (\lambda t_{13}\, t_{12}; \beta_{t_{13},t_{12}}; \mathcal{I}(t_{13}[t_{12}])) \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, \gamma_{12}; t_5\, t_{22} \dagger \alpha_6\, t_{22}; \beta_{t_6,t_{22}}; \gamma_6[\gamma_{22}] \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, \gamma_{12}; t_5\, t_{22} \dagger t_5\, t_{22}; \mathcal{I}(t_5)\, (\mathcal{I}(t_{22});\, ; \epsilon_{51}[t_{22}]); \epsilon_{52}[t_{22}] \\
&\qquad \dagger \gamma_4[\gamma_{22}] \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, (\gamma_{12};\, ; \epsilon_{51}[t_{22}]); \epsilon_{52}[t_{22}] \dagger \gamma_4[\gamma_{22}] \dagger \gamma_{31} \\
&= \gamma_1 \dagger (\gamma_2 \dagger \gamma_3)
\end{aligned}
$$

case 1.4: $\gamma_{21} \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} = \alpha_4; \lambda\gamma_4; \lambda t_{33}$,
$\gamma_{11} \dagger \alpha_4; \lambda\mathcal{I}(t_{23}); \lambda t_{23} = \gamma_5; \eta_{t_5}; \lambda(t_5^1\, \epsilon_{51}; \epsilon_{52})$,
$t_5^1\, 1; \mathcal{I}(t_5^1)\, (1; 1; \epsilon_{51}); \epsilon_{52} \dagger \gamma_4 = \alpha_6^1\, 1; \beta_{t_6,1}; \gamma_6$
and $\gamma_5 \dagger \alpha_6; \lambda\gamma_6; \lambda t_{33} = \gamma_7; \eta_{t_7}; \lambda(t_7^1\, \epsilon_{71}; \epsilon_{72})$.

Then

$$
\begin{aligned}
(\gamma_{11} \dagger \gamma_{21}) \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} &= \gamma_{11} \dagger \alpha_4; \lambda\gamma_4; \lambda t_{33} \\
&= \gamma_5; \eta_{t_5}; \lambda(t_5^1\, \epsilon_{51}; \epsilon_{52}) \dagger \lambda t_{23}; \lambda\gamma_4; \lambda t_{33} \\
&= \gamma_5 \dagger \alpha_6; \lambda\gamma_6; \lambda t_{33} \\
&= \gamma_7; \eta_{t_7}; \lambda(t_7^1\, \epsilon_{71}; \epsilon_{72})
\end{aligned}
$$

so

$$
\begin{aligned}
(\gamma_1 \dagger \gamma_2) \dagger \gamma_3 &= \alpha_1; \gamma_7\, (\gamma_{12} \dagger \gamma_{22};; \epsilon_{71}[t_{32}]); \epsilon_{72}[t_{32}] \dagger \gamma_{31} \\
&= \alpha_1; (\gamma_7; \eta_{t_7}; \lambda(t_7^1\, \epsilon_{71}; \epsilon_{72})\, (\gamma_{12} \dagger \gamma_{22}); \lambda t_{33}\, t_{32} \\
&\qquad \dagger (\lambda t_{13}\, t_{12}; \beta_{t_{13},t_{12}}; \mathcal{I}(t_{13}[t_{12}])) \dagger \gamma_{31} \\
&= \alpha_1; (\gamma_5 \dagger \alpha_6; \lambda\gamma_6; \lambda t_{33})\, (\gamma_{12} \dagger \gamma_{22}); \lambda t_{33}\, t_{32} \\
&\qquad \dagger (\lambda t_{13}\, t_{12}; \beta_{t_{13},t_{12}}; \mathcal{I}(t_{13}[t_{12}])) \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, \gamma_{12}; t_5\, t_{22} \dagger \alpha_6\, t_{22}; \beta_{t_6,t_{22}}; \gamma_6[\gamma_{22}] \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, \gamma_{12}; t_5\, t_{22} \dagger t_5\, t_{22}; \mathcal{I}(t_5)\, (\mathcal{I}(t_{22});; \epsilon_{51}[t_{22}]); \epsilon_{52}[t_{22}] \\
&\qquad \dagger \gamma_4[\gamma_{22}] \dagger \gamma_{31} \\
&= \alpha_1; \gamma_5\, (\gamma_{12};; \epsilon_{51}[t_{22}]); \epsilon_{52}[t_{22}] \dagger \gamma_4[\gamma_{22}] \dagger \gamma_{31} \\
&= \gamma_1 \dagger (\gamma_2 \dagger \gamma_3)
\end{aligned}
$$

case 1.5: $\gamma_{21} \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} = \gamma_4; \eta_{t_4}; \lambda(t_4^1\, \epsilon_{41}; \epsilon_{42})$.

Then

$$
(\gamma_{11} \dagger \gamma_{21}) \dagger \alpha_3; \lambda\mathcal{I}(t_{33}); \lambda t_{33} = \gamma_{11} \dagger \gamma_4; \eta_{t_4}; \lambda(t_4^1\, \epsilon_{41}; \epsilon_{42})
$$

so

$$
\begin{aligned}
(\gamma_1 \dagger \gamma_2) \dagger \gamma_3 &= \alpha_1; (\gamma_{11} \dagger \gamma_4)\, (\gamma_{12} \dagger \gamma_{22};; \epsilon_{41}[t_{32}]); \epsilon_{42}[t_{32}] \dagger \gamma_{31} \\
&= \gamma_1 \dagger (\gamma_2 \dagger \gamma_3)
\end{aligned}
$$

This completes the first main case.

case 2: $\gamma_1 = \gamma_{11}; \eta_{t_{11}}; \lambda(t_{11}^1\, \epsilon_{11}; \epsilon_{12})$,
$\gamma_2 = \lambda t_{21}; \lambda\gamma_{21}; \lambda t_{31}$
and $\gamma_3 = \lambda t_{31}; \lambda\gamma_{31}; \epsilon_3$.

Again there are several subcases:

case 2.1: $t_{11}^1\, 1; \mathcal{I}(t_{11}^1)\, (1; 1; \epsilon_{11}); \epsilon_{12} \dagger \gamma_{21} = t_{11}^1\, 1; \gamma_4\, (1; 1; \epsilon_{41}); \epsilon_{42}$
and $t_4^1\, 1; \mathcal{I}(t_4^1)\, (1; 1; \epsilon_{41}); \epsilon_{42} \dagger \gamma_{31} = t_4^1\, 1; \gamma_5\, (1; 1; \epsilon_{51}); \epsilon_{52}$.

Then

$$\begin{aligned} t_{11}^1\,1; \mathcal{I}(t_{11}^1)\,(1;1;\epsilon_{11}); \epsilon_{12} \dagger (\gamma_{21} \dagger \gamma_{31}) &= t_{11}^1\,1; \gamma_4\,(1;1;\epsilon_{41}); \epsilon_{42} \dagger \gamma_{31} \\ &= t_{11}^1\,1; \gamma_4\,\mathcal{I}(1); t_4^1\,1 \dagger t_4^1\,1; \gamma_5\,(1;1;\epsilon_{51}); \epsilon_{52} \\ &= t_{11}^1\,1; (\gamma_4 \dagger \gamma_5)\,(1;1;\epsilon_{51}); \epsilon_{52} \end{aligned}$$

so

$$\begin{aligned} \gamma_1 \dagger (\gamma_2 \dagger \gamma_3) &= \gamma_{11} \dagger (\gamma_4 \dagger \gamma_5); \eta_{t_5}; \lambda(t_5^1\,\epsilon_{51}; \epsilon_{52}) \\ &= (\gamma_1 \dagger \gamma_2) \dagger \gamma_3 \end{aligned}$$

case 2.2: $t_{11}^1\,1; \mathcal{I}(t_{11}^1)\,(1;1;\epsilon_{11}); \epsilon_{12} \dagger \gamma_{21} = t_{11}^1\,1; \gamma_4\,(1;1;\epsilon_{41}); \epsilon_{42}$,
$t_4^1\,1; \mathcal{I}(t_4^1)\,(1;1;\epsilon_{41}); \epsilon_{42} \dagger \gamma_{31} = \alpha_5\,1; \beta_{t_5,1}; \gamma_5$
and $\gamma_4 \dagger \alpha_5; \lambda\mathcal{I}(t_5); \lambda t_5 = \alpha_6; \lambda\gamma_6; \lambda t_5$.

Then

$$\begin{aligned} t_{11}^1\,1; \mathcal{I}(t_{11}^1)\,(1;1;\epsilon_{11}); \epsilon_{12} \dagger (\gamma_{21} \dagger \gamma_{31}) &= t_{11}^1\,1; \gamma_4\,(1;1;\epsilon_{41}); \epsilon_{42} \dagger \gamma_{31} \\ &= t_{11}^1\,1; \gamma_4\,\mathcal{I}(1); t_4^1\,1 \dagger \alpha_5\,1; \beta_{t_5,1}; \gamma_5 \\ &= \alpha_6\,1; \beta_{t_6,1}; \gamma_6 \dagger \gamma_5 \end{aligned}$$

so

$$\begin{aligned} \gamma_1 \dagger (\gamma_2 \dagger \gamma_3) &= \gamma_{11} \dagger \alpha_6; \lambda(\gamma_6 \dagger \gamma_5); \epsilon_3 \\ &= \gamma_{11} \dagger \gamma_4 \dagger \alpha_5; \lambda\gamma_5; \epsilon_3 \\ &= (\gamma_1 \dagger \gamma_2) \dagger \gamma_3 \end{aligned}$$

case 2.3: $t_{11}^1\,1; \mathcal{I}(t_{11}^1)\,(1;1;\epsilon_{11}); \epsilon_{12} \dagger \gamma_{21} = t_{11}^1\,1; \gamma_4\,(1;1;\epsilon_{41}); \epsilon_{42}$,
$t_4^1\,1; \mathcal{I}(t_4^1)\,(1;1;\epsilon_{41}); \epsilon_{42} \dagger \gamma_{31} = \alpha_5\,1; \beta_{t_5,1}; \gamma_5$
$\gamma_4 \dagger \alpha_5; \lambda\mathcal{I}(t_5); \lambda t_5 = \gamma_6; \eta_{t_6}; \lambda(t_6^1\,\epsilon_{61}; \epsilon_{62})$
and $t_{11}^1; \gamma_6\,(1;1;\epsilon_{61}); \epsilon_{62} \dagger \gamma_5 = t_{11}^1\,1; \gamma_7\,(1;1;\epsilon_{71}); \epsilon_{72}$.

Then

$$\begin{aligned} t_{11}^1\,1; \mathcal{I}(t_{11}^1)\,(1;1;\epsilon_{11}); \epsilon_{12} \dagger (\gamma_{21} \dagger \gamma_{31}) &= t_{11}^1\,1; \gamma_4\,(1;1;\epsilon_{41}); \epsilon_{42} \dagger \gamma_{31} \\ &= t_{11}^1\,1; \gamma_4\,\mathcal{I}(1); t_4^1\,1 \dagger \alpha_5\,1; \beta_{t_5,1}; \gamma_5 \\ &= t_{11}^1; \gamma_6\,(1;1;\epsilon_{61}); \epsilon_{62} \dagger \gamma_5 \\ &= t_{11}^1\,1; \gamma_7\,(1;1;\epsilon_{71}); \epsilon_{72} \end{aligned}$$

so

$$\begin{aligned}\gamma_1 \dagger (\gamma_2 \dagger \gamma_3) &= \gamma_{11} \dagger \gamma_7; \eta_{t_7}; \lambda(t_7^1\, \epsilon_{71}; \epsilon_{72});\,; \epsilon_3 \\ &= \gamma_{11} \dagger \mathcal{I}(t_{11}); \eta_{t_{11}}; \lambda(t_{11}^1\, 1; t_{11}^1\, 1) \\ &\qquad \dagger \lambda(t_{11}^1\, 1); \lambda(t_{11}^1; \gamma_6\, (1; 1; \epsilon_{61}); \epsilon_{62} \dagger \gamma_5); \epsilon_3 \\ &= \gamma_{11} \dagger \gamma_6; \eta_{t_6}; \lambda(t_6^1\, \epsilon_{61}; \epsilon_{62}) \dagger \lambda t_5; \lambda\gamma_5; \epsilon_3 \\ &= \gamma_{11} \dagger \gamma_4 \dagger \alpha_5; \lambda\gamma_5; \epsilon_3 \\ &= (\gamma_1 \dagger \gamma_2) \dagger \gamma_3\end{aligned}$$

case 2.4: $t_{11}^1\, 1; \mathcal{I}(t_{11}^1)\, (1; 1; \epsilon_{11}); \epsilon_{12} \dagger \gamma_{21} = t_{11}^1\, 1; \gamma_4\, (1; 1; \epsilon_{41}); \epsilon_{42}$,
$t_4^1\, 1; \mathcal{I}(t_4^1)\, (1; 1; \epsilon_{41}); \epsilon_{42} \dagger \gamma_{31} = \alpha_5\, 1; \beta_{t_5,1}; \gamma_5$
$\gamma_4 \dagger \alpha_5; \lambda\mathcal{I}(t_5); \lambda t_5 = \gamma_6; \eta_{t_6}; \lambda(t_6^1\, \epsilon_{61}; \epsilon_{62})$
and $t_{11}^1; \gamma_6\, (1; 1; \epsilon_{61}); \epsilon_{62} \dagger \gamma_5 = \alpha_7\, 1; \beta_{t_7,1}; \gamma_7$.

Then

$$\begin{aligned}t_{11}^1\, 1; \mathcal{I}(t_{11}^1)\, (1; 1; \epsilon_{11}); \epsilon_{12} \dagger (\gamma_{21} \dagger \gamma_{31}) &= t_{11}^1\, 1; \gamma_4\, (1; 1; \epsilon_{41}); \epsilon_{42} \dagger \gamma_{31} \\ &= t_{11}^1\, 1; \gamma_4\, \mathcal{I}(1); t_4^1\, 1 \dagger \alpha_5\, 1; \beta_{t_5,1}; \gamma_5 \\ &= t_{11}^1; \gamma_6\, (1; 1; \epsilon_{61}); \epsilon_{62} \dagger \gamma_5 \\ &= \alpha_7\, 1; \beta_{t_7,1}; \gamma_7\end{aligned}$$

so

$$\begin{aligned}\gamma_1 \dagger (\gamma_2 \dagger \gamma_3) &= \gamma_{11} \dagger \alpha_7; \lambda\gamma_7; \epsilon_3 \\ &= \gamma_{11} \dagger \mathcal{I}(t_{11}); \eta_{t_{11}}; \lambda(t_{11}^1\, 1; t_{11}^1\, 1) \\ &\qquad \dagger \lambda(t_{11}^1\, 1); \lambda(t_{11}^1; \gamma_6\, (1; 1; \epsilon_{61}); \epsilon_{62} \dagger \gamma_5); \epsilon_3 \\ &= \gamma_{11} \dagger \gamma_6; \eta_{t_6}; \lambda(t_6^1\, \epsilon_{61}; \epsilon_{62}) \dagger \lambda t_5; \lambda\gamma_5; \epsilon_3 \\ &= \gamma_{11} \dagger \gamma_4 \dagger \alpha_5; \lambda\gamma_5; \epsilon_3 \\ &= (\gamma_1 \dagger \gamma_2) \dagger \gamma_3\end{aligned}$$

case 2.5: $t_{11}^1\, 1; \mathcal{I}(t_{11}^1)\, (1; 1; \epsilon_{11}); \epsilon_{12} \dagger \gamma_{21} = \alpha_4\, 1; \beta_{t_4,1}; \gamma_4$.

Then

$$t_{11}^1\, 1; \mathcal{I}(t_{11}^1)\, (1; 1; \epsilon_{11}); \epsilon_{12} \dagger (\gamma_{21} \dagger \gamma_{31}) = \alpha_4\, 1; \beta_{t_4,1}; \gamma_4 \dagger \gamma_{31}$$

so

$$\begin{aligned}\gamma_1 \dagger (\gamma_2 \dagger \gamma_3) &= \gamma_{11} \dagger \alpha_4; \lambda(\gamma_4 \dagger \gamma_{31}); \epsilon_3 \\ &= (\gamma_1 \dagger \gamma_2) \dagger \gamma_3\end{aligned}$$

This completes the proof. □

# References

[1] Y. Akama. On Mints' reductions for ccc-calculus. In M. Bezem and J. Groote, editors, *Typed Lambda Calculus and Applications*, number 664 in Lecture Notes in Computer Science, pages 1–12. Springer-Verlag, 1993.

[2] H. P. Barendregt. *The Lambda Calculus.* Number 103 in Studies in Logic and the Foundations of Mathematics. North-Holland, revised edition, 1984.

[3] T. Coquand and G. Huet. The calculus of constructions. *Information and Computation*, 76(2/3):95–120, 1988.

[4] N. G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation. *Indag. Math.*, 34:381–392, 1972.

[5] R. Di Cosmo and D. Kesner. A confluent reduction for the extensional typed $\lambda$-calculus with pairs, sums, recursion and terminal object. In A. Lingas, editor, *International Conference on Automata, Languages and Programming*, number 700 in Lecture Notes in Computer Science, pages 645–656. Springer-Verlag, July 1993.

[6] R. Di Cosmo and D. Kesner. Combining first order algebraic rewriting systems, recursion and extensional lambda calculi. In S. Abiteboul and E. Shamir, editors, *International Conference on Automata, Languages and Programming*, number 820 in Lecture Notes in Computer Science, pages 462–472. Springer-Verlag, July 1994.

[7] R. Di Cosmo and D. Kesner. Simulating expansions without expansions. *Mathematical Structures in Computer Science*, 4(3):315–362, Sept. 1994.

[8] P. Gabriel and M. Zisman. *Calculus of Fractions and Homotopy Theory.* Number 35 in Ergebnisse der Mathematik. Springer-Verlag, 1967.

[9] J.-Y. Girard, Y. Lafont, and P. J. Taylor. *Proofs and Types.* Number 7 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[10] J. W. Gray. *Formal Category Theory I: Adjointness for 2-categories.* Number 391 in Lecture Notes in Mathematics. Springer-Verlag, 1974.

[11] G. Huet. *Résolution d'équations dans des langages d'ordre* $1, 2, \ldots, \omega$. Thèse d'etat, Université de Paris VII, 1976.

[12] C. B. Jay. Long $\beta\eta$ normal forms and confluence. LFCS Report Series ECS-LFCS-91-183, University of Edinburgh, 1991.

[13] C. B. Jay and N. Ghani. The virtues of eta-expansion. LFCS Report Series ECS-LFCS-92-243, University of Edinburgh, Oct. 1992.

[14] J.-J. Lévy. Optimal reductions in the lambda calculus. In J. R. Hindley and J. P. Seldin, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism*, pages 159–192. Academic Press, 1980.

[15] D. E. Rydeheard and J. Stell. Foundations of equational deduction: A categorical treatment of equational proofs and unification algorithms. In D. Pitt, A. Poigné, and D. E. Rydeheard, editors, *Category Theory and Computer Science*, number 283 in Lecture Notes in Computer Science, pages 114–139. Springer-Verlag, September 1987.

[16] R. A. G. Seely. Modeling computations: A 2-categorical framework. In *Proceedings of the Second Symposium on Logic in Computer Science*, pages 65–71. IEEE Computer Science Press, June 1987.