Higher-Order Rewriting with Dependent Types

Roberto Virga

October 20, 1999

CMU-CS-99-167

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematical Sciences

Thesis Commitee:
Frank Pfenning, Chair
Peter Andrews
Robert Harper
Richard Statman

This research was sponsored in part by NSF Grant CCR-9619584
The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of NSF or the U.S. government

Keywords: meta logic, logical frameworks, lambda calculus, type theory, rewriting, equational logic

Abstract

Higher-order, typed logic programming languages such as lambda-Prolog and Twelf have emerged in this last decade as the solution to many of the shortcomings of first-order languages like Prolog. A strongly-typed system allows early detection of many common programming mistakes, and it accounts for smaller and faster runtime environments. Lambda abstraction provides proper representation to the concept of bound variable, ubiquitous in mathematics. Because of these features, these languages have found in the last few years applications in several interesting areas, such as security protocols for mobile code. Unfortunately, use of these tools also evidenced a few problems in these languages, mostly due to the inadequate representation of equality that they offer.

Early development of Prolog faced similar problems, which were ultimately tackled through two different approaches. Both of these founded their roots in the theory of Term Rewriting Systems, but evolved along quite different paths. *Constraint Logic Programming* extended Prolog by integrating decision procedures for specific theories to the inference engine. *Rewriting Logic* offered a new approach to logic programming, where term rewriting, rather than proof search, was used as model of computation.

We believe that the development of higher-order languages, in order to address their current inadequacies, should evolve along similar paths. For this reason, we study in this dissertation a theory of Higher-order Term Rewriting for the LF calculus, on which Twelf is based.

The results presented here can be divided in two parts. In the first part,we analyze an extension to LF where types can also be converted modulo an equational theory. As the LF calculus is very sensitive to changes in the type conversion relation used, all the confluence properties of β-reduction and (restricted) η-expansion, and existence of normal forms have to re-examined. We show that our extension is conservative, and all the usual properties continue to hold, albeit in a weaker form.

In the second part, we proceed in defining a notion of rewriting. Since in a dependently typed calculus terms are allowed to appear inside types, a naive definition, extending the one given by Nipkow for simply typed calculi, is inapplicable, as it may lead to situations where rewriting invalidates the type of an expression. Hence, we turn our attention to the study of special preorders, called dependence relations, that allow us to extract type information that is implicit in a LF signature, and use it to restrict rewriting to well-behaved instances.

Dependence relations turn out also to be useful when studying confluence of rewriting, which, as usual, can be reduced to checking some special rewriting configurations known as *critical pairs*. Together with a general Critical Pair Criterion, we present a specialized version, where fewer critical pairs need to be checked thanks to the type information conveyed by these preorders.

A few examples conclude the presentation, demonstrating some of the potential of the concepts introduced. We present applications to Milner's Action and Process Calculi, Category Theory, and Proof Theory.

Acknowledgments

First and foremost, I would like to thank my advisor, Frank Pfenning, for his guidance, support, and encouragement during the development of this thesis. I feel I have matured during my stay here at Carnegie Mellon, both as a researcher and as a person, and much of this I owe to him.

I am also grateful to the other members of my thesis committee. To Rick Statman, who first introduced me to the fascinating subject of the lambda calculus. To Bob Harper, whose courses helped me get a better insight of the techniques and the ideas involved in the design and implementation of programming languages. To Peter Andrews, who taught me the importance of rigor and precision in doing mathematics.

I am thankful to all the professors and researchers who participate to the interdisciplinary program of Pure and Applied Logic. I've benefited a lot from this program, which, through their enthusiasm and dedication, keeps being one of the best offered in this country. Of particular mention here are Rami Grossberg, the aforementioned Peter Andrews, and Wilfried Sieg who, through frequently offered logic seminars, constantly stimulate independent research and exchange of ideas.

One big thanks goes to Carsten Schürmann and Alberto Momigliano, who have always been invaluable friends, as well as the source of many useful suggestions in writing this thesis.

Last, but certainly not least, I am grateful to my family, who waited patiently to see this work completed. I would also like to remember my grandmother, who I'm sure would be very happy for me right now; although she is not not with me in person, she is in spirit, as she'll always have a special place in my heart.

Contents

List of Figures

Chapter 1

Introduction

1.1 Historical Background and Motivations

Mathematical logic is often the result of the synergistic combination of two distinct techniques: computation and deduction. Computation involves the symbolic manipulation of expressions, which are transformed according to some specified equivalences. Deduction is the process of deriving logical consequences from a set of initial postulates called axioms, through the application of some fixed rules of inference.

To see how these two processes interact, consider the following example. We define even numbers as integers that are multiples of two; this is formalized by the single axiom

$$\forall n \in \mathbb{Z}.\ \mathbf{even}\ (2 \cdot n)$$

Logical rules allow us to specialize this general statement to, say, the case $n = 5$:

$$\frac{\forall n \in \mathbb{Z}.\ \mathbf{even}\ (2 \cdot n)}{\mathbf{even}\ (2 \cdot 5)}$$

This gets us a proof that $(2 \cdot 5)$ is even; computing this latter expression, we conclude that 10 is even.

Unfortunately, early attempts to apply the principles of mathematical logic to programming language design failed to realize the importance of both processes, and leaned in favor of a purely deductive approach. Probably the most important byproduct of these attempts is the Prolog language [69], which based its syntax on a subset of prefix-quantified first-order logic (Horn clauses), and its semantics on depth-first proof search. As efficient compilation techniques emerged [73], and Prolog could finally be tested on real-life applications, some of the limitations of its design clearly emerged. Upon closer inspection, most of these drawbacks can be traced back, as expected, to the inadequate treatment of the equality predicate that the language provides. Integer and floating-point arithmetic, an essential feature in any general-purpose language, did not fit in the operational semantic of pure Prolog, and had to be implemented as extra-logical constructs. Also, the inability to treat cases of "don't care" non-determinism (any choice leads to success, and therefore one can be picked at random) any differently from cases of "don't know" non-determinism (only some of the possible choices lead to success, and backtrack points have to be introduced) made it difficult for programmers to generate efficient code, and motivated the introduction of yet another extra-logical construct (cut).

The proliferation of such extra-logical extensions led researchers to the search for paradigms that, while addressing these shortcomings, had a more logical underpinning, and therefore could be studied formally. Such search unfolded in all directions, concluding in two very different solutions.

The first, *Constraint Logic Programming* [15, 32], was evolutional, as it extended Prolog operational semantic rather than subverting it. Decision procedures for specific domains (e.g. floating-point numbers, sets, strings) were added to the Prolog engine in order to solve many commonly encountered equations. "Hard" equations, which could not be handled by these procedures, were deferred as constraints until they could be furtherly simplified. A natural evolution of this approach can be seen in Frühwirth's *Constraint Handling Rules* (CHR) [18], that generalize this type of Prolog extensions by providing a general language for the manipulation of user-definable constraints. This language is based on conditional rewriting, and non-determinism stemming from the simultaneous applicability of more than one rule is resolved in a "don't care" fashion.

Another approach was more revolutionary in nature, as it tried to look at the problem of using logic as a foundation of a programming language from a fresh new perspective. This led to *Rewriting Logic* [40], where the operational semantics of the language was chosen to be term rewriting rather than proof search. Many advantages stem from such choice: arithmetic and other equational theories are elegantly handled; the possibility of specifying different reduction strategies give the user more control over how non-deterministic choices are resolved during program execution.

Concurrently to these activities, the ground-breaking work of Martin Löf on constructive type theory [41, 42] influenced the development of many systems based on higher-order logic and type theory, with applications to several fields of computer science, from program synthesis [38, 11, 16] to automated theorem proving [57].

Martin Löf's ideas eventually made their way into logic programming, producing a new generation of typed, higher-order languages, which finds its most illustrious representatives in lambda-Prolog [46] and Twelf [63]. These offer several advantages over their first-order counterparts. The use of a type system makes the task of isolating meaningless expressions easier, and therefore leads to earlier detection of many common programming mistakes. Lambda-abstraction provides a convenient way to represent the notion of bound variable, that is used pervasively in mathematics. Built-in β-reduction offers a computational component in an otherwise purely deductive programming paradigm.

In recent years, as implementations of these languages have matured, their full potential is finally beginning to be explored. This has brought applications of extreme significance; of these, we briefly describe here Necula's *proof-carrying code* technique [51]. The main idea of this technique is to endow compiled code with a proof of its correctness. In a mobile code environment, where a program (applet) residing on a server is transmitted to a client for execution, carrying along these proofs in the transmission bears several advantages. On the one hand, it provides security, since the the proof and the code can be checked against each other, to make sure that the latter has not been tampered with. On the other hand, it translates into better performance, since proofs ensure that the code is well-behaved, making it unnecessary for it to be run in a secure environment (sandbox). The meta-representational capabilities of Twelf made it the ideal language to write these proofs in.

However, if this application outlined the strengths of the language, it also pointed out one of its weaknesses. Often, correctness proofs rely on equational properties about the machine data types (e.g. integers, floating-point numbers, booleans, strings). Since Twelf, like its first-order ancestor Prolog, does not provide primitives to deal with equality, all these properties must be stated (and proved) explicitly, increasing the size of the proofs.

It is our thesis that the proper way to address the current shortcomings of higher-order logic programming languages is to follow the example of what has been done at the first-order level, and to go through the development of a theory of rewriting for richly typed systems. Such development should certainly draw upon the seminal work done by Nipkow [54] and Prehofer [65] on higher-order rewriting for the simply-typed lambda calculus. However, an array of new issues arises when we consider the more complex type systems on which languages like Twelf are based, and these issues have hence to be analyzed and solved. In this dissertation we make a first step in this direction, by studying rewriting in the LF calculus, which provides the foundation for the Twelf language.

Once the theory of rewriting we advocate is in place, two new paths of development open up, mirroring at the higher-order level the evolution of Prolog we described before. One path, which goes through on the development of equational, constraint-based extensions to the host language, has already been partially investigated by us, through the development of a extension of Twelf [72]. The other direction, suggesting the definition of new languages based on higher-order versions of Rewriting Logic, has not yet been explored, to the best of our knowledge, but certainly looks promising and deserving of further study in the future.

```
item : type.

list : nat -> type.

nil : list 0.
,   : item -> list N -> list (1 + N). %infix right 50 ,.

append : list M -> list N -> list (M + N) -> type.

append_nil  : append nil L2 L2.
append_cons : append L1 L2 L3 -> append (X , L1) L2 (X , L3).
```

Figure 1.1: List Concatenation - Equational

1.2 Some Preliminary Examples

To give the flavor of how the equational theories and rewriting systems we propose could be used in practice, we present here a couple of preliminary examples, based on Twelf. More complex examples can be found in the final part of this dissertation.

Concatenation of lists is an operation which is traditionally represented as a relation rather than a function, and therefore fits well inside the programming paradigm offered by Twelf. Having dependent types at our disposal, however, it becomes possible for us to specify general invariants about this operation, that any (correct) implementation should satisfy. Moreover, it is desirable that, whenever one such implementation is provided, these invariants are automatically verified by the type-checking mechanism, without any user intervention. In this specific case, we observe that concatenating two lists, of length m and n respectively, should always produce a list of length $(m + n)$. Using this observation as invariant, we have the Twelf signature of Figure 1.1.

The implementation of `append` we chose in this example is recursive on the first argument. In order verify it conforms to the invariant, we need the type-checking algorithm to automatically apply a few simple arithmetical equations. These are, as the reader can easily verify, the following:

$$0 + M = M \qquad\qquad 1 + (M + N) = (1 + M) + N$$

Unfortunately, before our results, such equational extensions of the type-check mechanism could not be theoretically justified, and were consequently not implemented. In Figure 1.2 we show how the same example has to be represented in the current version of Twelf. Note that equality needs to be expressed explicitly, and that the programmer is responsible for providing proofs (which are often long and complex) of all the equations he/she needs.

The second example we present is from Category Theory. As we will have the opportunity to show in Chapter 12, categorical notions have a very strong equational component, and therefore lend themselves naturally to be represented by term rewriting systems. In this section, we focus on SUBST, a fragment of the axioms of Cartesian Closed Category, that has been proved in [25] to be orientable into a strongly-normalizing (i.e. rewriting always terminates in a finite number of steps), confluent (i.e. normal forms are unique) system. A representation of SUBST in a Twelf-like syntax is given in Figure 1.3.

With dependent types, we can store information about the source and target of a morphism inside its type; a consequence of this is, for example, that composition of morphism is well-typed only in those cases when it is well-defined.

Since this formalization captures all and only well-formed entities, proving that rewriting is closed with respect to the set of terms representing valid morphisms is unnecessary, as this is automatically guaranteed by the calculus. Moreover, "catch-all" rules like the ones for the terminal object (see Figure 1.4) can be added to the system without causing it to collapse to the trivial one, equating all terms.

```
item : type.

eq : nat -> nat -> type.

eq_refl  : eq M M.
eq_symm  : eq M N -> eq N M.
eq_trans : eq M N -> eq N P -> eq M P.

eq_congl : {M : nat} eq N N' -> eq (M + N) (M + N').
eq_congr : {M : nat} eq N N' -> eq (N + M) (N' + M).

eq_idl   : eq (0 + M) M.
eq_idr   : eq (M + 0) M.
eq_assoc : eq ((M + N) + P) (M + (N + P)).

list : nat -> type.

nil : list 0.
,   : item -> list N -> list (1 + N). %infix right 50 ,.

append : list M -> list N -> list P -> eq (M + N) P -> type.

append_nil  : append nil L2 L2 eq_idl.
append_cons : append (X , L1) L2 (X , L3) (eq_trans (eq_congl 1 D) (eq_symm eq_assoc))
                <- append L1 L2 L3 D.
```

Figure 1.2: List Concatenation - Non-Equational Version

1.3 Outline of This Dissertation

The two examples presented in the previous sections correspond to the two parts in which the results contained in this dissertation are divided.

In the first part we investigate an equational extension to LF, where type conversion is performed modulo an equational theory $\mathcal{E}$, as well as modulo the traditional β- and η-relations. This can be seen as the theoretical justification to the first example presented in Section 1.2. LF is very sensitive to changes in the type-conversion notion employed: the proofs presented in [67] for $\beta\eta$-conversion are more complex, and often quite different, from the one found in the original LF paper by Harper et al. [27], which uses only β-conversion. Hence, results about confluence and existence of normal forms for our equational extension can not be reasonably assumed to hold, and have to be re-examined. We indeed show that the one proposed is a conservative extension, and in particular long-η β-normal forms (we will call these forms *canonical*) still exist, and are unique, modulo conversion of the types of the lambda-abstracted variables. Uniqueness in the absolute sense is not achievable, as one can easily see by considering the famous Nederpelt's counterexample [52] in this context

$$\begin{tikzcd} & \lambda x : A_1.\ (\lambda x : A_2.\ M)x \arrow[dl, "\beta"'] \arrow[dr, "\eta"] & \\ \lambda x : A_1.\ M & & \lambda x : A_2.\ M \end{tikzcd}$$

and observing that the types A_1 and A_2 may have been obtained one from the other through conversion modulo $\mathcal{E}$. Hence our uniqueness result is the best that we could hope for, given the circumstances, and it proves to be enough to the development of the remaining of our dissertation.

Part I can be furtherly subdivided as follows. In Chapter 2, we introduce the type system and prove

```
obj   : type.

prod  : obj -> obj -> type.  %infix right 50 prod.
arrow : obj -> obj -> type.  %infix right 50 arrow.

morph : obj -> obj -> type.

id : {A : obj} morph A A.
o  : morph B C -> morph A B -> morph A C.  %infix right 50 o.

pair : morph C A -> morph C B -> morph C (A prod B).

pr1 : morph (A prod B) A.
pr2 : morph (A prod B) B.

lambda : morph (A prod B) C -> morph A (B prod C).

fst   : pr1 o (pair S T) => S.
snd   : pr2 o (pair S T) => T.
idl   : (id B) o T => T.
idr   : T o (id A) => T.
spair : pair (pr1 o T) (pr2 o T) => T.
assoc : (S o T) o U => S o (T o U).
dpair : (pair S T) o U => pair (S o U) (T o U).
dlam  : (lambda S) o T => lambda (S o (pair (T o pr1) pr2))
fsid  : pair pr1 pr2 => id C.
```

Figure 1.3: The Rewrite System SUBST

```
1 : obj.

t : {A : obj} morph A 1.

tall : F => (t A).
```

Figure 1.4: Adding a Terminal Object to SUBST

its most basic properties; these will be extensively used throughout the thesis as the foundation over which most of the theory is constructed. In Chapter 3 we study β-reduction, showing it to be Church-Rosser (Theorem 3.3.2) and strong normalizing (Theorem 3.3.10) by using a modification of the proof found in [28]. We work in a slightly different (but easily seen equivalent) system (Definition 3.1.1, where some terms are marked with their type. This is necessary to carry out a construction, due to Dowek et al. [17] (Theorem 3.4.1, that will be instrumental in proving the existence of canonical forms. Chapter 4 wraps up this first part, by studying the behavior of η-conversion and by using all the machinery developed in the previous two to prove confluence of the system with respect to β-reduction and η-expansion (Theorem 4.4.20).

The second part tackles the problem of giving a definition of a *Higher-order Term Rewriting System* (HTRS) in this setting. This relies, as usual, on the notion of contextual replacement of expression, which however in this case, unlike the simply-typed one investigated by Nipkow [54], appears to be problematic. Replacing only expressions with equal type no longer suffice to ensure the well-typedness of the overall expression, as we demonstrate in Chapter 5. The search for conditions that guarantee the well-behavior properties we desire motivates us to study special preorders called *dependence relations*. Using these notions, we are able to formulate a sound theory of rewriting, and to prove a series of results about it, culminating in a higher-order version of the Knuth-Bendix's Critical Pair Criterion [30].

Looking at the structure of Part II at a finer level of detail, we have the following divisions. In Chapter 5, dependence relations, already introduced in Part I, are thoroughly discussed. We explain the main ideas behind their introduction, and their defining properties. In Chapter 6 we define term rewriting. We provide, together with the natural definition (Definition 6.2.3), a second, more technical one(Definition 6.3.1), that we prove equivalent to the first(Theorem 6.3.14), and which will be more convenient to use when studying the properties of rewriting systems. Chapter 7 deals with those cases when the equational theory $\mathcal{E}$ is "simpler" than the HTRS $\mathcal{R}$, i.e. it is defined on a class of types that lay below the type hierarchy than the one over which $\mathcal{R}$ is defined. Simpler equational theories can be easily detected by using dependence relations (Definition 7.1.2). We prove that in these cases no critical overlaps are possible between an equation in $\mathcal{E}$ and a rule in $\mathcal{R}$. This leads to a simplified version of the Critical Pair Criterion (Theorem 7.4.3), where only critical pairs coming from overlaps of rules (Definition 7.3.2) need to be considered. Finally, in Chapter 8 we deal with the general case, where overlaps between equations and rewrite rules are possible. The Critical Pair Criterion in its general form is stated and proved (Theorem 8.3.7); we require here equations to have patterns on both sides, as now critical overlaps are possible (Definition 8.3.1).

These two parts, which contain the main theoretical results of this work, are followed by a third one, where we show some of the applications of the theory. The three examples we picked come from very different areas of theoretical computer science, thus demonstrating the wide range of applicability of higher-order rewriting techniques. The first of these, developed in Chapter 11, deals with concurrency, formalized by Milner's *action structures* and *process calculus*. We show how the algebraic nature of action structures make them an ideal candidate for being represented by higher-order equational theories. We also show that weak head-normal form reduction in the process calculus can be modeled by a strongly normalizing, confluent system.

Our second example focuses on Category Theory. That categorical notions could be elegantly represented by dependent-type rewriting had already been shown by Gehrke in [19]. In Chapter 12, we give an alternative formulation to the one he adopted, which makes a better use of the type structure.

Finally, one of the strengths of LF is the ability to be used as a logical framework to describe different logical systems. In this context, rewriting can be used to express proof-transformation technique within a single object logic or between two different ones. In Chapter 13, we present term rewriting systems to convert between intuitionistic natural deduction and sequent calculus, as well as a cut-elimination system that produces cut-free sequent derivations.

Part I

Equational LF

Chapter 2

Preliminaries

In this chapter we present an equational version of the LF calculus. Type conversion is expressed explicitly, in the form of an equality judgment. An implicit representation, more common in LF literature, is usually better suited to systems where only β- or $\beta\eta$-conversions are used, and confluence is proven for the contractionary direction of these conversions. In these conditions, subject reduction holds, and therefore well-formed terms can be guaranteed to admit valid normal forms. In our case, however, the presence of $\mathcal{E}$-conversion, which in general cannot be oriented, as well as the need to consider η-conversion in its expansionary direction (more about this will be said at the beginning of Chapter 4) make it necessary to express conversion as an equational judgement.

2.1 Equational LF

The equational version of the LF calculus, which we will concentrate upon in this dissertation, is described as follows:

Definition 2.1.1. The LF calculus is a three-level calculus for *objects*, *type families*, and *kinds*

$$\begin{array}{llll} \textit{Objects} & M & := & c \mid x \mid \lambda x : A.\ M \mid M\ N \\ \textit{Type Families} & A & := & a \mid \Pi x : A.\ B \mid A\ M \\ \textit{Kinds} & K & := & \text{type} \mid \Pi x : A.\ K \end{array}$$

Notation. We will use the letters M, N, U for objects, A, B, C, D for types, K, L for kinds. The letters c and a will range over object and type family constants, respectively; x, y, z, u, v will be used for object variables.

Terms of the calculus (i.e. objects, type families, and kinds) will always be be considered equal modulo α-conversion, i.e. renaming of the variables bounded by Π and λ.

By $\mathcal{FV}(M)$ (respectively: $\mathcal{FV}(A), \mathcal{FV}(K)$) we will denote the set of variables that are free in the object M (respectively: type family A, kind K).

The notation $[N/x]M$ will be used to indicate the replacement of all the free occurrences of the variable x by N in the object M; replacement inside a type family ($[N/x]A$), or a kind ($[N/x]K$) is similarly defined.

We will assume the usual notions of β- and η-reduction, as the smallest relations containing all the pairs

$$\begin{array}{l} (\lambda x : A.\ M)\ N \to_\beta [N/x]M \\ \lambda x : A.\ (M\ x) \to_\eta M, \text{ provided } x \notin \mathcal{FV}(M) \end{array}$$

and compatible with the term structure.

Given any binary relation R over terms of the calculus, we will denote by $R^=$, R^+, and R^* its reflexive, transitive, and reflexive-transitive closures, respectively.

An object M will be said to be *a* λ if $M = \lambda x : A.\ N$. Similarly, a type family A is *a* Π if $A = \Pi x : B.\ C$. A type A that is not a Π is said to be *a base type*. We will use the letter P to indicate base types.

In what follows, we will define well-formedness of terms of the calculus by assigning types to objects, and kinds to types. In order to do so, however we need to introduce the concepts of signatures, that map well-formed object and type constants to their type or kind, contexts, that will play a similar role for free variables, and equational theories, that tell us which types are to be considered equivalent.

Definition 2.1.2. We define *signatures*, *equational theories*, and *contexts*:

$$\begin{array}{llll} \textit{Signatures} & \Sigma & := & \cdot \mid \Sigma, c : A \mid \Sigma, a : K \\ \textit{Equational Theories} & \mathcal{E} & := & \cdot \mid \mathcal{E}, (\Gamma \vdash M = N : A) \\ \textit{Contexts} & \Gamma & := & \cdot \mid \Gamma, x : A \end{array}$$

Notation. We will use Γ and Δ to range over contexts, $\mathcal{E}$ for equational theories, and Σ for signatures.

We will often see signatures and contexts as functions over sets, and borrow some of the traditional set-theoretic notation. For example, given a context

$$\Gamma = x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n$$

its domain will be, as usual,

$$\mathrm{dom}(\Gamma) = x_1, x_2, \ldots, x_n$$

Example 1. The following simple signature could be used to represent propositional logic formulas

$$\mathrm{dom}(\Sigma) = \{\mathbf{o}, \mathbf{not}, \mathbf{and}, \mathbf{or}, \mathbf{implies}\}$$

$$\begin{aligned} &\Sigma(\mathbf{o}) = \mathrm{type} \\ &\Sigma(\mathbf{not}) = \Pi x : \mathbf{o}.\ \mathbf{o} \\ &\Sigma(\mathbf{and}) = \Pi x : \mathbf{o}.\ \Pi y : \mathbf{o}.\ \mathbf{o} \\ &\Sigma(\mathbf{or}) = \Pi x : \mathbf{o}.\ \Pi y : \mathbf{o}.\ \mathbf{o} \\ &\Sigma(\mathbf{implies}) = \Pi x : \mathbf{o}.\ \Pi y : \mathbf{o}.\ \mathbf{o} \end{aligned}$$

We can also define an equational theory over this signature, for example to describe some well known classical logic equivalences:

$$\begin{aligned} x : \mathbf{o} &\vdash \mathbf{not}\ (\mathbf{not}\ x) = x\ \ : \mathbf{o} \\ x : \mathbf{o}, y : \mathbf{o} &\vdash \mathbf{not}\ (\mathbf{and}\ x\ y) = \mathbf{or}\ (\mathbf{not}\ x)\ (\mathbf{not}\ y)\ \ : \mathbf{o} \\ x : \mathbf{o}, y : \mathbf{o} &\vdash \mathbf{not}\ (\mathbf{or}\ x\ y) = \mathbf{and}\ (\mathbf{not}\ x)\ (\mathbf{not}\ y)\ \ : \mathbf{o} \\ x : \mathbf{o}, y : \mathbf{o} &\vdash \mathbf{implies}\ x\ y = \mathbf{or}\ (\mathbf{not}\ x)\ y\ \ : \mathbf{o} \end{aligned}$$

We will make use here of the definition of *dependence relation* introduced by us in [71].

Definition 2.1.3 (Dependence Relation). A *dependence relation* $\prec$ for a signature Σ is a pair of binary relations $\prec^{\tau}$ and $\prec^{\mathrm{t}}$ defined over the set of type family constants of Σ, and satisfying

- $\mathrm{head}(A_i) \prec^{\tau} a$ if $\Sigma(a) = \Pi x_1 : A_1.\ \Pi x_2 : A_2.\ \ldots \Pi x_n : A_n.$ type
- $a \prec^{\tau} a'$ if $a \prec^{\tau} b \prec^{\mathrm{t}} a'$ for some b
- $a \prec^{\tau} a'$ if $a \prec^{\mathrm{t}} b \prec^{\tau} a'$ for some b
- $a \prec^{\mathrm{t}} a'$ if $a \prec^{\mathrm{t}} b \prec^{\mathrm{t}} a'$ for some b
- $a \prec^{\mathrm{t}} a'$ if $a \prec^{\tau} a'$

where the *head* of a type family head(A) is defined as follows:

$$\begin{aligned}\text{head}(a) &= a\\ \text{head}(\Pi x : A.\ B) &= \text{head}(B)\\ \text{head}(A\ M) &= \text{head}(A)\end{aligned}$$

Notation. The definitions of the binary relations $\prec^\tau$ and $\prec^{\mathrm{t}}$ are extended to pairs of type families over Σ by letting

$$A \prec^\tau B \text{ iff } \text{head}(A) \prec^\tau \text{head}(B) \qquad\qquad A \prec^{\mathrm{t}} B \text{ iff } \text{head}(A) \prec^{\mathrm{t}} \text{head}(B)$$

We will also use the notation

$$\begin{aligned}A \preceq^\tau B &\text{ iff } A \prec^\tau B \text{ or } \text{head}(A) = \text{head}(B)\\ A \preceq^{\mathrm{t}} B &\text{ iff } A \prec^{\mathrm{t}} B \text{ or } \text{head}(A) = \text{head}(B)\end{aligned}$$

Dependence relations will be used when defining rewriting, in Chapter 6, to keep track of the types of objects that may appear inside LF expressions. Specifically, we will show that $A \prec^{\mathrm{t}} B$ only if it is possible for an object of type A to be contained inside an object of type B, and similarly $A \prec^\tau B$ only if objects of type A may appear inside the type family B.

As dependence relations will play no role in the results presented in Chapters 2-4, we will delay any further discussion of their motivations and formal properties to Chapter 5.

Definition 2.1.4. Well-formed objects, type families, and kinds are constructed using the judgments

$$\begin{array}{ll}\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K\ \mathit{Kind} & K \text{ is a kind}\\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K & A \text{ is a type of kind } K\\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A & M \text{ is an object of type } A\end{array}$$

These, in turn, are formed by means of the auxiliary judgments

$$\begin{array}{ll}\vdash^{\prec}_{\mathcal{E}} \Sigma\ \mathit{Sig} & \Sigma \text{ is a valid signature}\\ \vdash^{\prec}_{\Sigma} \mathcal{E}\ \mathit{Eq} & \mathcal{E} \text{ is a valid equational theory}\\ \vdash^{\prec}_{\Sigma,\mathcal{E}} \Gamma\ \mathit{Ctx} & \Gamma \text{ is a valid context}\end{array}$$

specifying how valid signatures, equational theories, and contexts are built.

Conversion between pairs of kinds, type families, and objects is defined making use of *substitutions*:

$$\mathit{Substitutions}\quad \theta \quad ::= \quad \cdot \mid \theta, x \mapsto N$$

and expressed by the judgments

$$\begin{array}{ll}\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K' & K \text{ and } K' \text{ are } \mathcal{E}'\beta\eta\text{-convertible kinds}\\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' & A \text{ and } A' \text{ are } \mathcal{E}'\beta\eta\text{-convertible type families}\\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M' & M \text{ and } M' \text{ are } \mathcal{E}'\beta\eta\text{-convertible objects}\end{array}$$

where $\mathcal{E}' \subseteq \mathcal{E}$, and

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \quad \theta \text{ is a valid substitution from } \Delta \text{ to } \Gamma$$

The rules for the calculus are listed in Figure 2.1

$$\frac{\Sigma(c) = A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} c : A} \qquad \frac{\Gamma(x) = A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} x : A} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} M : B \quad A \preceq^{\mathrm{t}} B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M : \Pi x : A.\ B}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N : [N/x]B} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' : \mathrm{type}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A'}$$

$$\frac{\Sigma(a) = K}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} a : K} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} B : \mathrm{type} \quad A \preceq^{\mathrm{t}} B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B : \mathrm{type}}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \Pi x : B.\ K \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A\ M : [M/x]K} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}\beta\eta} K' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K'\ Kind}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K'}$$

$$\frac{}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \mathrm{type}\ \ Kind} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ K\ Kind}$$

$$\frac{}{\vdash^{\prec}_{\mathcal{E}} \cdot\ Sig} \qquad \frac{\vdash^{\prec}_{\mathcal{E}} \Sigma\ Sig \quad \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type}}{\vdash^{\prec}_{\mathcal{E}} \Sigma, c : A\ Sig} \qquad \frac{\vdash^{\prec}_{\mathcal{E}} \Sigma\ Sig \quad \vdash^{\prec}_{\Sigma,E} K\ Kind}{\vdash^{\prec}_{\mathcal{E}} \Sigma, a : K\ Sig}$$

$$\frac{}{\vdash^{\prec}_{\Sigma} \cdot\ Eq} \qquad \frac{\vdash^{\prec}_{\Sigma} \mathcal{E}\ Eq \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : P \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : P}{\vdash^{\prec}_{\Sigma} \mathcal{E}, (\Gamma \vdash M = N : P)\ Eq}$$

$$\frac{}{\vdash^{\prec}_{\Sigma,\mathcal{E}} \cdot\ Ctx} \qquad \frac{\vdash^{\prec}_{\Sigma,\mathcal{E}} \Gamma\ Ctx \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type}}{\vdash^{\prec}_{\Sigma,\mathcal{E}} \Gamma, x : A\ Ctx}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : A.\ M)\ N : C}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : A.\ M)\ N =_{\mathcal{E}'\beta\eta} [N/x]M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} \lambda x : A.\ (M\ x)}$$

$$\frac{(\Delta \vdash_{\Sigma,\mathcal{E}} M = N : A) \in \mathcal{E}' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta M =_{\mathcal{E}'\beta\eta} \theta N}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' =_{\mathcal{E}'\beta\eta} M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' =_{\mathcal{E}'\beta\eta} M''}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M''}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M =_{\mathcal{E}'\beta\eta} \lambda x : A'.\ M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M =_{\mathcal{E}'\beta\eta} \lambda x : A.\ M'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N =_{\mathcal{E}'\beta\eta} M'\ N} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N =_{\mathcal{E}'\beta\eta} N'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N =_{\mathcal{E}'\beta\eta} M\ N'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' =_{\mathcal{E}'\beta\eta} A} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' =_{\mathcal{E}'\beta\eta} A''}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A''}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} B : \mathrm{type}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B =_{\mathcal{E}'\beta\eta} \Pi x : A'.\ B} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}'\beta\eta} B'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B =_{\mathcal{E}'\beta\eta} \Pi x : A.\ B'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A\ M =_{\mathcal{E}'\beta\eta} A'\ M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \Pi x : B.\ K \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A\ M =_{\mathcal{E}'\beta\eta} A\ M'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K' =_{\mathcal{E}'\beta\eta} K} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K' =_{\mathcal{E}'\beta\eta} K''}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K''}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ K =_{\mathcal{E}'\beta\eta} \Pi x : A'.\ K} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ K =_{\mathcal{E}'\beta\eta} \Pi x : A.\ K'}$$

$$\frac{}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \cdot : \cdot} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : \theta A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta, x \mapsto N : \Delta, x : A}$$

Figure 2.1: Typing Rules for the LF Calculus

Remark. A few observations about the introduced definitions are necessary.

First, we defined conversion as a parametric judgment over an equational theory $\mathcal{E}' \subseteq \mathcal{E}$. This allows us to group under a common notation both $\beta\eta$-conversion ($\mathcal{E}' = \emptyset$) and full conversion modulo $\mathcal{E}$ and $\beta\eta$ ($\mathcal{E}' = \mathcal{E}$). Although we will occasionally prove results that apply only to one of these two conversion notions, most of the results presented hold for both, and this expedient allows us to greatly simplify and shorten our presentation.

A similar desire not to overburden the reader with unnecessary notation has led us to avoid tagging the conversion statement with the type of the expressions being converted. A small price we have to pay for this is to allow conversion on some classes of non-well-formed terms. In formulating our definitions, however, we were careful in requiring that the conversion steps we were mostly concerned about, namely β- and η-expansions, were always performed on well-formed objects.

Finally, the LF calculus presented here differs from the one found in [59] in two important aspects. The first is that conversion is presented here as a judgment, rather than being defined on raw objects. In this respect, this presentation is more similar to [26]. The motivation behind this choice is twofold: we wanted to give a uniform description of both $\mathcal{E}$- and $\beta\eta$-conversion, and we also needed a way to control η-expansion, so that Subject Reduction holds.

The second difference is that we do not allow abstraction over types. Being able to abstract over type families adds very little power to the calculus, and has the disadvantage of making the Uniqueness of Products property:

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B =_{\mathcal{E}'\beta\eta} \Pi x : A'.\ B' \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \text{ and } \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}'\beta\eta} B'$$

very hard to prove.

Notation. To simplify the notation, we stipulate, from now on, and throughout this thesis, that all the contexts, signatures, and equational theories mentioned are well-typed.

Example 1. The signature Σ presented before, where

$$\mathrm{dom}(\Sigma) = \{\mathbf{o}, \mathbf{not}, \mathbf{and}, \mathbf{or}, \mathbf{implies}\}$$

can be proven to be well-formed for some choices of dependence relations $\prec$. The most obvious candidate for one such relation, in this case as well as in general, is the full one $\prec_{\mathrm{F}}$:

$$\prec_{\mathrm{F}}{}^{\tau} = \prec_{\mathrm{F}}{}^{\mathrm{t}} = \{(a, a') \mid a, a' \in \mathrm{dom}(\Sigma)\}$$

This choice can be easily seen to always satisfy the conditions given in Definitions 2.1.3 and 2.1.4. This simple observation forms the basis of our decision of postpone the discussion of these relations to Chapter 5, where some other, less crude choices, where available, will become highly desirable.

At the other side of the spectrum, we might ask ourselves what are the minimal (with respect to inclusion) relations that make Σ well-formed. Definition 2.1.3 can be satisfied by $\prec_0{}^{\tau} = \prec_0{}^{\mathrm{t}} = \emptyset$. This seem to suffice in Definition 2.1.4, where for functional objects of type $\Pi x : A.\ B$ we are required to check $A \preceq^{\mathrm{t}} B$; since

$$\mathbf{o} \preceq^{\mathrm{t}} \mathbf{o} \qquad\qquad \mathbf{o} \preceq^{\mathrm{t}} \Pi x : \mathbf{o}.\ \mathbf{o}$$

in this special, fortunate case the empty dependence relation seems to work as well. Moreover, it is immediate to see that any dependence relation $\prec_0 \subseteq \prec \subseteq \prec_{\mathrm{F}}$ that satisfies Definition 2.1.3 will also well-type Σ (this is formally proved in Proposition 2.2.1 below). Hence, for example, we have also

$$\prec^{\tau} = \emptyset \qquad\qquad \prec^{\mathrm{t}} = \{(\mathbf{o}, \mathbf{o})\}$$

Notation. A special class of substitutions that will be used frequently are identity substitutions. We define a special notation for them: given a context Γ, we will denote by id_Γ the identity substitution over Γ

$$id_\Gamma = \{x \mapsto x \mid x : A \in \Gamma\}$$

It is convenient in our presentation to extend the conversion judgment to substitutions:

Definition 2.1.5. We introduce the judgment

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta =_{\mathcal{E}'\beta\eta} \theta' \quad \theta \text{ and } \theta' \text{ are } \mathcal{E}'\beta\eta\text{-convertible substitutions}$$

constructed according to the following two rules

$$\frac{}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \cdot =_{\mathcal{E}'\beta\eta} \cdot} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta =_{\mathcal{E}'\beta\eta} \theta' \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N =_{\mathcal{E}'\beta\eta} N'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta, x \mapsto N =_{\mathcal{E}'\beta\eta} \theta', x \mapsto N'}$$

Finally, a concept introduced by D. Miller [47] that will play an important role in the theory we are going to develop is that of a (higher-order) *pattern*:

Definition 2.1.6. An object $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A$ is said to be a pattern if every free occurrence of a variable x in Γ) appears enclosed in sub-objects of the form $M_\circ = x\ N_1\ N_2 \ldots N_k$ and the N_i are all $\beta\eta$-convertible to distinct bound variables. A sub-object $M_\circ$ of the above form is called a generalized variable.

2.2 Basic Properties

We list now a few basic properties of the judgments defined above. Unless specified otherwise, all the proofs are obtained by a simple (and somewhat tedious) induction on the derivation, and will be omitted.

Proposition 2.2.1 (Weakening). *Let* $\prec \subseteq \prec'$, $\Sigma \subseteq \Sigma'$, $\mathcal{E}'' \subseteq \mathcal{E} \subseteq \mathcal{E}'$, $\mathcal{E}'' \subseteq \mathcal{E}''' \subseteq \mathcal{E}'$, $\Gamma \subseteq \Gamma'$,

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ implies } \Gamma' \vdash^{\prec'}_{\Sigma',\mathcal{E}'} M : A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}''\beta\eta} M' \textit{ implies } \Gamma' \vdash^{\prec'}_{\Sigma',\mathcal{E}'} M =_{\mathcal{E}'''\beta\eta} M'$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ implies } \Gamma' \vdash^{\prec'}_{\Sigma',\mathcal{E}'} A : K \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}''\beta\eta} A' \textit{ implies } \Gamma' \vdash^{\prec'}_{\Sigma',\mathcal{E}'} A =_{\mathcal{E}'''\beta\eta} A'$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind \textit{ implies } \Gamma' \vdash^{\prec'}_{\Sigma',\mathcal{E}'} K\ Kind \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}''\beta\eta} K' \textit{ implies } \Gamma' \vdash^{\prec'}_{\Sigma',\mathcal{E}'} K =_{\mathcal{E}'''\beta\eta} K'$$

Proposition 2.2.2. *Assume $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta : \Delta$, then*

$$\Delta \vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta M : \theta A \qquad \Delta \vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta M =_{\mathcal{E}'\beta\eta} \theta M'$$
$$\Delta \vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta A : \theta K \qquad \Delta \vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta A =_{\mathcal{E}'\beta\eta} \theta A'$$
$$\Delta \vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta K \; Kind \qquad \Delta \vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}'\beta\eta} K' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta K =_{\mathcal{E}'\beta\eta} \theta K'$$

Corollary 2.2.3 (Substitution). *Let $\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} N : C$,*

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M : [N/x]A$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]A : [N/x]K$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]K \; Kind$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M =_{\mathcal{E}'\beta\eta} [N/x]M'$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]A =_{\mathcal{E}'\beta\eta} [N/x]A'$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}'\beta\eta} K' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]K =_{\mathcal{E}'\beta\eta} [N/x]K'$$

Proof. First, by using Proposition 2.2.2 repeatedly, we obtain from the assumptions a derivation of

$$\Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} (id_{\Gamma_1}, x \mapsto N, id_{\Gamma_2}) : \Gamma_1, x : C, \Gamma_2$$

Then, one final application of Proposition 2.2.2 concludes the proof. □

Corollary 2.2.4. *Let $\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} C =_{\mathcal{E}'\beta\eta} C' : \mathrm{type}$,*

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ implies } \Gamma_1, x : C', \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M : A$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ implies } \Gamma_1, x : C', \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} A : K$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind \textit{ implies } \Gamma_1, x : C', \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind$$

Proof. We exploit the same strategy as in Corollary 2.2.3: using the type conversion rule, we show

$$\Gamma_1, x : C', \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} id_{\Gamma_1, x:C, \Gamma_2} : \Gamma_1, x : C, \Gamma_2$$

and then we apply Proposition 2.2.2. □

Lemma 2.2.5 (Type Uniqueness). *Types of well-formed objects are unique up to conversion:*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}\beta\eta} A'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : K' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}\beta\eta} K'$$

Lemma 2.2.6 (Type Consistency).

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma'} A : \mathrm{type} \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma'} K \; Kind$$

Proposition 2.2.7.

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M' : A$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M' : A \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A' : K$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A' : K \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : K$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}'\beta\eta} K' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} K' \; Kind$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} K' \; Kind \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}'\beta\eta} K' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind$$

Proof. By induction on the derivations of $\beta\eta$-equivalence, using inversion on the derivations of well-typedness. The cases of the three conversion rules might be of some interest:

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} (\lambda x : A.\ M)\ N : C}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} (\lambda x : A.\ M)\ N =_{\mathcal{E}'\beta\eta} [N/x]M}$$

We want to prove $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} [N/x]M : C$. By inversion on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} (\lambda x : A.\ M)\ N : C$, using type conversion if necessary, we can show

$$\begin{aligned}
&\Gamma, x : A \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : B\\
&\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N : A\\
&\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} [N/x]B =_{\mathcal{E}'\beta\eta} C
\end{aligned}$$

Thus, the result follows from Substitution:

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} [N/x]M : [N/x]B$$

and type conversion.

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} \lambda x : A.\ (M\ x)}$$

We want to show $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : A.\ (M\ x) : \Pi x : A.\ B$. By Type Consistency and inversion, $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A : \mathrm{type}$ and $A \preceq^{\mathrm{t}} B$; hence by Weakening

$$\Gamma, x : A \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B$$

and the result follows by using the application and abstraction rules.

- Case

$$\frac{(\Delta \vdash M = N : A) \in \mathcal{E}' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta : \Delta}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta M =_{\mathcal{E}'\beta\eta} \theta N}$$

Immediate by Proposition 2.2.2.

$\square$

Corollary 2.2.8. *We have:*

$$\begin{aligned}
\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta : \Delta \text{ and } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta =_{\mathcal{E}'\beta\eta} \theta' \text{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta' : \Delta\\
\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta' : \Delta \text{ and } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta =_{\mathcal{E}'\beta\eta} \theta' \text{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta : \Delta
\end{aligned}$$

Proposition 2.2.9. *Let $\mathcal{E}' \subseteq \mathcal{E}$, the following are derived rules:*

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta : \Delta}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta =_{\mathcal{E}'\beta\eta} \theta} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta =_{\mathcal{E}'\beta\eta} \theta'}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta' =_{\mathcal{E}'\beta\eta} \theta} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta =_{\mathcal{E}'\beta\eta} \theta' \quad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta' =_{\mathcal{E}'\beta\eta} \theta''}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta =_{\mathcal{E}'\beta\eta} \theta''}$$

Proposition 2.2.10. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta : \Delta$ *and* $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta =_{\mathcal{E}'\beta\eta} \theta'$*,*

$$\begin{aligned}
\Delta \vdash^{\prec}_{\Sigma,\varepsilon} M : A \ &implies\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta M =_{\mathcal{E}'\beta\eta} \theta' M \\
\Delta \vdash^{\prec}_{\Sigma,\varepsilon} A : K \ &implies\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta A =_{\mathcal{E}'\beta\eta} \theta' A \\
\Delta \vdash^{\prec}_{\Sigma,\varepsilon} K\ Kind \ &implies\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta K =_{\mathcal{E}'\beta\eta} \theta' K
\end{aligned}$$

Corollary 2.2.11. *Let* $\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} N : C$ *and* $\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} N =_{\mathcal{E}'\beta\eta} N'$*,*

$$\begin{aligned}
\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M : A \ &implies\ \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M =_{\mathcal{E}'\beta\eta} [N'/x]M \\
\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} A : K \ &implies\ \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]A =_{\mathcal{E}'\beta\eta} [N'/x]A \\
\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} K\ Kind \ &implies\ \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]K =_{\mathcal{E}'\beta\eta} [N'/x]K
\end{aligned}$$

Proof. The proof is very similar to the one of Corollary 2.2.3. Using Propositions 2.2.2 and 2.2.10 we show

$$\begin{aligned}
&\Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} (id_{\Gamma_1}, x \mapsto N, id_{\Gamma_2}) : \Gamma_1, x : C, \Gamma_2 \\
&\Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} (id_{\Gamma_1}, x \mapsto N, id_{\Gamma_2}) =_{\mathcal{E}'\beta\eta} (id_{\Gamma_1}, x \mapsto N', id_{\Gamma_2})
\end{aligned}$$

and the result follows by one final application of Proposition 2.2.10. □

2.3 Summary

In Section 2.1 we defined the rules of the calculus. This definition relies on several typing judgments, which have to be all introduce at once, since each is defined recursively in terms of (some of) the others.

The properties proven in Section 2.2 are those which are usually found in most presentations of LF. The fact that they continue to hold here, and that their proofs are, mutatis mutandis, the same as those found in the literature reassures us on the well-behaved nature of the extensions we propose.

Chapter 3

Properties of β-Reduction

We start our study of $\beta\eta$-conversion by analyzing the behavior of β-reduction. We will show that it is confluent and strong-normalizing, and hence β-normal forms exist and are unique. Of course, this result alone does not suffice, since in general η-convertible expressions may have different β-normal forms, but it will turn out to be useful in the next chapter, where we will build upon the theory developed here to deal with full $\beta\eta$-conversion.

3.1 Marked LF

In this chapter we work with a slightly richer system than the one presented in Chapter 2, where some objects and type families are marked by their types and kinds, respectively.

Working in such system, as we will see, does not increase significantly the complexity of the proofs, and, on the other hand, it allows us to prove a result (Theorem 3.4.1) that will play a key role in showing the existence of long-η β-normal forms in Chapter 4.

Definition 3.1.1. Marked objects, type families, and kinds are constructed according the following syntax:

$$
\begin{array}{llll}
\textit{Marked Objects} & M & := & c^A \mid x^A \mid (\lambda x : A.\ M)^B \mid (M\ N)^A \\
\textit{Marked Type Families} & A & := & a^K \mid \Pi x : A.\ B \mid (A\ M)^K \\
\textit{Marked Kinds} & K & := & \mathrm{type} \mid \Pi x : A.\ K
\end{array}
$$

Well-formed terms are built, as before, making use of marked signatures, equational theories, contexts, and substitutions:

$$
\begin{array}{llll}
\textit{Marked Signatures} & \Sigma & := & \cdot \mid \Sigma, c : A \mid \Sigma, a : K \\
\textit{Marked Equational Theories} & \mathcal{E} & := & \cdot \mid \mathcal{E}, (\Gamma \Vdash M = N : A) \\
\textit{Marked Contexts} & \Gamma & := & \cdot \mid \Gamma, x : A \\
\textit{Marked Substitutions} & \theta & := & \cdot \mid \theta, x \mapsto N
\end{array}
$$

according to the inference system shown in Figure 3.1.

The intuition behind this marked extension is to embed the type of well-formed objects inside the object itself. Doing this will allow us to use more powerful inductive principles than the ones used so far, arguing by well-founded induction on both an object and its type.

Most of the remarks made about Definition 2.1.4 still hold here. In particular, the restrictions we impose on the conversion judgment are, again, just enough to guarantee that conversion of well-founded terms give rise to other well-founded terms (in the spirit of Proposition 2.2.7), but not sufficient to enforce conversion to be defined only on well-typed terms. This accounts for some apparent oddities in the conversion rules listed in Figure 3.1, where there seem no condition regarding the type used in the mark.

The only point of divergence between the definitions used here from the ones adopted in Chapter 2 involves the notion of replacement of a free variable x by an object N. Replacing inside marked terms

$$\frac{\Sigma(c) = A}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} c^{A} : A} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} c^{B} : A \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}\beta\eta} B' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} B' : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} c^{B'} : A}$$

$$\frac{\Gamma(x) = A}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} x^{A} : A} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} x^{B} : A \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}\beta\eta} B' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} B' : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} x^{B'} : A}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : B \quad A \preceq^{\mathrm{t}} B}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : A.\ M)^{\Pi x : A.\ B} : \Pi x : A.\ B}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : C.\ M)^{B} : A \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}\beta\eta} B' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} B' : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : C.\ M)^{B'} : A}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} N : A}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (M\ N)^{[N/x]B} : [N/x]B} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (M\ N)^{B} : A \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}\beta\eta} B' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} B' : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (M\ N)^{B'} : A}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A' : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : A'}$$

$$\frac{\Sigma(a) = K}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} a^{K} : K} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} a^{L} : K \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} L =_{\mathcal{E}\beta\eta} L' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} L'\ Kind}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} a^{L'} : K}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : \Pi x : B.\ K \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : B}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (A\ M)^{[M/x]K} : [M/x]K} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (A\ M)^{L} : K \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} L =_{\mathcal{E}\beta\eta} L' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} L'\ Kind}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (A\ M)^{L'} : K}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\mathcal{E}} B : \mathrm{type} \quad A \preceq^{\mathrm{t}} B}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B : \mathrm{type}}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}\beta\eta} K' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K'\ Kind}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : K'}$$

$$\frac{}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \mathrm{type}\ \ Kind} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ K\ Kind}$$

$$\frac{}{\Vdash^{\prec}_{\mathcal{E}} \cdot\ Sig} \qquad \frac{\Vdash^{\prec}_{\mathcal{E}} \Sigma\ Sig \quad \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type}}{\Vdash^{\prec}_{\mathcal{E}} \Sigma, c : A\ Sig} \qquad \frac{\Vdash^{\prec}_{\mathcal{E}} \Sigma\ Sig \quad \Vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind}{\Vdash^{\prec}_{\mathcal{E}} \Sigma, a : K\ Sig}$$

$$\frac{}{\Vdash^{\prec}_{\Sigma} \cdot\ Eq} \qquad \frac{\Vdash^{\prec}_{\Sigma} \mathcal{E}\ Eq \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : P \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : P}{\Vdash^{\prec}_{\Sigma} \mathcal{E}, (\Gamma \Vdash M = N : P)\ Eq}$$

$$\frac{}{\Vdash^{\prec}_{\Sigma,\mathcal{E}} \cdot\ Ctx} \qquad \frac{\Vdash^{\prec}_{\Sigma,\mathcal{E}} \Gamma\ Ctx \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type}}{\Vdash^{\prec}_{\Sigma,\mathcal{E}} \Gamma, x : A\ Ctx}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} ((\lambda x : A.\ M)^C\ N)^B : D}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} ((\lambda x : A.\ M)^C\ N)^B =_{\mathcal{E}'\beta\eta} [N/x]M} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : \Pi x : A.\ B}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} (\lambda x : A.\ (M\ x)^B)^{\Pi x : A.\ B}}$$

$$\frac{(\Delta \Vdash M = N) \in \mathcal{E}' \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} \theta : \Delta}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} \theta M =_{\mathcal{E}'\beta\eta} \theta N}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : A}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M' =_{\mathcal{E}'\beta\eta} M} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M' =_{\mathcal{E}'\beta\eta} M''}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M''}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} c^A : C \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} c^A =_{\mathcal{E}'\beta\eta} c^{A'}} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} x^A : C \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} x^A =_{\mathcal{E}'\beta\eta} x^{A'}}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (\lambda x : A.\ M)^A : C \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (\lambda x : A.\ M)^A =_{\mathcal{E}'\beta\eta} (\lambda x : A.\ M)^{A'}} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (M\ N)^A : C \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (M\ N)^A =_{\mathcal{E}'\beta\eta} (M\ N)^{A'}}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \qquad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\varepsilon} M : B \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} C : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (\lambda x : A.\ M)^C =_{\mathcal{E}'\beta\eta} (\lambda x : A'.\ M)^C}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : \mathrm{type} \qquad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} C : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (\lambda x : A.\ M)^C =_{\mathcal{E}'\beta\eta} (\lambda x : A.\ M')^C}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} N : B \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} C : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (M\ N)^C =_{\mathcal{E}'\beta\eta} (M'\ N)^C}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : \Pi x : A.\ B \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} N =_{\mathcal{E}'\beta\eta} N' \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} C : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (M\ N)^C =_{\mathcal{E}'\beta\eta} (M\ N')^C}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : K}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A' =_{\mathcal{E}'\beta\eta} A} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A' =_{\mathcal{E}'\beta\eta} A''}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A''}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} a^K : L \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}'\beta\eta} K'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} a^K =_{\mathcal{E}'\beta\eta} a^{K'}} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (A\ N)^K : L \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}'\beta\eta} K'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (M\ N)^K =_{\mathcal{E}'\beta\eta} (A\ N)^{K'}}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \qquad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\varepsilon} B : \mathrm{type}}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} \Pi x : A.\ B =_{\mathcal{E}'\beta\eta} \Pi x : A'.\ B} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : \mathrm{type} \qquad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\varepsilon} B =_{\mathcal{E}'\beta\eta} B'}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} \Pi x : A.\ B =_{\mathcal{E}'\beta\eta} \Pi x : A.\ B'}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : B \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} L\ Kind}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (A\ M)^L =_{\mathcal{E}'\beta\eta} (A'\ M)^L}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : \Pi x : B.\ K \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} L\ Kind}{\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} (A\ M)^L =_{\mathcal{E}'\beta\eta} (A\ M')^L}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K'}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K' =_{\mathcal{E}'\beta\eta} K} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K' \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K' =_{\mathcal{E}'\beta\eta} K''}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K''}$$

$$\frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \quad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ K =_{\mathcal{E}'\beta\eta} \Pi x : A'.\ K} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K'}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ K =_{\mathcal{E}'\beta\eta} \Pi x : A.\ K'}$$

$$\frac{}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \cdot : \cdot} \qquad \frac{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \quad \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} N : \theta A}{\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \theta, x \mapsto N : \Delta, x : A}$$

Figure 3.1: Typing Rules for the Marked LF Calculus

involves stripping each occurrence of the variable x from his mark. More formally, replacement can be defined by the following recursive rules:

$$\begin{aligned}
[N/x]c^A &= c^{[N/x]A} \\
[N/x]x^A &= N \\
[N/x]y^A &= y^{[N/x]A} \\
[N/x](\lambda z : A_1.\ M)^{A_2} &= (\lambda z : [N/x]A_1.\ [N/x]M)^{[N/x]A_2} \\
[N/x](M_1\ M_2)^A &= ([N/x]M_1\ [N/x]M_2)^{[N/x]A} \\
\\
[N/x]a^K &= a^{[N/x]K} \\
[N/x](\Pi z : A_1.\ A_2) &= \Pi z : [N/x]A_1.\ [N/x]A_2 \\
[N/x](A\ M)^K &= ([N/x]A\ [N/x]M)^{[N/x]K} \\
\\
[N/x]\,\mathrm{type} &= \mathrm{type} \\
[N/x](\Pi z : A.\ K) &= \Pi z : [N/x]A.\ [N/x]K
\end{aligned}$$

The properties we presented in Chapter 2 for the unmarked LF calculus continue to hold in this new setting, and have similar proofs.

Proposition 3.1.2 (Weakening). *Let* $\prec \subseteq \prec'$, $\Sigma \subseteq \Sigma'$, $\mathcal{E}'' \subseteq \mathcal{E} \subseteq \mathcal{E}'$, $\mathcal{E}'' \subseteq \mathcal{E}''' \subseteq \mathcal{E}'$, $\Gamma \subseteq \Gamma'$,

$$\begin{array}{cc}
\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ implies } \Gamma' \Vdash^{\prec'}_{\Sigma',\mathcal{E}'} M : A & \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}''\beta\eta} M' \textit{ implies } \Gamma' \Vdash^{\prec'}_{\Sigma',\mathcal{E}'} M =_{\mathcal{E}'''\beta\eta} M' \\
\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ implies } \Gamma' \Vdash^{\prec'}_{\Sigma',\mathcal{E}'} A : K & \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}''\beta\eta} A' \textit{ implies } \Gamma' \Vdash^{\prec'}_{\Sigma',\mathcal{E}'''} A =_{\mathcal{E}'\beta\eta} A' \\
\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind \textit{ implies } \Gamma' \Vdash^{\prec'}_{\Sigma',\mathcal{E}'} K\ Kind & \Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}''\beta\eta} K' \textit{ implies } \Gamma' \Vdash^{\prec'}_{\Sigma',\mathcal{E}'} K =_{\mathcal{E}'''\beta\eta} K'
\end{array}$$

Proposition 3.1.3 (Substitution). *Let* $\Gamma_1 \Vdash^{\prec}_{\Sigma,\mathcal{E}} N : C$,

$$\begin{array}{c}
\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]M : [N/x]A \\
\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]A : [N/x]K \\
\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} K\ Kind \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]K\ Kind \\
\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]M =_{\mathcal{E}'\beta\eta} [N/x]M' \\
\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]A =_{\mathcal{E}'\beta\eta} [N/x]A' \\
\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]K =_{\mathcal{E}'\beta\eta} [N/x]K'
\end{array}$$

Proposition 3.1.4. *Let* $\Gamma_1 \Vdash^{\prec}_{\Sigma,\varepsilon} C =_{\mathcal{E}'\beta\eta} C' : \mathrm{type}$, *the following holds:*

$$\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} M : A \ \textit{implies}\ \Gamma_1, x : C', \Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} M : A$$
$$\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} A : K \ \textit{implies}\ \Gamma_1, x : C'\Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} A : K$$
$$\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} K\ Kind \ \textit{implies}\ \Gamma_1, x : C', \Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} K\ Kind$$

Lemma 3.1.5 (Type Uniqueness). *Types of well-formed objects are unique up to conversions:*

$$\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : A \ \textit{and}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : A' \ \textit{implies}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A'$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : K \ \textit{and}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : K' \ \textit{implies}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}'\beta\eta} K'$$

Proposition 3.1.6.

$$\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : A \ \textit{and}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\mathcal{E}'\beta\eta} M' \ \textit{implies}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M' : A$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : K \ \textit{and}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\mathcal{E}'\beta\eta} A' \ \textit{implies}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A' : K$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K\ Kind \ \textit{and}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K =_{\mathcal{E}'\beta\eta} K' \ \textit{implies}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K'\ Kind$$

Lemma 3.1.7 (Type Consistency).

$$\Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : A \ \textit{implies}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon'} A : \mathrm{type} \qquad\qquad \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : K \ \textit{implies}\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon'} K\ Kind$$

Proposition 3.1.8. *Let* $\Gamma_1 \Vdash^{\prec}_{\Sigma,\varepsilon} N : C$ *and* $\Gamma_1 \Vdash^{\prec}_{\Sigma,\varepsilon} N =_{\mathcal{E}'\beta\eta} N'$,

$$\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} M : A \ \textit{implies}\ \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M =_{\mathcal{E}'\beta\eta} [N'/x]M$$
$$\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} A : K \ \textit{implies}\ \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} [N/x]A =_{\mathcal{E}'\beta\eta} [N'/x]A$$
$$\Gamma_1, x : C, \Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} K\ Kind \ \textit{implies}\ \Gamma_1, [N/x]\Gamma_2 \Vdash^{\prec}_{\Sigma,\varepsilon} [N/x]K =_{\mathcal{E}'\beta\eta} [N'/x]K$$

Definition 3.1.9. We introduce the erasure map, which recursively strips an object of all its marks:

$$\begin{aligned}
(c^A)^\# &= c \\
(x^A)^\# &= x \\
((\lambda x : A.\ M)^B)^\# &= \lambda x : A^\#.\ M^\# \\
((M\ N)^A)^\# &= M^\#\ N^\# \\
(a^K)^\# &= a \\
(\Pi x : A.\ B)^\# &= \Pi x : A^\#.\ B^\# \\
((A\ M)^A)^\# &= A^\#\ M^\# \\
\mathrm{type}^\# &= \mathrm{type} \\
(\Pi x : A.\ K)^\# &= \Pi x : A^\#.\ K^\#
\end{aligned}$$

Figure 3.2: Erasure Map

This map extends in the obvious away to signatures Σ, equational theories $\mathcal{E}$, contexts Γ, and substitutions θ.

The following shows that, as observed earlier, marks are used to store the type of an object inside its structure:

Proposition 3.1.10.

$$\Gamma \Vdash^{\preceq}_{\Sigma} c^B : A \ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} c^B : B$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} x^B : A \ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} x^B : B$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (M\ N)^B : A \ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (M\ N)^B : B$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (\lambda x : C.\ M)^B : A \ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (\lambda x : C.\ M)^B : B$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} a^L : K \ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} a^L : L$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (A\ M)^L : K \ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (A\ M)^L : L$$

Proof. The result follows easily arguing by induction on the derivation: the last inference rule must be either the rule that created the mark, or one of the two conversion rules (the one for marks or the one for the overall term). □

3.2 Correspondence with the Unmarked Version

We present a sequence of results that link this marked version of LF to the usual one.

Proposition 3.2.1.

$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} M_i : A_i \ and\ (M_i)^{\#} = M\ (i = 1, 2),\ where\ \Gamma^{\#} \vdash^{\preceq}_{\Sigma^{\#},\varepsilon^{\#}} M : B,\ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} M_1 =_{\varepsilon\beta\eta} M_2$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} A_i : K_i \ and\ (A_i)^{\#} = A\ (i = 1, 2),\ where\ \Gamma^{\#} \vdash^{\preceq}_{\Sigma^{\#},\varepsilon^{\#}} A : L,\ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} A_1 =_{\varepsilon\beta\eta} A_2$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} K_i\ Kind \ and\ (K_i)^{\#} = K\ (i = 1, 2),\ where\ \Gamma^{\#} \vdash^{\preceq}_{\Sigma^{\#},\varepsilon^{\#}} K\ Kind,\ implies\ \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} K_1 =_{\varepsilon\beta\eta} K_2$$

Proof. The proof proceeds by induction on the unmarked derivation, arguing by inversion on the marked ones. All cases are dealt with by using the same technique. We show one:

- Case:

$$\frac{\Gamma^{\#} \vdash^{\preceq}_{\Sigma^{\#},\varepsilon^{\#}} A' : \mathrm{type} \qquad \Gamma^{\#}, x : A' \vdash^{\preceq}_{\Sigma^{\#},\varepsilon^{\#}} M' : B' \qquad A' \preceq^{\mathrm{t}} B'}{\Gamma^{\#} \vdash^{\preceq}_{\Sigma^{\#},\varepsilon^{\#}} \lambda x : A'.\ M' : \Pi x : A'.\ B'}$$

  Then $M_i = (\lambda x : A'_i.\ M'_i)^{D_i}$ $(i = 1, 2)$ for some tagged objects M'_i and types A'_i. By inversion on the derivation, we show

$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} A'_i : \mathrm{type}$$
$$\Gamma, x : A'_i \Vdash^{\preceq}_{\Sigma,\varepsilon} M'_i : B'_i$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} \Pi x : A'_i.\ B'_i =_{\varepsilon\beta\eta} D_i$$

  By inductive hypothesis, $\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} A'_1 =_{\varepsilon\beta\eta} A'_2$, hence, using Proposition 3.1.4,

$$\Gamma, x : A'_2 \Vdash^{\preceq}_{\Sigma,\varepsilon} M'_1 : B_1$$

  We can therefore apply the induction hypothesis once more to obtain $\Gamma, x : A'_2 \Vdash^{\preceq}_{\Sigma,\varepsilon} M'_1 =_{\varepsilon\beta\eta} M'_2$. By Type Consistency,

$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} D_i : \mathrm{type}\ (i = 1, 2)$$

  hence we can derive

$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (\lambda x : A'_1.\ M'_1)^{D_2} =_{\varepsilon\beta\eta} (\lambda x : A'_2.\ M'_1)^{D_2}$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (\lambda x : A'_2.\ M'_1)^{D_2} =_{\varepsilon\beta\eta} (\lambda x : A'_2.\ M'_2)^{D_2}$$

Moreover, using Proposition 3.1.6 and Type Uniqueness, we conclude $\Gamma, x : A_2' \Vdash^{\prec}_{\Sigma,\mathcal{E}} B_1 =_{\mathcal{E}\beta\eta} B_2$, that get us a derivation of

$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} D_1 =_{\mathcal{E}\beta\eta} D_2$$

and finally one of

$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : A_1'.\ M_1')^{D_1} =_{\mathcal{E}\beta\eta} (\lambda x : A_1'.\ M_1')^{D_2}$$

Two applications of the transitivity rule conclude this case.

□

Proposition 3.2.2. *We have:*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ implies } \Gamma^m \Vdash^{\prec}_{\Sigma^m,\mathcal{E}^m} M^m : A^m$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ implies } \Gamma^m \Vdash^{\prec}_{\Sigma^m,\mathcal{E}^m} A^m : K^m$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K \textit{ Kind implies } \Gamma^m \Vdash^{\prec}_{\Sigma^m,\mathcal{E}^m} K^m \textit{ Kind}$$
$$\vdash^{\prec}_{\mathcal{E}} \Sigma \textit{ Sig implies } \Vdash^{\prec}_{\mathcal{E}^m} \Sigma^m \textit{ Sig}$$
$$\vdash^{\prec}_{\Sigma} \mathcal{E} \textit{ Eq implies } \Vdash^{\prec}_{\Sigma^m} \mathcal{E}^m \textit{ Eq}$$
$$\vdash^{\prec}_{\Sigma,\mathcal{E}} \Gamma \textit{ Ctx implies } \Vdash^{\prec}_{\Sigma^m,\mathcal{E}^m} \Gamma^m \textit{ Ctx}$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \textit{ implies } \Gamma \Vdash^{\prec}_{\Sigma^m,\mathcal{E}^m} \theta^m : \Delta^m$$

and

$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (M_i)^m : A\ (i=1,2) \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 =_{\mathcal{E}'\beta\eta} M_2 \textit{ implies } \Gamma^m \Vdash^{\prec}_{\Sigma^m,\mathcal{E}^m} (M_1)^m =_{(\mathcal{E}')^m\beta\eta} (M_2)^m$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (A_i)^m : K\ (i=1,2) \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_1 =_{\mathcal{E}'\beta\eta} A_2 \textit{ implies } \Gamma^m \Vdash^{\prec}_{\Sigma^m,\mathcal{E}^m} (A_1)^m =_{(\mathcal{E}')^m\beta\eta} (A_2)^m$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} (K_i)^m \textit{ Kind } (i=1,2) \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} K_1 =_{\mathcal{E}'\beta\eta} K_2 \textit{ implies } \Gamma^m \Vdash^{\prec}_{\Sigma^m,\mathcal{E}^m} (K_1)^m =_{(\mathcal{E}')^m\beta\eta} (K_2)^m$$

for some $(M^m)^\# = M$, $(A^m)^\# = A$, $(K^m)^\# = K$, $(\Sigma^m)^\# = \Sigma$, $(\mathcal{E}^m)^\# = \mathcal{E}$, $((\mathcal{E}')^m)^\# = \mathcal{E}'$, $(\Gamma^m)^\# = \Gamma$, $(\Delta^m)^\# = \Delta$, *and* $(\theta^m)^\# = \theta$.

Proof. The proof proceeds by induction on the derivations. Proposition 3.2.1 is used when dealing with the conversion rules to make sure that the marks match up. □

Proposition 3.2.3. *The erasure map preserves well-typedness:*

$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ implies } \Gamma^\# \vdash^{\prec}_{\Sigma^\#,\mathcal{E}^\#} M^\# : A^\#$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ implies } \Gamma^\# \vdash^{\prec}_{\Sigma^\#,\mathcal{E}^\#} A^\# : K^\#$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K \textit{ Kind implies } \Gamma^\# \vdash^{\prec}_{\Sigma^\#,\mathcal{E}^\#} K^\# \textit{ Kind}$$
$$\Vdash^{\prec}_{\mathcal{E}} \Sigma \textit{ Sig implies } \vdash^{\prec}_{\mathcal{E}^\#} \Sigma^\# \textit{ Sig}$$
$$\Vdash^{\prec}_{\Sigma} \mathcal{E} \textit{ Eq implies } \vdash^{\prec}_{\Sigma^\#} \mathcal{E}^\# \textit{ Eq}$$
$$\Vdash^{\prec}_{\Sigma,\mathcal{E}} \Gamma \textit{ Ctx implies } \vdash^{\prec}_{\Sigma^\#,\mathcal{E}^\#} \Gamma^\# \textit{ Ctx}$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M' \textit{ implies } \Gamma^\# \vdash^{\prec}_{\Sigma^\#,\mathcal{E}^\#} M^\# =_{\mathcal{E}'\beta\eta} (M')^\#$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \textit{ implies } \Gamma^\# \vdash^{\prec}_{\Sigma^\#,\mathcal{E}^\#} A^\# =_{\mathcal{E}'\beta\eta} (A')^\#$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K' \textit{ implies } \Gamma^\# \vdash^{\prec}_{\Sigma^\#,\mathcal{E}^\#} K^\# =_{\mathcal{E}'\beta\eta} (K')^\#$$
$$\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \textit{ implies } \Gamma^\# \vdash^{\prec}_{\Sigma^\#,\mathcal{E}^\#} \theta^\# : \Delta^\#$$

Proof. We argue by (simultaneous) induction on the derivations. The proof is straightforward. □

Proposition 3.2.4. *Let M, A, and K be marked objects, type families, and kinds, respectively, the following holds:*

$$\begin{aligned} M^{\#} \to_\beta N \textit{ implies } \exists M' \textit{ s. t. } M \to_\beta M' \textit{ and } (M')^{\#} = N \\ A^{\#} \to_\beta B \textit{ implies } \exists A' \textit{ s. t. } A \to_\beta A' \textit{ and } (A')^{\#} = B \\ K^{\#} \to_\beta L \textit{ implies } \exists K' \textit{ s. t. } K \to_\beta K' \textit{ and } (K')^{\#} = L \end{aligned}$$

Proof. By induction on the structure of $M^{\#}$, $A^{\#}$, and $K^{\#}$, we lift the β-reduction to marked terms. □

Proposition 3.2.5. *Let M, A, K marked terms, we have*

$$\begin{aligned} M \to_\beta N \textit{ implies } M^{\#} \to_\beta^{=} N^{\#} \\ A \to_\beta B \textit{ implies } A^{\#} \to_\beta^{=} B^{\#} \\ K \to_\beta L \textit{ implies } K^{\#} \to_\beta^{=} L^{\#} \end{aligned}$$

Proof. Immediate. Equality holds if the β-redex lies in one of the marks. □

Notation. Since Propositions 3.2.2, 3.2.3, 3.2.4, and 3.2.5 above allow us to translate immediately all the results we are going to prove in this chapter (Subject Reduction, Strong Normalization, Church-Rosser Property) to the unmarked LF calculus, we will state these results only once, for the marked setting.

3.3 Behavior of β-Reduction

We have now most tools we need to examine the behavior of β-reduction.

Proposition 3.3.1 (β-Subject Reduction). *β-reduction on raw objects preserves well-typedness:*

$$\begin{aligned} \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ and } M \to_\beta M' \textit{ implies } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M' : A \\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ and } A \to_\beta A' \textit{ implies } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A' : K \\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind \textit{ and } K \to_\beta K' \textit{ implies } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K' \; Kind \end{aligned}$$

Proof. By induction on the derivations, we show:

$$\begin{aligned} \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : A \text{ and } M \to_\beta M' \text{ implies } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M =_{\beta\eta} M' \\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : K \text{ and } A \to_\beta A' \text{ implies } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A =_{\beta\eta} A' \\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind \text{ and } K \to_\beta K' \text{ implies } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K =_{\beta\eta} K' \end{aligned}$$

and the result follows from Proposition 3.1.6. □

Theorem 3.3.2 (CR$_\beta$). *β-reduction is Church-Rosser on well-formed terms, i.e.*

$$\begin{aligned} \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ and } M \to_\beta^* M_i \; (i = 1,2) \textit{ implies } \exists N \textit{ s. t. } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} N : A \textit{ and } M_i \to_\beta^* N \; (i = 1,2) \\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ and } A \to_\beta^* A_i \; (i = 1,2) \textit{ implies } \exists B \textit{ s. t. } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} B : K \textit{ and } A_i \to_\beta^* B \; (i = 1,2) \\ \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} K \; Kind \textit{ and } K \to_\beta^* K_i \; (i = 1,2) \textit{ implies } \exists L \textit{ s. t. } \Gamma \Vdash^{\prec}_{\Sigma,\varepsilon} L \; Kind \textit{ and } K_i \to_\beta^* L \; (i = 1,2) \end{aligned}$$

In pictures:

$$\begin{tikzcd} M \arrow[r, "\beta^*"] \arrow[d, "\beta^*"'] & M_1 \arrow[d, dashed, "\beta^*"] \\ M_2 \arrow[r, dashed, "\beta^*"'] & N \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "\beta^*"] \arrow[d, "\beta^*"'] & A_1 \arrow[d, dashed, "\beta^*"] \\ A_2 \arrow[r, dashed, "\beta^*"'] & B \end{tikzcd} \qquad \begin{tikzcd} K \arrow[r, "\beta^*"] \arrow[d, "\beta^*"'] & K_1 \arrow[d, dashed, "\beta^*"] \\ K_2 \arrow[r, dashed, "\beta^*"'] & L \end{tikzcd}$$

Proof. The proof is the usual one, by Martin-Löf and Tait, found, for example, in [7]. $\square$

We are left to prove Strong Normalization of β-reduction (SN_β). We will prove this adapting the technique used in [28] for a version of LF with type conversion modulo β-reduction only.

Definition 3.3.3. We define a map τ from type families and kinds to the set S of simple types constructed from the type constant ω.

$$
\begin{aligned}
\tau(\text{type}) &= \omega \\
\tau(\Pi x : A.\ K) &= \tau(A) \to \tau(K) \\
\tau(a^K) &= \omega \\
\tau((A\ M)^K) &= \tau(A) \\
\tau(\Pi x : A.\ B) &= \tau(A) \to \tau(B)
\end{aligned}
$$

Figure 3.3: The Type Embedding τ

Furthermore, let

$$\mathcal{C} = \{c \mid \Sigma(c) = A\} \cup \{a \mid \Sigma(a) = K\} \cup \{\pi_s \mid s \in S\} \cup \{\text{type}\},$$

we define another map $|\cdot|$ from LF terms to simply-typed objects constructed from the set of constants $\mathcal{C}$.

$$
\begin{aligned}
|c^A| &= (\lambda y : \omega.\ c)\ |A| \\
|x^A| &= (\lambda y : \omega.\ x)\ |A| \quad y \neq x \\
|(\lambda x : A.\ M)^B| &= (\lambda x : \omega.\ (\lambda x : \omega.\ (\lambda x : \tau(A).\ |M|))\ |A|)\ |B| \\
|(M\ N)^A| &= (\lambda x : \omega.\ |M|\ |N|)\ |A| \quad x \notin \mathcal{FV}(M\ N) \\
|a^K| &= (\lambda x : \omega.\ a)\ |K| \\
|\Pi x : A.\ B| &= \pi_{\tau(A)}\ |A|\ (\lambda x : \tau(A).\ |B|) \\
|(A\ M)^K| &= (\lambda x : \omega.\ |A|\ |M|)\ |K| \quad x \notin \mathcal{FV}(A\ M) \\
|\,\text{type}\,| &= \text{type} \\
|\Pi x : A.\ K| &= \pi_{\tau(A)}\ |A|\ (\lambda x : \tau(A).\ |K|)
\end{aligned}
$$

Figure 3.4: The Object Embedding $|\cdot|$

Proposition 3.3.4.

$$
\begin{aligned}
\tau([N/x]A) &= \tau(A) \\
\tau([N/x]K) &= \tau(K)
\end{aligned}
$$

Proof. By an easy induction on A and K. $\square$

Corollary 3.3.5.

$$
\begin{aligned}
&\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \ \textit{implies}\ \tau(A) = \tau(A') \\
&\Gamma \Vdash^{\prec}_{\Sigma,\mathcal{E}} K =_{\mathcal{E}'\beta\eta} K' \ \textit{implies}\ \tau(K) = \tau(K')
\end{aligned}
$$

Proof. By induction on the derivations, using Proposition 3.3.4. $\square$

Proposition 3.3.6.

$$[|N|/x]|M| \to_\beta^* |[N/x]M|$$
$$[|N|/x]|A| \to_\beta^* |[N/x]A|$$
$$[|N|/x]|K| \to_\beta^* |[N/x]K|$$

Proof. By induction on the structure of M, A, and K. We examine a few cases in detail:

- Case $M = x^B$

  By unraveling the definitions

  $$[|N|/x]|M| = [|N|/x]((\lambda y : \omega.\ x)\ |B|) = (\lambda y : \omega.\ |N|)\ ([|N|/x]B) \to_\beta |N| = |[N/x]M|,$$

  where the variable y above has been chosen so to satisfy $y \notin \mathcal{FV}(N)$.

  As we see here, one β-reduction is necessary to "strip" the mark from the occurrence of x that is being replaced by the object N.

- Case $M = (M_1\ M_2)^B$

  By inductive hypothesis we have

  $$[|N|/x]|M_i| \to_\beta^* |[N/x]M_i| \qquad (i = 1, 2)$$
  $$[|N|/x]|B| \to_\beta^* |[N/x]B|,$$

  hence

  $$[|N|/x]|M| = [|N|/x]((\lambda y : \omega.\ |M_1|\ |M_2|)\ |B|) \to_\beta^* (\lambda y.\ |[N/x]M_1|\ |[N/x]M_2|)\ |[N/x]B| = |[N/x]M|,$$

□

We now define a type assignment for the constants in $\mathcal{C}$ and show that, under this assignment, $|.|$ maps well-formed LF objects to well-typed lambda objects.

Definition 3.3.7. Given a signature Σ and a context Γ, we define

$$\tau(\Sigma) = \{c : \tau(A)\ \mid \Sigma(c) = A\} \cup \{a : \tau(K)\ \mid \Sigma(a) = K\} \cup \{\pi_s : \omega \to (s \to \omega) \to \omega\ \mid s \in S\} \cup \{\text{type} : \omega\}$$
$$\tau(\Gamma) = \{x : \tau(A)\ \mid \Gamma(x) = A\}$$

Lemma 3.3.8.

$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} M : A \textit{ implies } \tau(\Gamma) \vdash_{\tau(\Sigma)} |M| : \tau(A) \textit{ and } \tau(\Gamma) \vdash_{\tau(\Sigma)} |A| : \omega$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} A : K \textit{ implies } \tau(\Gamma) \vdash_{\tau(\Sigma)} |A| : \tau(K) \textit{ and } \tau(\Gamma) \vdash_{\tau(\Sigma)} |K| : \omega$$
$$\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} K\ Kind \textit{ implies } \tau(\Gamma) \vdash_{\tau(\Sigma)} |K| : \omega$$

Proof. By induction on the derivations. Subject Reduction of β-reduction for simply-typed objects is used extensively throughout the proof. We show some representative cases:

- Case

  $$\frac{\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} M : \Pi x : A.\ B \qquad \Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} N : A}{\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} (M\ N)^{[N/x]B} : [N/x]B}$$

  By inductive hypothesis

  $$\tau(\Gamma) \vdash_{\tau(\Sigma)} \pi_{\tau(A)}\ |A|\ (\lambda x : \tau(A).\ |B|) : \omega$$
  $$\tau(\Gamma) \vdash_{\tau(\Sigma)} |N| : \tau(A).$$

Hence

$$\tau(\Gamma) \vdash_{\tau(\Sigma)} (\lambda x : \tau(A).\ |B|)\ |N| : \omega$$

and by Proposition 3.3.6 and Subject Reduction

$$\tau(\Gamma) \vdash_{\tau(\Sigma)} |[N/x]B| : \omega$$

Similarly, using the inductive hypothesis, one shows

$$\tau(\Gamma) \vdash_{\tau(\Sigma)} |M| : \tau(A) \to \tau(B)$$

and therefore, using Proposition 3.3.4,

$$\tau(\Gamma) \vdash_{\tau(\Sigma)} (\lambda y : \omega.\ |M|\ |N|)|[N/x]B| : \tau([N/x]B)$$

- Case:

$$\frac{\Gamma \Vdash^{\preceq}_{\Sigma,\mathcal{E}} x^{A'} : A \qquad \Gamma \Vdash^{\preceq}_{\Sigma,\mathcal{E}} A' =_{\mathcal{E}'\beta\eta} A'' \qquad \Gamma \Vdash^{\preceq}_{\Sigma,\mathcal{E}} A'' : \mathrm{type}}{\Gamma \Vdash^{\preceq}_{\Sigma,\mathcal{E}} x^{A''} : A}$$

By inductive hypothesis

$$\begin{aligned} &\tau(\Gamma) \vdash_{\tau(\Sigma)} (\lambda y : \omega.\ x)\ |A'| : \tau(A) \\ &\tau(\Gamma) \vdash_{\tau(\Sigma)} |A| : \omega \\ &\tau(\Gamma) \vdash_{\tau(\Sigma)} |A''| : \omega \end{aligned}$$

By subject reduction,

$$\tau(\Gamma) \vdash_{\tau(\Sigma)} x : \tau(A)$$

and hence

$$\tau(\Gamma) \vdash_{\tau(\Sigma)} (\lambda y : \omega.\ x)\ |A''| : \tau(A)$$

as required.

□

Proposition 3.3.9.

$$\begin{gathered} M \to_\beta M' \ \textit{implies}\ |M| \to^+_\beta |M'| \\ A \to_\beta A' \ \textit{implies}\ |A| \to^+_\beta |A'| \\ K \to_\beta K' \ \textit{implies}\ |K| \to^+_\beta |K'| \end{gathered}$$

Proof. By a simple induction on the structure of M, A, and K. Most cases are immediately dealt with using the inductive hypothesis. The main case is:

$$((\lambda x : A.\ M)^C\ N)^B \to_\beta [N/x]M$$

which is easily proved using Proposition 3.3.6:

$$\begin{aligned} |((\lambda x : A.\ M)^C\ N)^B| &= \lambda y : \omega.\ ((\lambda x : \omega.\ (\lambda x : \omega.\ (\lambda x : \tau(A).\ |M|)\ |A|)\ |C|)\ |N|)\ |B| \\ &\to^+_\beta (\lambda x : \tau(A).\ |M|)\ |N| \\ &\to^+_\beta [|N|/x]|M| \to^*_\beta |[N/x]M| \end{aligned}$$

□

Theorem 3.3.10 (SN$_\beta$). *β-reduction of well-formed marked objects, type families, and kinds is strongly normalizing.*

Proof. Assume, for sake of contradiction, that we have an infinite reduction sequence

$$M = M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta \ldots \to_\beta M_n \to_\beta M_{n+1} \to_\beta \ldots$$

starting from a well-formed object $\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} M : A$.

By Lemma 3.3.8, $|M|$ is a well-typed object in the simply-typed lambda calculus:

$$\tau(\Gamma) \vdash_{\tau(\Sigma)} |M| : \tau(A)$$

and, using Proposition 3.3.9 repeatedly, we can form an infinite β-reduction sequence starting from $|M|$:

$$|M| = |M_0| \to^+_\beta |M_1| \to^+_\beta |M_2| \to^+_\beta \ldots \to^+_\beta |M_n| \to_\beta |M_{n+1}| \to^+_\beta \ldots$$

a contradiction. $\square$

3.4 An Important Term Ordering Lemma

We are now ready to present Theorem 3.4.1, the *raison d'être of* the marked system presented in this chapter. However, before we can proceed, we still need some additional notation.

Notation. Given a derivation of $\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} M : A$ (respectively $\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} A : K$, $\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} K\ Kind$) witnessing the well-formedness of an object M (resp. A, K), this will also give us, for each occurrence of an object N or type family B inside M (resp. A, K), a proof that N and B are well-formed. We will call $\Gamma(M, N)$ and $\Gamma(M, B)$ (resp. $\Gamma(A, N)$ and $\Gamma(A, B)$, $\Gamma(K, N)$ and $\Gamma(K, B)$) the contexts under which N and B are well-formed.

Note that these contexts do not depend on the particular derivation of $\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} M : A$ (resp. $\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} A : K$, $\Gamma \Vdash^{\preceq}_{\Sigma,\varepsilon} K\ Kind$), and they can be easily computed by adding to Γ the variables of the λ-abstractions that enclose N and B.

Also, we will denote by $type(\Gamma(M, N))$ and $kind(\Gamma(M, N))$ (resp. $type(\Gamma(A, N))$ and $kind(\Gamma(A, B))$, $type(\Gamma(K, N))$ and $kind(\Gamma(K, B))$) one of the (pairwise convertible, by Type Uniqueness) types and kinds such that

$$\Gamma(M,N) \Vdash^{\preceq}_{\Sigma,\varepsilon} N : type(\Gamma(M,N)) \quad (\text{resp. } \Gamma(A,N) \Vdash^{\preceq}_{\Sigma,\varepsilon} N : type(\Gamma(A,N)), \Gamma(K,N) \Vdash^{\preceq}_{\Sigma,\varepsilon} N : type(\Gamma(K,N)))$$
$$\Gamma(M,B) \Vdash^{\preceq}_{\Sigma,\varepsilon} B : kind(\Gamma(M,B)) \quad (\text{resp. } \Gamma(A,B) \Vdash^{\preceq}_{\Sigma,\varepsilon} B : kind(\Gamma(A,B)), \Gamma(K,B) \Vdash^{\preceq}_{\Sigma,\varepsilon} B : kind(\Gamma(K,B)))$$

Similar definitions are given for unmarked objects too.

The following result is due to G. Dowek et al. [17]:

Theorem 3.4.1. *Assume $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M : A$ (respectively $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A : K$, $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} K\ Kind$). Then there is a well-founded partial order $<_M$ (resp. $<_A$, $<_K$) defined on the set of LF terms (i.e. objects, type families, and kinds) such that*

1. *on all terms appearing inside M (resp. A, K), $<_M$ (resp. $<_A$, $<_K$) subsumes the strict containment order.*
2. *if $N \le_M M$ and $B \le_M M$ (resp. $N \le_A A$ and $B \le_A A$, $N \le_K K$ and $B \le_K K$), there are β-normal type families C and kinds L satisfying*
   (a) *$C <_M N$ and $L <_M B$ (resp. $C <_A N$ and $L <_A B$, $C <_K N$ and $L <_K B$)*
   (b) *$\Gamma(M,N) \vdash^{\preceq}_{\Sigma,\varepsilon} N : C$ and $\Gamma(M,B) \vdash^{\preceq}_{\Sigma,\varepsilon} B : L$ (resp. $\Gamma(A,N) \vdash^{\preceq}_{\Sigma,\varepsilon} N : C$ and $\Gamma(A,B) \vdash^{\preceq}_{\Sigma,\varepsilon} B : L$, $\Gamma(K,N) \vdash^{\preceq}_{\Sigma,\varepsilon} N : C$ and $\Gamma(K,B) \vdash^{\preceq}_{\Sigma,\varepsilon} B : L$)*
   (c) *$<_C \subseteq <_M$ and $<_L \subseteq <_M$ (resp. $<_C \subseteq <_A$ and $<_L \subseteq <_A$, $<_C \subseteq <_K$ and $<_L \subseteq <_K$)*

Proof. We will prove the statement for $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A$; the other two cases are proved similarly.

From Proposition 3.2.2, there is a derivation $\mathcal{D}^m$ of $\Gamma^m \Vdash^{\prec}_{\Sigma,\mathcal{E}} M^m : A^m$ such that $(\Gamma^m)^{\#} = \Gamma$, $(M^m)^{\#} = M$, (A^m). Using Theorem 3.3.10, we can furthermore assume that all the marks in M^m are β-normal.

We define, for all terms (i.e. objects, type families and kinds) X and Y appearing inside M^m,

$$Y^{\#} <_M X^{\#} \text{ iff } Y \text{ is properly contained inside } X$$

The order $<_M$ is as required: if two unmarked term are contained one inside the other, so are their marked correspondents; moreover each object contains its tag, which is by construction β-normal. □

Remark. Although the usefulness of this result will become clear when we will present Lemma 4.2.3, it is beneficial to the curious reader to give here an intuition of its motivation.

Most of the proofs presented so far argued by induction on the structure of the expressions or the derivations involved. Such an approach becomes ineffective when proving the existence of long-η β-normal forms, as we are now going to show. Since we have proven existence and uniqueness of β-normal forms, we can ease our task by assuming that all the objects and types discussed are β-normal. The problem in constructing η-long normal forms by structural induction on the expressions lies in handling terms of the form

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x_1 : A_1.\ \ldots \Pi x_n : A_n.\ B$$

where M is not a λ. We can η-expand it to

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x_1 : A_1.\ \ldots \lambda x_n : A_n.\ M\ x_1\ \ldots x_n : \Pi x_1 : A_1.\ \ldots \Pi x_n : A_n.\ B$$

However, in a dependently typed calculus like LF that alone does not suffice to obtain a η-long form. In general, the A_i may contain objects that in turn need to be expanded, and unfortunately our induction principle is not strong enough to ensure that they are. Hence we see we need a better form of induction, that allows us to argue not only on the structure of an object but also on that of its type, which is essentially what the well-founded ordering of Theorem 3.4.1 provides.

3.5 Summary

In this chapter we analyzed the properties of β-reduction. Our presentation followed the one used by Harper et al. in [28], but needed to be carried out in a more general system than they used. General not only because of the explicit nature of conversion, and the additional $\mathcal{E}$-conversion rules we added to the calculus, but also because of the "marked" system we define in Section 3.1. While the notation became heavier because of this choice, the introduction of marks did not affect the development of the theory, since, as we proved in Section 3.2, the marked version is equivalent to the unmarked one.

In Section 3.3 we established strong normalization of β by projecting LF onto the simply-typed Lambda Calculus, and outlined how confluence can be proved by the well-known Martin-Löf and Tait technique of defining parallel reductions.

Finally, in Section 3.4 we use existence and uniqueness of β-normal forms, as well as the marked system, to carry a construction, due to Dowek et al., that gives us a more powerful induction principle than structural induction alone.

Chapter 4

Properties of η-Expansion

In this chapter we analyze full $\beta\eta$-conversion. In doing so, our strategy will be to first concentrate on η-conversion alone, and then generalize to both conversion relations, integrating results from the previous chapter.

Unlike β-conversion, which is naturally oriented in the reductionary direction, in orienting η we can go both ways. And, while η-reduction is clearly a strongly normalizing relation, since it reduces the size of the overall expression, η-expansion, endowed with a few simple syntactic restrictions, can also be also shown to be well-founded. Since our interest lies in the study of higher-order rewriting, the latter reduction notion turns out to be the most convenient to use. To explain what makes η-reduction somewhat less palatable to us, we present the following:

Example 2. Consider the following signature representing the unit type and its unique inhabiting object:

$$\mathrm{dom}(\Sigma) = \{\mathbf{unit}, \mathbf{1}\}$$

$$\Sigma(\mathbf{unit}) = \mathrm{type} \qquad\qquad \Sigma(\mathbf{1}) = \mathbf{unit}$$

To express that all the objects of type **unit** must be congruent to **1**, we give the single rewriting rule:

$$x : \mathbf{unit} \vdash^{\prec}_{\Sigma} x \Rightarrow \mathbf{1} : \mathbf{unit}$$

This rule interacts with η-reduction giving rise to a non-confluent system. Let M be an object of type $\Pi x : \mathbf{unit}.\ \mathbf{unit}$, and $y \notin \mathcal{FV}(M)$, we have the following divergent pair:

$$\begin{tikzcd}
 & \lambda y : \mathbf{unit}.\ (M\ y) \arrow[dl, "\eta"'] \arrow[dr, "\mathcal{R}"] & \\
M & & \lambda y : \mathbf{unit}.\ (M\ \mathbf{1})
\end{tikzcd}$$

Unfortunately, until recently all the proofs in the literature [67, 22] used η in its reductionary direction. In this chapter, we adapt a proof presented by N. Ghani in [23] for the Calculus of Construction, that is, to our knowledge, the first to make use of η-expansion.

4.1 Canonical Forms

The main problem in using η-conversion in the expansionary direction is to ensure termination. For this reason, in the simply-typed lambda calculus we allow expansion of an object only if it is not an abstraction and it is not applied to some other object.

In the presence of dependent types, these two restrictions alone no longer suffice, as the following example shows.

Example 3. Let us add to the signature of Example 2 an additional constant $\mathbf{p}$

$$\mathrm{dom}(\Sigma) = \{\mathbf{unit}, \mathbf{1}, \mathbf{p}\}$$

$$\Sigma(\mathbf{unit}) = \text{type} \qquad \Sigma(\mathbf{1}) = \mathbf{unit} \qquad \Sigma(\mathbf{p}) = \Pi x : \mathbf{unit}.\ \text{type}$$

and let

$$A = \Pi y : \mathbf{p}\ \mathbf{1}.\ \mathbf{p}\ \mathbf{1} \qquad B(x) = \mathbf{p}\ ((\lambda y : A.\ \mathbf{1})\ x)$$

we have the following infinite sequence of η-expansions:

$$x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} x \ {}_{\eta}\!\leftarrow \lambda z : B(x).\ (x\ z) \ {}_{\eta}\!\leftarrow \lambda z : B(\lambda z : B(x).\ (x\ z)).\ (x\ z) \ {}_{\eta}\!\leftarrow \ldots$$

To avoid these problems, in defining our η-expansionary rule, we will require, in addition to the two restrictions mentioned before, the type of the expansion to be in long-η β-normal form.

This may seem to introduce circularity in our definitions: we use the η-expansion rule to define long-η β-normal forms. Actually, our approach will be to first give a syntactical characterization of these forms, and use this to define the expansion rule. Later, we will show that the normal forms of this β-reducing/η-expanding rewriting relation are exactly those objects satisfying our syntactical definition.

Long-η β-normal forms in LF are traditionally called *canonical forms*:

Definition 4.1.1. Canonical forms are defined by the judgments

$$\begin{array}{ll} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A & M \text{ is a canonical object of type } A \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} & A \text{ is a canonical type} \end{array}$$

defined in objects of the auxiliary ones

$$\begin{array}{ll} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A & M \text{ is an atomic object of type } A \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \downarrow K & A \text{ is an atomic type of kind K} \end{array}$$

by the inference system shown in Figure 4.1.

We list below a few simple properties of atomic and canonical forms, that will be used frequently in this section. All the proofs are obtained by a straightforward induction on the derivations and are omitted.

Proposition 4.1.2. *Canonical and atomic terms are well-formed:*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \text{type}$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \downarrow K \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K$$

Proposition 4.1.3. *Canonical and atomic terms are β-normal:*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A \textit{ implies } M \textit{ is } \beta\textit{-normal} \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} \textit{ implies } A \textit{ is } \beta\textit{-normal}$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A \textit{ implies } M \textit{ is } \beta\textit{-normal} \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \downarrow K \textit{ implies } A \textit{ is } \beta\textit{-normal}$$

Proposition 4.1.4 (Weakening). *Let $\Gamma \subseteq \Gamma'$, $\Sigma \subseteq \Sigma'$, $\mathcal{E} \subseteq \mathcal{E}'$, the following holds:*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A \textit{ implies } \Gamma' \vdash^{\prec}_{\Sigma',\mathcal{E}'} M \Downarrow A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} \textit{ implies } \Gamma' \vdash^{\prec}_{\Sigma',\mathcal{E}'} A \Downarrow \text{type}$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A \textit{ implies } \Gamma' \vdash^{\prec}_{\Sigma',\mathcal{E}'} M \downarrow A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \downarrow K \textit{ implies } \Gamma' \vdash^{\prec}_{\Sigma',\mathcal{E}'} A \downarrow K$$

Proposition 4.1.5. *Let $\Gamma_1 \vdash^{\prec}_{\Sigma,\mathcal{E}} C =_{\mathcal{E}'\beta\eta} C'$,*

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A \textit{ implies } \Gamma_1, x : C', \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A$$

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} \textit{ implies } \Gamma_1, x : C', \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \text{type}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \downarrow A \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \downarrow \text{type}}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Downarrow A} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} \qquad \Gamma, x : A \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Downarrow B \qquad A \preceq^{\mathrm{t}} B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : A.\ M \Downarrow \Pi x : A.\ B}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Downarrow A \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A' : \text{type}}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Downarrow A'}$$

$$\frac{\Sigma(c) = A}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} c \downarrow A} \qquad \frac{\Gamma(x) = A}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} x \downarrow A} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \downarrow \Pi x : A.\ B \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N \Downarrow A}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M\ N \downarrow [N/x]B}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \downarrow A \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A' : \text{type}}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \downarrow A'}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \downarrow \text{type}}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \Downarrow \text{type}} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} \qquad \Gamma, x : A \vdash^{\preceq}_{\Sigma,\mathcal{E}} B \Downarrow \text{type} \qquad A \preceq^{\mathrm{t}} B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \Pi x : A.\ B \Downarrow \text{type}}$$

$$\frac{\Sigma(a) = K}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} a \downarrow K} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \downarrow \Pi x : B.\ K \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N \Downarrow B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A\ N \downarrow [N/x]K}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \downarrow K' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} K' =_{\zeta} K \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} K\ Kind}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \downarrow K}$$

Figure 4.1: Canonical and Atomic Forms

We use canonical forms to formulate our restricted version of η-expansion:

Definition 4.1.6. We define the judgments

$$\begin{array}{ll} \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta} M' & M\ \eta\text{-expands to } M' \\ \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \Rightarrow_{\eta} A' & A\ \eta\text{-expands to } A' \end{array}$$

using the auxiliary ones

$$\begin{array}{ll} \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \stackrel{\mathcal{I}}{\Rightarrow}_{\eta} M' & M\ \eta\text{-expands internally to } M' \\ \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \stackrel{\mathcal{I}}{\Rightarrow}_{\eta} A' & A\ \eta\text{-expands internally to } A' \end{array}$$

according to the rules of Figure 4.2

Remark. It is not difficult to see that the definition of $\Rightarrow_{\eta}$ given above meets all the requirements we stated at the beginning of this section. In particular, the rule

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} \qquad M \text{ not a } \lambda}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta} \lambda x : A.\ (M\ x)}$$

requires, in order to be applicable, that the term M is not an abstraction, thereby avoiding creating new β-redexes. We also demand the abstracted type to be canonical, ruling out situations like the one presented in Example 3.

Also, we forbid η-expansion of the functional argument of an application by the use of the auxiliary judgment $\stackrel{\mathcal{I}}{\Rightarrow}_{\eta}$. It follows from our definitions that $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \stackrel{\mathcal{I}}{\Rightarrow}_{\eta} M'$ only if M' is obtained from M by

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : \Pi x : A.\ B \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Downarrow \text{type} \qquad M \text{ not a } \lambda}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\eta \lambda x : A.\ (M\ x)} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \stackrel{\mathcal{I}}{\Rightarrow}_\eta M'}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\eta M'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_\eta A' \qquad \Gamma, x : A \vdash^{\prec}_{\Sigma,\varepsilon} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \lambda x : A.\ M \stackrel{\mathcal{I}}{\Rightarrow}_\eta \lambda x : A'.\ M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : \text{type} \qquad \Gamma, x : A \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\eta M'}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \lambda x : A.\ M \stackrel{\mathcal{I}}{\Rightarrow}_\eta \lambda x : A.\ M'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \stackrel{\mathcal{I}}{\Rightarrow}_\eta M' \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} N : B}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M\ N \stackrel{\mathcal{I}}{\Rightarrow}_\eta M'\ N} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : \Pi x : A.\ B \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} N \Rightarrow_\eta N'}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M\ N \stackrel{\mathcal{I}}{\Rightarrow}_\eta M\ N'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \stackrel{\mathcal{I}}{\Rightarrow}_\eta A'}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_\eta A'} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_\eta A' \qquad \Gamma, x : A \vdash^{\prec}_{\Sigma,\varepsilon} B : \text{type}}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \Pi x : A.\ B \stackrel{\mathcal{I}}{\Rightarrow}_\eta \Pi x : A'.\ B} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : \text{type} \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} B \Rightarrow_\eta B'}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \Pi x : A.\ B \stackrel{\mathcal{I}}{\Rightarrow}_\eta \Pi x : A.\ B'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \stackrel{\mathcal{I}}{\Rightarrow}_\eta A' \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A\ M \stackrel{\mathcal{I}}{\Rightarrow}_\eta A'\ M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : \Pi x : B.\ K \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\eta M'}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A\ M \stackrel{\mathcal{I}}{\Rightarrow}_\eta A\ M'}$$

Figure 4.2: Rules for η-Expansion

η-expanding at an internal (i.e. non-topmost) position. We use this fact in the rule

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \stackrel{\mathcal{I}}{\Rightarrow}_\eta M' \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} N : B}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M\ N \stackrel{\mathcal{I}}{\Rightarrow}_\eta M'\ N}$$

to ensure that, again, no new β-redexes are created as a result of an η-expansion.

Notation. For a more uniform treatment of β-reduction and η-expansion, it will be convenient to express the former as a judgment too. For this reason we define

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\beta M' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A \text{ and } M \rightarrow_\beta M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_\beta A' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : K \text{ and } A \rightarrow_\beta A'$$

Furthermore, we define

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\beta\eta} M' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\eta M' \text{ or } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\beta M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\beta\eta} A' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_\eta A' \text{ or } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_\beta A'$$

and similarly

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \stackrel{\mathcal{I}}{\Rightarrow}_{\beta\eta} M' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \stackrel{\mathcal{I}}{\Rightarrow}_\eta M' \text{ or } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\beta M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \stackrel{\mathcal{I}}{\Rightarrow}_{\beta\eta} A' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \stackrel{\mathcal{I}}{\Rightarrow}_\eta A' \text{ or } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_\beta A'$$

Some basic properties of η-expansion are listed below.

Proposition 4.1.7.

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\beta\eta} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M =_{\beta\eta} M' \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\beta\eta} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A =_{\beta\eta} A'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \stackrel{\mathcal{I}}{\Rightarrow}_{\beta\eta} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M =_{\beta\eta} M' \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \stackrel{\mathcal{I}}{\Rightarrow}_{\beta\eta} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A =_{\beta\eta} A'$$

Proposition 4.1.8. *η-expansion preserves β-normality:*

$$
\begin{array}{c}
M \ \beta\text{-normal, and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_\eta M' \text{ implies } M' \ \beta\text{-normal} \\
M \ \beta\text{-normal, and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \overset{\mathcal{I}}{\Rightarrow}_\eta M' \text{ implies } M' \ \beta\text{-normal} \\
A \ \beta\text{-normal, and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_\eta A' \text{ implies } A' \ \beta\text{-normal} \\
A \ \beta\text{-normal, and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \overset{\mathcal{I}}{\Rightarrow}_\eta A' \text{ implies } A' \ \beta\text{-normal}
\end{array}
$$

4.2 Normality of Canonical Forms

The purpose of this section is to show that canonical objects coincide with the set $(\Rightarrow_{\beta\eta})$-normal forms. We will do so by proving that the two sets are included one in each other; the first of these two inclusions is trivial:

Lemma 4.2.1.

$$
\begin{array}{c}
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A \text{ implies } M \text{ is a } (\Rightarrow_{\beta\eta})\text{-normal form} \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \mathrm{type} \text{ implies } A \text{ is a } (\Rightarrow_{\beta\eta})\text{-normal form} \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A \text{ implies } M \text{ is a } (\overset{\mathcal{I}}{\Rightarrow}_{\beta\eta})\text{-normal form} \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \downarrow K \text{ implies } A \text{ is a } (\overset{\mathcal{I}}{\Rightarrow}_{\beta\eta})\text{-normal form}
\end{array}
$$

Proof. We already proved β-normality in Proposition 4.1.3. To prove normality with respect to η-expansion we proceed by induction on the derivations. The only challenge comes from the rule

$$
\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \downarrow \mathrm{type}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A}
$$

By inductive hypothesis we know M is $\overset{\mathcal{I}}{\Rightarrow}_{\beta\eta}$-normal. We are left to show that there is no type $\Pi x : B.\ C$ that satisfies the requirements of the expansion rule:

$$
\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : B.\ C \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B \Downarrow \mathrm{type} \qquad M \text{ not a } \lambda}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_\eta \lambda x : B.\ (M\ x)}
$$

By a simple induction on the $\mathcal{E}\beta\eta$-conversion judgment, we can prove the following result:

$$
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : B.\ C =_{\mathcal{E}\beta\eta} D \ \text{ implies } \ D = \Pi x : B'.\ C'
$$

Using this, we see that the existence of a derivation $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A$ guarantees, by Type Uniqueness, that the expansion rule is not applicable on M. Hence M is $(\Rightarrow_{\beta\eta})$-normal, quod erat demostrandum. ☐

Proving that $(\Rightarrow_{\beta\eta})$-normal forms are canonical requires some preliminary work. It will suffice to show that any term can be, by repeated use of $\Rightarrow_{\beta\eta}$, transformed into a canonical one. This is proved using the machinery developed in the previous chapter, and in particular Theorem 3.4.1.

Proposition 4.2.2. *Assume*

$$
\begin{array}{c}
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \mathrm{type}
\end{array}
$$

then there is an object M' such that

$$
\begin{array}{c}
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow^*_\eta M' \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \Downarrow A
\end{array}
$$

Proof. By induction on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A \Downarrow \mathrm{type}$:

- Case:

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A \downarrow \mathrm{type}}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A \Downarrow \mathrm{type}}$$

  In this case we let $M' = M$, since

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \downarrow A \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A \downarrow \mathrm{type}}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Downarrow A}$$

- Case:

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} B \Downarrow \mathrm{type} \qquad \Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} C \Downarrow \mathrm{type} \qquad B \preceq^{\mathrm{t}} C}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} \Pi x : B.\ C \Downarrow \mathrm{type}}$$

  We can apply the inductive hypothesis to $\Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} x \downarrow B$ obtaining an object N_x such that

$$\Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} x \Rightarrow^{*}_{\eta} N_x$$
$$\Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} N_x \Downarrow B$$

  From this, using Weakening:

$$\Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} M\ N_x \downarrow [N_x/x]C.$$

  Using Proposition 4.1.7 and Corollary 2.2.11, we can show $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} [N_x/x]C =_{\varepsilon\beta\eta} C$; hence

$$\Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} M\ N_x \downarrow C.$$

  We apply the inductive hypothesis once again to get an object N such that

$$\Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} M\ N_x \Rightarrow^{*}_{\eta} N$$
$$\Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} N \Downarrow C$$

  Letting $M' = \lambda x : B.\ N$, we see M' is as required, since

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} B \Downarrow \mathrm{type} \qquad \Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} N \Downarrow C \qquad B \preceq^{\mathrm{t}} C}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} \lambda x : B.\ N \Downarrow \Pi x : B.\ C}$$

  and

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow_{\eta} \lambda x : B.\ (M\ x) \Rightarrow^{*}_{\eta} \lambda x : B.\ (M\ N_x) \Rightarrow^{*}_{\eta} \lambda x : B.\ N$$

  Note that we can η-expand M since $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \downarrow A$, and therefore, in particular, M is not a λ.

□

Lemma 4.2.3.

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M : A \textit{ implies } \exists M_{\Downarrow} \textit{ s. t. } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow^{*}_{\beta\eta} M_{\Downarrow} \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_{\Downarrow} \Downarrow A$$
$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A : \mathrm{type} \textit{ implies } \exists A_{\Downarrow} \textit{ s. t. } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A \Rightarrow^{*}_{\beta\eta} A_{\Downarrow} \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A_{\Downarrow} \Downarrow \mathrm{type}$$

Proof. In the previous chapter, we have shown that β-reduction is strongly normalizing and Church-Rosser. Hence every object and type family reduces to its (unique) β-normal form. Since $\Rightarrow_\beta \subseteq \Rightarrow_{\beta\eta}$, without loss of generality, we will assume M and A in β-normal form.

We will prove the first of the two statements; the proof of the second is similar.

Let $<_M$ the partial order defined by Theorem 3.4.1: we will show, by well-founded induction on $<_M$, that all well-formed objects $N \leq_M M$ and types $B <_M M$ have corresponding canonical objects $N_\Downarrow$ and $B_\Downarrow$.

The only interesting case is with N application object. Since N is β-normal, $N = c\ N_1\ \ldots\ N_m$ or $N = x\ N_1\ \ldots\ N_m$; we will assume, without loss of generality, the latter.

$$\Gamma(M,N) \vdash^{\prec}_{\Sigma,\varepsilon} x\ N_1\ \ldots\ N_m : type(\Gamma(M,N))$$

Since $N_i <_M N$, by inductive hypothesis, there are canonical objects $(N_i)_\Downarrow$ for every $i = 1, \ldots, m$. We can therefore construct a derivation of

$$\Gamma(M,N) \vdash^{\prec}_{\Sigma,\varepsilon} x\ (N_1)_\Downarrow\ \ldots\ (N_m)_\Downarrow \downarrow type(\Gamma(M,N))$$

where

$$\Gamma(M,N) \vdash^{\prec}_{\Sigma,\varepsilon} N \Rightarrow^*_\eta x\ (N_1)_\Downarrow\ \ldots\ (N_m)_\Downarrow$$

By the properties of the partial order $<_M$, there is a β-normal type C such that $C < N$ and

$$\Gamma(M,N) \vdash^{\prec}_{\Sigma,\varepsilon} N : C$$

By inductive hypothesis, there is a type $C_\Downarrow$ such that

$$\begin{gathered}\Gamma(M,N) \vdash^{\prec}_{\Sigma,\varepsilon} C_\Downarrow \Downarrow \mathrm{type} \\ \Gamma(M,N) \vdash^{\prec}_{\Sigma,\varepsilon} C \Rightarrow^*_\eta C_\Downarrow\end{gathered}$$

Hence, by type conversion

$$\Gamma(M,N) \vdash^{\prec}_{\Sigma,\varepsilon} x\ (N_1)_\Downarrow\ \ldots\ (N_m)_\Downarrow \downarrow C_\Downarrow$$

and one application of Proposition 4.2.2 concludes the proof. □

Corollary 4.2.4.

$$\begin{gathered}\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A,\ \textit{and}\ M\ (\Rightarrow_{\beta\eta})\textit{-normal form}\ \ \textit{implies}\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Downarrow A \\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : \mathrm{type},\ \textit{and}\ A\ (\Rightarrow_{\beta\eta})\textit{-normal form}\ \ \textit{implies}\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Downarrow \mathrm{type}\end{gathered}$$

Proof. Assume M as in the statement, by Lemma 4.2.3 there is $M_\Downarrow$ such that

$$\begin{gathered}\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_\Downarrow \Downarrow A \\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow^*_{\beta\eta} M_\Downarrow\end{gathered}$$

By normality of M we conclude $M = M_\Downarrow$. The proof of the second statement is similar. □

4.3 λ-Equivalence

Now we would like to prove that $\Rightarrow_{\beta\eta}$ is Church-Rosser, and ultimately that the canonical objects given by Lemma 4.2.3 are unique. Unfortunately, this is not the case, and it can be shown that Church-Rosser fails even on $\Rightarrow_\eta$ alone.

Consider, for example, the critical pair

$$\begin{tikzcd} & M \arrow[dl, Rightarrow, "\eta"'] \arrow[dr, Rightarrow, "\eta"] & \\ \lambda x : A_1.\ (M\ x) & & \lambda x : A_2.\ (M\ x) \end{tikzcd}$$

Using the definitions and Type Uniqueness, we conclude in this case

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_i \Downarrow \text{type}$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_1 =_{\mathcal{E}\beta\eta} A_2$$

In converting from A_1 to A_2, we may have used the equations in $\mathcal{E}$ several times, and thus it is possible that no common reduct of these two types can be found by using β-reduction and η-expansion alone.

From this example, we see that the best that we can hope to achieve is to prove the Church-Rosser Property modulo a relation $\approx$ that relates objects obtained one from the other by converting the types of the λ-abstracted variables. We will show that this is the case, and prove uniqueness of canonical forms modulo such a relation.

Definition 4.3.1. We define the following judgments:

$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'$	M and M' are $\mathcal{E}'\lambda$-equivalent objects
$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A'$	A and A' are $\mathcal{E}'\lambda$-equivalent type families

formed according to the inference system shown in Figure 4.3.

We list below a few basic properties of these new judgments. The proofs are easily obtained arguing by induction on the derivations.

Proposition 4.3.2.

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A'$$

Proof. The proof is quite straightforward. The only interesting case is

- Case

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A' \qquad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M \approx_{\mathcal{E}'} \lambda x : A'.\ M}$$

  Note that this case is not trivial, since in general $\mathcal{E}' \subsetneq \mathcal{E}$. We use a variant of the Nederpelt counterexample [52]. We show

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M =_{\mathcal{E}'\beta\eta} \lambda x : A'.\ (\lambda x : A.\ M)\ x$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A'.\ (\lambda x : A.\ M)\ x =_{\mathcal{E}'\beta\eta} \lambda x : A'.\ M$$

  and the result follows by transitivity.

□

Proposition 4.3.3 (Weakening). *Let* $\Sigma \subseteq \Sigma'$, $\mathcal{E}'' \subseteq \mathcal{E} \subseteq \mathcal{E}'$, $\mathcal{E}'' \subseteq \mathcal{E}''' \subseteq \mathcal{E}'$, $\Gamma \subseteq \Gamma'$,

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}''} M' \textit{ implies } \Gamma' \vdash^{\prec}_{\Sigma',\mathcal{E}'} M \approx_{\mathcal{E}'''} M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}''} A' \textit{ implies } \Gamma' \vdash^{\prec}_{\Sigma',\mathcal{E}'} A \approx_{\mathcal{E}'''} A'$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \approx_{\mathcal{E}'} M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \approx_{\mathcal{E}'} M''}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M''}$$

$$\frac{(\Delta \vdash^{\prec}_{\Sigma} M_\circ = N_\circ : A_\circ) \in \mathcal{E}' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\beta\eta} \theta M_\circ \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N =_{\beta\eta} \theta N_\circ}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} N}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A' \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M \approx_{\mathcal{E}'} \lambda x : A'.\ M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M \approx_{\mathcal{E}'} \lambda x : A.\ M'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N \approx_{\mathcal{E}'} M'\ N} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \approx_{\mathcal{E}'} N'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N \approx_{\mathcal{E}'} M\ N'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' \approx_{\mathcal{E}'} A} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' \approx_{\mathcal{E}'} A''}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A''}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A' \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} B : \mathrm{type}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B \approx_{\mathcal{E}'} \Pi x : A'.\ B} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} B \approx_{\mathcal{E}'} B'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B \approx_{\mathcal{E}'} \Pi x : A.\ B'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A\ M \approx_{\mathcal{E}'} A'\ M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \Pi x : B.\ K \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A\ M \approx_{\mathcal{E}'} A\ M'}$$

Figure 4.3: Rules for $\mathcal{E}'\lambda$-Equivalence

Proposition 4.3.4 (Substitution). *Let* $\Gamma_1 \vdash^{\prec}_{\Sigma,\mathcal{E}} N : C$,

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]M \approx_{\mathcal{E}'} [N/x]M'$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]A \approx_{\mathcal{E}'} [N/x]A'$$

Proposition 4.3.5. *Let* $\Gamma_1 \vdash^{\prec}_{\Sigma,\mathcal{E}} C =_{\mathcal{E}'\beta\eta} C' : \mathrm{type}$,

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M' \textit{ implies } \Gamma_1, x : C', \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A' \textit{ implies } \Gamma_1, x : C', \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A'$$

Proposition 4.3.6. *Let* $\Gamma_1 \vdash^{\prec}_{\Sigma,\mathcal{E}} N : C$ *and* $\Gamma_1 \vdash^{\prec}_{\Sigma,\mathcal{E}} N \approx_{\mathcal{E}'} N'$

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]M \approx_{\mathcal{E}'} [N'/x]M$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]A \approx_{\mathcal{E}'} [N'/x]A$$

4.4 Confluence Modulo $\approx$

Finally, in this section we will show that $\Rightarrow_{\beta\eta}$ is confluent modulo this newly introduced equivalence relation. In order to do this, however, we have first to relativize all our reduction notions to $\approx_{\mathcal{E}'}$:

Definition 4.4.1. We define β-reduction and η-expansion modulo $\mathcal{E}'\lambda$-equivalence by means of the judgments

$$\begin{array}{ll}
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M' & M\ \beta\text{-reduces to } M' \text{ modulo } \mathcal{E}'\lambda\text{-equivalence} \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\beta/\approx_{\mathcal{E}'}} A' & A\ \beta\text{-reduces to } A' \text{ modulo } \mathcal{E}'\lambda\text{-equivalence} \\
\\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta/\approx_{\mathcal{E}'}} M' & M\ \eta\text{-expands to } M' \text{ modulo } \mathcal{E}'\lambda\text{-equivalence} \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\eta/\approx_{\mathcal{E}'}} A' & A\ \eta\text{-expands to } A' \text{ modulo } \mathcal{E}'\lambda\text{-equivalence} \\
\\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \stackrel{\mathcal{I}}{\Rightarrow}_{\eta/\approx_{\mathcal{E}'}} M' & M\ \eta\text{-expands internally to } M' \text{ modulo } \mathcal{E}'\lambda\text{-equivalence} \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \stackrel{\mathcal{I}}{\Rightarrow}_{\eta/\approx_{\mathcal{E}'}} A' & A\ \eta\text{-expands internally to } A' \text{ modulo } \mathcal{E}'\lambda\text{-equivalence}
\end{array}$$

defined as

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M' \text{ iff } \exists N, N' \text{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} N \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \Rightarrow_{\beta} N' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N' \approx_{\mathcal{E}'} M' \end{cases} \text{ and not } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\beta/\approx_{\mathcal{E}'}} A' \text{ iff } \exists B, B' \text{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} B \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B \Rightarrow_{\beta} B' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B' \approx_{\mathcal{E}'} A' \end{cases} \text{ and not } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A'$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta/\approx_{\mathcal{E}'}} M' \text{ iff } \exists N, N' \text{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} N \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \Rightarrow_{\eta} N' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N' \approx_{\mathcal{E}'} M' \end{cases} \text{ and not } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\eta/\approx_{\mathcal{E}'}} A' \text{ iff } \exists B, B' \text{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} B \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B \Rightarrow_{\eta} B' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B' \approx_{\mathcal{E}'} A' \end{cases} \text{ and not } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A'$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \stackrel{\mathcal{I}}{\Rightarrow}_{\eta/\approx_{\mathcal{E}'}} M' \text{ iff } \exists N, N' \text{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} N \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \stackrel{\mathcal{I}}{\Rightarrow}_{\eta} N' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N' \approx_{\mathcal{E}'} M' \end{cases} \text{ and not } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \stackrel{\mathcal{I}}{\Rightarrow}_{\eta/\approx_{\mathcal{E}'}} A' \text{ iff } \exists B, B' \text{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} B \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B \stackrel{\mathcal{I}}{\Rightarrow}_{\eta} B' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B' \approx_{\mathcal{E}'} A' \end{cases} \text{ and not } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A'$$

Moreover, we let

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M' \text{ or } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta/\approx_{\mathcal{E}'}} M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\beta/\approx_{\mathcal{E}'}} A' \text{ or } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\eta/\approx_{\mathcal{E}'}} A'$$

Finally, we define the judgments

$$\begin{array}{ll}
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta/\approx_{\mathcal{E}'}} M' & M \text{ is } (\Rightarrow_{\beta\eta/\approx_{\mathcal{E}'}})\text{-convertible to } M' \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Leftrightarrow_{\beta/\approx_{\mathcal{E}'}} A' & A \text{ is } (\Rightarrow_{\beta\eta/\approx_{\mathcal{E}'}})\text{-convertible to } A'
\end{array}$$

constructed from the rules

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M'} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma} M \Rightarrow_{\beta\approx_{\mathcal{E}'}} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M'} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta/\approx_{\mathcal{E}'}} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M''}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M''}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A'} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\beta/\approx_{\mathcal{E}'}} A'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A'} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\eta/\approx_{\mathcal{E}'}} A'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A''}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A''}$$

Not surprisingly, $\mathcal{E}'\beta\eta$-conversion coincides with $\Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}}$, as the following sequence of results will show.

Proposition 4.4.2. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B$. *Then*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx} \lambda x : A.\ (M\ x)$$

Proof. We distinguish between two mutually disjoint cases:

- M is not a λ

  By Type Consistency and inversion, $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \text{type}$, and by Lemma 4.2.3, there is a type $A_{\Downarrow}$ such that

  $$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_{\Downarrow} \Downarrow \text{type}$$
  $$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow^{*}_{\beta\eta} A_{\Downarrow}$$

  Using Proposition 4.1.7 and Corollary 2.2.4, by type conversion we show $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A_{\Downarrow}.\ B$; hence

  $$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta} \lambda x : A_{\Downarrow}.\ (M\ x)$$

  Also, notice that

  $$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ (M\ x) \approx \lambda x : A_{\Downarrow}.\ (M\ x)$$

  and, consequently,

  $$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta/\approx} \lambda x : A.\ (M\ x).$$

- $M = \lambda x : A'.\ M'$

  By inversion, from the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B$ we can extract a derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A'$. Hence

  $$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ (M\ x) \approx \lambda x : A'.\ (M\ x)$$

  and the result follows since $\lambda x : A.\ (M\ x) \rightarrow_{\beta} M$.

□

Proposition 4.4.3.

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N \Rightarrow_{\eta} M'\ N \textit{ or } M'\ N \rightarrow_{\beta} M\ N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\eta} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A\ M \Rightarrow_{\eta} A'\ M \textit{ or } A'\ M \rightarrow_{\beta} A\ M$$

Proof. We will prove the first of the two statements; the proof of the second one is similar.

If we can show $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \overset{\mathcal{I}}{\Rightarrow}_{\eta} M'$ we are done, since we have

$$\cfrac{\cfrac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \overset{\mathcal{I}}{\Rightarrow}_{\eta} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N \overset{\mathcal{I}}{\Rightarrow}_{\eta} M'\ N}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N \Rightarrow_{\eta} M'\ N}$$

The only case in which $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \overset{\mathcal{I}}{\Rightarrow}_{\eta} M'$ does not hold is when the last rule in the derivation is the restricted η-expansion:

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B \quad M \text{ not a } \lambda}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta} \lambda x : A.\ (M\ x)}$$

In this case we see $M' = \lambda x : A.\ (M\ x)$, and therefore $M'\ N \rightarrow_{\beta} M\ N$, which concludes the proof. □

Theorem 4.4.4. *Let M, M' and A, A' well-formed terms, then*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}'\beta\eta} A' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A'$$

Proof. Both directions are proved by induction on the derivations. One is trivial; in the other there are a few cases which deserve some attention:

- Case

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} \lambda x : A.\ (M\ x)}$$

  From Proposition 4.4.2 we have immediately $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} \lambda x : A.\ (M\ x)$.

- Case

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M' \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N =_{\mathcal{E}'\beta\eta} M'\ N}$$

  By inductive hypothesis, we have a derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M'$. We recursively transform this into a derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M'\ N$. The only non-trivial step is when we encounter an application of the rule

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} P \Rightarrow_{\eta/\approx_{\mathcal{E}'}} P'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} P \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} P'}$$

  In this case, to obtain a derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} P\ N \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} P'\ N$ we apply Proposition 4.4.3.

□

We now investigate the properties of β-reduction and η-expansion modulo $\mathcal{E}'\lambda$-equivalence. For both, $\approx_{\mathcal{E}'}$-postponement is easily shown:

Lemma 4.4.5. *Let M and A be well-formed, the following holds:*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M_1, \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\beta} M_2 \text{ implies } \exists N \text{ s. t. } \Gamma \vdash^{\prec} M_2 \approx_{\mathcal{E}'} N \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 \Rightarrow^{=}_{\beta} N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A_1, \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\beta} A_2 \text{ implies } \exists B \text{ s. t. } \Gamma \vdash^{\prec} A_2 \approx_{\mathcal{E}'} B \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_1 \Rightarrow^{=}_{\beta} B$$

Moreover, $M_1 = N$ iff $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M_2$, and $A_1 = B$ iff $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}'} A_2$.

Proof. By induction on the derivations, using Propositions 4.3.4 and 4.3.6.

This is the place where we use the assumption that the equations in $\mathcal{E}$ are all of base type, as we rule out situations where

$$M = (\lambda x : C.\ M')\ P \to_{\beta} [P/x]M' = M_2,$$

and $M_1 = M_1'\ P$, with

$$\frac{(\Delta \vdash^{\prec}_{\Sigma} M_\circ = N_\circ : A_\circ) \in \mathcal{E}' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : C.\ M') =_{\beta\eta} \theta M_\circ \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1' =_{\beta\eta} \theta N_\circ}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : C.\ M') \approx_{\mathcal{E}'} M_1'}$$

for which the statement may not hold. □

Corollary 4.4.6. *For all well-formed objects M and type families A, the following holds:*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\beta} N \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} N \approx_{\mathcal{E}'} M' \textit{ for some } N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\beta/\approx_{\mathcal{E}'}} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\beta} B \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} B \approx_{\mathcal{E}'} A' \textit{ for some } B$$

In pictures:

$$\begin{tikzcd}
M \arrow[rr, Rightarrow, "\beta/\approx_{\mathcal{E}'}"] \arrow[dr, Rightarrow, dashed, "\beta"'] & & M' \\
& N \arrow[ur, dash, dashed, "\approx_{\mathcal{E}'}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[rr, Rightarrow, "\beta/\approx_{\mathcal{E}'}"] \arrow[dr, Rightarrow, dashed, "\beta"'] & & A' \\
& B \arrow[ur, dash, dashed, "\approx_{\mathcal{E}'}"'] &
\end{tikzcd}$$

Proof. Immediate by the definitions and Lemma 4.4.5. □

Lemma 4.4.7. *We have*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \approx_{\mathcal{E}'} M_1, \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\eta} M_2 \textit{ implies } \exists N \textit{ s. t. } \Gamma \vdash^{\prec} M_2 \approx_{\mathcal{E}'} N \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1 \Rightarrow^{=}_{\eta} N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \approx_{\mathcal{E}'} A_1, \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\eta} A_2 \textit{ implies } \exists B \textit{ s. t. } \Gamma \vdash^{\prec} A_2 \approx_{\mathcal{E}'} B \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_1 \Rightarrow^{=}_{\eta} B$$

Moreover, $M_1 = N$ iff $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \approx_{\mathcal{E}'} M_2$, and $A_1 = B$ iff $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \approx_{\mathcal{E}'} A_2$.

Proof. By induction on the derivations, similar to the proof of Lemma 4.4.5. □

Corollary 4.4.8. *The following holds:*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\eta/\approx_{\mathcal{E}'}} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\eta} N \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} N \approx_{\mathcal{E}'} M' \textit{ for some } N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\eta/\approx_{\mathcal{E}'}} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\eta} B \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} B \approx_{\mathcal{E}'} A' \textit{ for some } B$$

Graphically:

$$\begin{tikzcd}
M \arrow[rr, Rightarrow, "\eta/\approx_{\mathcal{E}'}"] \arrow[dr, Rightarrow, dashed, "\eta"'] & & M' \\
& N \arrow[ur, dash, dashed, "\approx_{\mathcal{E}'}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[rr, Rightarrow, "\eta/\approx_{\mathcal{E}'}"] \arrow[dr, Rightarrow, dashed, "\eta"'] & & A' \\
& B \arrow[ur, dash, dashed, "\approx_{\mathcal{E}'}"'] &
\end{tikzcd}$$

Proof. Immediate by Lemma 4.4.7. □

Corollary 4.4.6 allows us to lift immediately most of the results proved in the previous chapter for $\rightarrow_{\beta}$ to $\Rightarrow_{\beta/\approx_{\mathcal{E}'}}$:

Lemma 4.4.9. *The relation $\Rightarrow_{\beta/\approx_{\mathcal{E}'}}$ is Church-Rosser on well-formed objects and type families, i.e.*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M_i \ (i = 1, 2) \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_i \Rightarrow_{\beta/\approx_{\mathcal{E}'}} N \textit{ for some } N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\beta/\approx_{\mathcal{E}'}} A_i \ (i = 1, 2) \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_i \Rightarrow_{\beta/\approx_{\mathcal{E}'}} B \textit{ for some } B$$

where $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : C$ and $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : K$ for some suitable C and K.

In pictures, the following diagrams hold:

$$\begin{tikzcd}
M \arrow[r, Rightarrow, "(\beta/\approx_{\mathcal{E}'})^*"] \arrow[d, Rightarrow, "(\beta/\approx_{\mathcal{E}'})^*"'] & M_1 \arrow[d, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^*"] \\
M_2 \arrow[r, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^*"'] & N
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, Rightarrow, "(\beta/\approx_{\mathcal{E}'})^*"] \arrow[d, Rightarrow, "(\beta/\approx_{\mathcal{E}'})^*"'] & A_1 \arrow[d, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^*"] \\
A_2 \arrow[r, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^*"'] & B
\end{tikzcd}$$

Proof. The result is easily obtained from Theorem 3.3.2 and Corollary 4.4.6 by diagram chasing:

$$\begin{tikzcd}
M \arrow[rrr, Rightarrow, "(\beta/\approx_{\mathcal{E}'})^*"] \arrow[ddd, Rightarrow, "(\beta/\approx_{\mathcal{E}'})^*"'] \arrow[drr, Rightarrow, dashed, "\beta^*"] \arrow[ddr, Rightarrow, dashed, "\beta^*"'] & & & M_1 \arrow[dl, dash, dashed, "\approx_{\mathcal{E}'}"'] \arrow[ddd, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^*"] \\
& & M_1' \arrow[ddr, Rightarrow, dashed, "\beta^*"] & \\
& M_2' \arrow[dl, dash, dashed, "\approx_{\mathcal{E}'}"] \arrow[drr, Rightarrow, dashed, "\beta^*"'] & & \\
M_2 \arrow[rrr, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^*"'] & & & N
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[rrr, Rightarrow, "(\beta/\approx_{\mathcal{E}'})^*"] \arrow[ddd, Rightarrow, "(\beta/\approx_{\mathcal{E}'})^*"'] \arrow[drr, Rightarrow, dashed, "\beta^*"] \arrow[ddr, Rightarrow, dashed, "\beta^*"'] & & & A_1 \arrow[dl, dash, dashed, "\approx_{\mathcal{E}'}"'] \arrow[ddd, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^*"] \\
& & A_1' \arrow[ddr, Rightarrow, dashed, "\beta^*"] & \\
& A_2' \arrow[dl, dash, dashed, "\approx_{\mathcal{E}'}"] \arrow[drr, Rightarrow, dashed, "\beta^*"'] & & \\
A_2 \arrow[rrr, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^*"'] & & & B
\end{tikzcd}$$

□

Lemma 4.4.10. *The relation $\Rightarrow_{\beta/\approx_{\mathcal{E}'}}$ is strongly normalizing on well-formed objects and type families.*

Proof. Assume $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A$ and an infinite reduction sequence starting from $M = M_0$:

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_0 \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M_1 \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M_1 \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M_1 \Rightarrow_{\beta/\approx_{\mathcal{E}'}} \ldots \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M_n \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M_{n+1} \Rightarrow_{\beta/\approx_{\mathcal{E}'}} \cdots$$

Using repeatedly Corollary 4.4.6 we construct an infinite β-reduction sequence

$$M = M_0' \rightarrow_\beta M_1' \rightarrow_\beta \ldots \rightarrow_\beta M_n' \rightarrow_\beta M_{n+1}' \rightarrow_\beta \ldots,$$

a contradiction to Theorem 3.3.10. □

When considering terms modulo λ-equivalence, η-expansions can be easily shown to be Church-Rosser:

Lemma 4.4.11.

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_\eta M_i \ (i = 1,2) \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i \Rightarrow_\eta^= N_i \textit{ for some } N_i \ (i = 1,2) \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 \approx N_2$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \stackrel{\mathcal{I}}{\Rightarrow}_\eta M_i \ (i = 1,2) \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i \stackrel{\mathcal{I}}{\Rightarrow}_\eta^= N_i \textit{ for some } N_i \ (i = 1,2) \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 \approx N_2$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_\eta A_i \ (i = 1,2) \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_i \Rightarrow_\eta^= B_i \textit{ for some } B_i \ (i = 1,2) \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B_1 \approx B_2$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \stackrel{\mathcal{I}}{\Rightarrow}_\eta A_i \ (i = 1,2) \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_i \stackrel{\mathcal{I}}{\Rightarrow}_\eta^= B_i \textit{ for some } B_i \ (i = 1,2) \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_1 \approx A_2$$

Proof. By induction on the derivations, keeping track of where the η-expansions take place.
The main case is

$$\Gamma \vdash^{\prec}_{\Sigma} M \Rightarrow_\eta \lambda x : A_i.\, (M\ x) \ (i = 1,2)$$

where, by Uniqueness of Types, we have $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_1 =_{\mathcal{E}\beta\eta} A_2$, and therefore

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A_1.\, (M\ x) \approx \lambda x : A_2.\, (M\ x)$$

□

Corollary 4.4.12. *Restricted η-expansion modulo $\mathcal{E}'\lambda$-equivalence is Church-Rosser:*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\eta/\approx_{\mathcal{E}'}} M_i \ (i = 1,2) \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i \Rightarrow_{\eta/\approx_{\mathcal{E}'}} N \ (i = 1,2) \textit{ for some } N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Rightarrow_{\eta/\approx_{\mathcal{E}'}} A_i \ (i = 1,2) \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A_i \Rightarrow_{\eta/\approx_{\mathcal{E}'}} B \ (i = 1,2) \textit{ for some } B$$

In pictures:

$$\begin{tikzcd} M \arrow[r, Rightarrow, "(\eta/\approx_{\varepsilon'})^*"] \arrow[d, Rightarrow, "(\eta/\approx_{\varepsilon'})^*"'] & M_1 \arrow[d, Rightarrow, dashed, "(\eta/\approx_{\varepsilon'})^*"] \\ M_2 \arrow[r, Rightarrow, dashed, "(\eta/\approx_{\varepsilon'})^*"'] & N \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, Rightarrow, "(\eta/\approx_{\varepsilon'})^*"] \arrow[d, Rightarrow, "(\eta/\approx_{\varepsilon'})^*"'] & A_1 \arrow[d, Rightarrow, dashed, "(\eta/\approx_{\varepsilon'})^*"] \\ A_2 \arrow[r, Rightarrow, dashed, "(\eta/\approx_{\varepsilon'})^*"'] & B \end{tikzcd}$$

Proof. We first show the following, using Lemma 4.4.11:

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\eta/\approx_{\varepsilon'}} M_i \ (i=1,2) \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_i \Rightarrow^{=}_{\eta/\approx_{\varepsilon'}} N \ (i=1,2) \text{ for some } N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\eta/\approx_{\varepsilon'}} A_i \ (i=1,2) \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_i \Rightarrow^{=}_{\eta/\approx_{\varepsilon'}} B \ (i=1,2) \text{ for some } B$$

that is, in pictures:

$$\begin{tikzcd} M \arrow[r, Rightarrow, "\eta/\approx_{\varepsilon'}"] \arrow[d, Rightarrow, "\eta/\approx_{\varepsilon'}"'] & M_1 \arrow[d, Rightarrow, dashed, "(\eta/\approx_{\varepsilon'})^="] \\ M_2 \arrow[r, Rightarrow, dashed, "(\eta/\approx_{\varepsilon'})^="'] & N \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, Rightarrow, "\eta/\approx_{\varepsilon'}"] \arrow[d, Rightarrow, "\eta/\approx_{\varepsilon'}"'] & A_1 \arrow[d, Rightarrow, dashed, "(\eta/\approx_{\varepsilon'})^="] \\ A_2 \arrow[r, Rightarrow, dashed, "(\eta/\approx_{\varepsilon'})^="'] & B \end{tikzcd}$$

Then, using the above, we prove the statement by induction on the length of the expansion sequences. □

In order to prove the Church-Rosser Property for $\Rightarrow_{\beta\eta/\approx_{\varepsilon'}}$ we will make use of the following:

Lemma 4.4.13. *Let $\to_R$ and $\to_S$ two relations that are both Church-Rosser. Assume, moreover, that $\to_R$ is strongly normalizing and that the following diagram holds:*

$$\begin{tikzcd} X \arrow[r, "R"] \arrow[d, "S"'] & X_1 \arrow[d, dashed, "S^*"] \\ X_2 \arrow[r, dashed, "R^+"'] & Y \end{tikzcd}$$

Then the relation

$$\to_{RS} = \to_R \cup \to_S$$

is Church-Rosser:

$$\begin{tikzcd} X \arrow[r, "(RS)^*"] \arrow[d, "(RS)^*"'] & X_1 \arrow[d, dashed, "(RS)^*"] \\ X_2 \arrow[r, dashed, "(RS)^*"'] & Y \end{tikzcd}$$

Proof. This fact can be proved in the very general setting of the Combinatory Reduction Systems (CRSs) defined by Klop in [36]. A proof is given by R. Di Cosmo in [12]. □

We want to instantiate R with $\Rightarrow_{\beta/\approx_{\varepsilon'}}$ and S with $\Rightarrow_{\eta/\approx_{\varepsilon'}}$ above. In order to do so, however, one final result, i.e. the commutativity of the diagram above, is left to prove.

Proposition 4.4.14. *If* $\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} N : C$ *then*

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_\eta M' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M \Rightarrow_{\eta/\approx} [N/x]M'$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M \overset{\mathcal{I}}{\Rightarrow}_\eta M' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M \overset{\mathcal{I}}{\Rightarrow}_{\eta/\approx} [N/x]M'$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_\eta A' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]A \Rightarrow_{\eta/\approx} [N/x]A'$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} A \overset{\mathcal{I}}{\Rightarrow}_\eta A' \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]A \overset{\mathcal{I}}{\Rightarrow}_{\eta/\approx} [N/x]A'$$

Proof. By simultaneous induction on the derivations. Non-internal expansions are using Lemma 4.2.3. □

Proposition 4.4.15. *Let* $\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} N : C$ *and* $\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} N \overset{\mathcal{I}}{\Rightarrow}_\eta N'$*, then*

$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta [N'/x]M$$
$$\Gamma_1, x : C, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ implies } \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]A \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta [N'/x]A$$

Proof. By a straightforward induction on the derivations. □

Proposition 4.4.16. *Let*

$$\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} N : \Pi z : A.\ B$$
$$\Gamma_1 \vdash^{\prec}_{\Sigma,\varepsilon} \Pi z : A.\ B \Downarrow \text{type},$$

then

$$\Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M : C \textit{ implies } \exists P \textit{ s. t. } \begin{cases} [(\lambda z : A.\ (N\ z))/x]M \rightarrow^*_\beta P \\ \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta P \end{cases}$$
$$\Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} C : K \textit{ implies } \exists D \textit{ s. t. } \begin{cases} [(\lambda z : A.\ (N\ z))/x]C \rightarrow^*_\beta D \\ \Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]C \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta D \end{cases}$$

Proof. By induction on the derivations of $\Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M : C$ and $\Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} C : K$. We show a few interesting cases:

- Case

$$\frac{\Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} x : \Pi z : A'.\ B' \qquad \Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} M' : B'}{\Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} x\ M' : [M'/x]B'}$$

  By inductive hypothesis we have an object P' such that

$$[(\lambda z : A.\ (N\ z))/x]M' \rightarrow^*_\beta P'$$
$$\Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M' \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta P'$$

  We see then $P = N\ P'$ is as required since

$$[(\lambda z : A.\ (N\ z))/x]M = (\lambda z : A.\ (N\ z))\ M' \rightarrow_\beta N\ M' \rightarrow^*_\beta N\ P'$$

  and

$$\Gamma_1, [N/x]\Gamma_2 \vdash^{\prec}_{\Sigma,\varepsilon} [N/x]M \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta N\ P'$$

- Case

$$\frac{\Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\preceq}_{\Sigma,\varepsilon} A' : \text{type} \qquad \Gamma_1, x : \Pi z : A.\ B, \Gamma_2, y : A' \vdash^{\preceq}_{\Sigma,\varepsilon} M' : B' \qquad A' \preceq^{\mathrm{t}} B'}{\Gamma_1, x : \Pi z : A.\ B, \Gamma_2 \vdash^{\preceq}_{\Sigma,\varepsilon} \lambda y : A'.\ M' : \Pi y : A'.\ B'}$$

  Applying the inductive hypothesis twice we get D' and P' such that

$$\begin{array}{l} [(\lambda z : A.\ (N\ z))/x]A' \to^*_\beta D' \\ [(\lambda z : A.\ (N\ z))/x]M' \to^*_\beta P' \\ \Gamma_1, [N/x]\Gamma_2 \vdash^{\preceq}_{\Sigma,\varepsilon} [N/x]A' \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta D' \\ \Gamma_1, [N/x]\Gamma_2, y : [N/x]A' \vdash^{\preceq}_{\Sigma,\varepsilon} [N/x]M' \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta P' \end{array}$$

  We observe that

$$\begin{array}{l} [(\lambda z : A.\ (N\ z))/x](\lambda y : A.\ M') \to^*_\beta \lambda y : D'.\ [(\lambda z : A.\ (N\ z))/x]M' \to^*_\beta \lambda y : D'.\ P' \\ \Gamma_1, [N/x]\Gamma_2 \vdash^{\preceq}_{\Sigma,\varepsilon} [N/x]M \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta \lambda y : [N/x]A'.\ P' \overset{\mathcal{I}}{\Rightarrow}{}^*_\eta \lambda y : D'.\ P' \end{array}$$

□

Lemma 4.4.17. *Let M and A well-formed objects and type families, respectively; then*

$$\begin{array}{c} \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow_\eta M_1 \textit{ and } M \to_\beta M_2 \textit{ implies } \exists N \textit{ s. t. } M_1 \to^+_\beta N \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_2 \Rightarrow^*_{\eta/\approx} N \\ \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \overset{\mathcal{I}}{\Rightarrow}_\eta M_1 \textit{ and } M \to_\beta M_2 \textit{ implies } \exists N \textit{ s. t. } M_1 \to^+_\beta N \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_2 \overset{\mathcal{I}}{\Rightarrow}{}^*_{\eta/\approx} N \\ \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A \Rightarrow_\eta A_1 \textit{ and } A \to_\beta A_2 \textit{ implies } \exists B \textit{ s. t. } A_1 \to^+_\beta B \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A_2 \Rightarrow^*_{\eta/\approx} B \\ \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A \overset{\mathcal{I}}{\Rightarrow}_\eta A_1 \textit{ and } A \to_\beta A_2 \textit{ implies } \exists B \textit{ s. t. } A_1 \to^+_\beta B \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A_2 \overset{\mathcal{I}}{\Rightarrow}{}^*_{\eta/\approx} B \end{array}$$

Proof. By simultaneous induction on the derivations. The two main cases are:

- Case:

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} \lambda x : C.\ M \overset{\mathcal{I}}{\Rightarrow}_\eta \lambda x : C.\ M' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} N : B}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} (\lambda x : C.\ M)\ N \overset{\mathcal{I}}{\Rightarrow}_\eta (\lambda x : C.\ M')\ N}$$

  By inversion, we get a derivation of $\Gamma, x : C \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow_\eta M'$, and the result follows from Proposition 4.4.14.

- Case:

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} \lambda x : C.\ M : \Pi x : C'.\ D \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} N \Rightarrow_\eta N'}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} (\lambda x : C.\ M)\ N \overset{\mathcal{I}}{\Rightarrow}_\eta (\lambda x : C.M)\ N'}$$

  We inspect the last rule used in the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} N \Rightarrow_\eta N'$. If it is the restricted η-expansion

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} N : \Pi x : A.\ B \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A \Downarrow \text{type} \qquad N \text{ not a } \lambda}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} N \Rightarrow_\eta \lambda x : A.\ (N\ x)}$$

  we use Proposition 4.4.15. Otherwise $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} N \overset{\mathcal{I}}{\Rightarrow}_\eta N'$ holds, and the result follows from Proposition 4.4.16.

□

Corollary 4.4.18. *Let M and A well-formed, the following holds:*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\beta/\approx_{\mathcal{E}'}} M_1 \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\eta/\approx_{\mathcal{E}'}} M_2 \textit{ implies } \exists N \textit{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1 \Rightarrow^{*}_{\eta/\approx_{\mathcal{E}'}} N \\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_2 \Rightarrow^{+}_{\beta/\approx_{\mathcal{E}'}} N \end{cases}$$

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\beta/\approx_{\mathcal{E}'}} A_1 \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\eta/\approx_{\mathcal{E}'}} A_2 \textit{ implies } \exists B \textit{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_1 \Rightarrow^{*}_{\eta/\approx_{\mathcal{E}'}} B \\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_2 \Rightarrow^{+}_{\beta/\approx_{\mathcal{E}'}} B \end{cases}$$

In pictures, the following diagrams hold:

$$\begin{tikzcd} M \arrow[r, Rightarrow, "\beta/\approx_{\mathcal{E}'}"] \arrow[d, Rightarrow, "\eta/\approx_{\mathcal{E}'}"'] & M_1 \arrow[d, dashed, "(\eta/\approx_{\mathcal{E}'})^*"] \\ M_2 \arrow[r, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^+"'] & N \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, Rightarrow, "\beta/\approx_{\mathcal{E}'}"] \arrow[d, Rightarrow, "\eta/\approx_{\mathcal{E}'}"'] & A_1 \arrow[d, Rightarrow, dashed, "(\eta/\approx_{\mathcal{E}'})^*"] \\ A_2 \arrow[r, Rightarrow, dashed, "(\beta/\approx_{\mathcal{E}'})^+"'] & B \end{tikzcd}$$

Proof. It follows immediately from Lemmas 4.4.17, 4.4.5, and 4.4.7. □

We have now all the tools needed to prove the most important results of this chapter:

Theorem 4.4.19. *On well-formed terms, the relation $\Rightarrow_{\beta\eta}$ is Church-Rosser modulo λ-equivalence, i.e.*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow^{*}_{\beta\eta} M_i \ (i=1,2) \textit{ implies } \exists N_1, N_2 \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_i \Rightarrow^{*}_{\beta\eta} N_i \ (i=1,2) \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} N_1 \approx N_2$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow^{*}_{\beta\eta} A_i \ (i=1,2) \textit{ implies } \exists B_1, B_2 \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_i \Rightarrow^{*}_{\beta\eta} B_i \ (i=1,2) \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} B_1 \approx B_2$$

Graphically:

$$\begin{tikzcd} M \arrow[r, Rightarrow, "(\beta\eta)^*"] \arrow[dd, Rightarrow, "(\beta\eta)^*"'] & M_1 \arrow[d, Rightarrow, dashed, "(\beta\eta)^*"] \\ & N_1 \arrow[d, dash, dashed, "\approx"'] \\ M_2 \arrow[r, Rightarrow, dashed, "(\beta\eta)^*"'] & N_2 \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, Rightarrow, "(\beta\eta)^*"] \arrow[dd, Rightarrow, "(\beta\eta)^*"'] & A_1 \arrow[d, Rightarrow, dashed, "(\beta\eta)^*"] \\ & B_1 \arrow[d, dash, dashed, "\approx"'] \\ A_2 \arrow[r, Rightarrow, dashed, "(\beta\eta)^*"'] & B_2 \end{tikzcd}$$

Proof. Applying Lemma 4.4.13 with $R = (\beta/\approx)$ and $S = (\eta/\approx)$ we conclude that $\Rightarrow_{\beta\eta/\approx}$ is Church-Rosser. The result then follows from Corollaries 4.4.6 and 4.4.8. □

Theorem 4.4.20. *For all well-formed types M_i $(i = 1, 2)$, and type families A_i $(i = 1, 2)$, the following holds:*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1 =_{\mathcal{E}'\beta\eta} M_2 \textit{ implies } \exists N_1, N_2 \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_i \Rightarrow^{*}_{\beta\eta} N_i \ (i=1,2) \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} N_1 \approx_{\mathcal{E}'} N_2$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_1 =_{\mathcal{E}'\beta\eta} A_2 \textit{ implies } \exists B_1, B_2 \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_i \Rightarrow^{*}_{\beta\eta} B_i \ (i=1,2) \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} B_1 \approx_{\mathcal{E}'} B_2$$

Proof. Using the assumption and Theorem 4.4.4, we construct derivations of

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1 \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M_2$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_1 \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} A_2$$

We argue by induction on these. The only interesting case is transitivity:

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M' \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M_2}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 \Leftrightarrow_{\beta\eta/\approx_{\mathcal{E}'}} M_2}$$

whose proof is given by the following diagram:

$$\begin{tikzcd}
M_1 \arrow[rr, Leftrightarrow, "\beta\eta/\approx_{\mathcal{E}'}"] \arrow[d, Rightarrow, "(\beta\eta)^*"'] & & M' \arrow[rr, Leftrightarrow, "\beta\eta/\approx_{\mathcal{E}'}"] \arrow[dl, Rightarrow, "(\beta\eta)^*"'] \arrow[dr, Rightarrow, "(\beta\eta)^*"] & & M_2 \arrow[d, Rightarrow, "(\beta\eta)^*"] \\
P_1 \arrow[r, dash, dashed, "\approx_{\mathcal{E}'}"'] \arrow[d, Rightarrow, "(\beta\eta)^*"'] & P_2 \arrow[d, Rightarrow, "(\beta\eta)^*"'] & & P_3 \arrow[r, dash, dashed, "\approx_{\mathcal{E}'}"'] \arrow[d, Rightarrow, "(\beta\eta)^*"] & P_4 \arrow[d, Rightarrow, "(\beta\eta)^*"] \\
N_1 \arrow[r, dash, dashed, "\approx_{\mathcal{E}'}"'] & N_2 \arrow[rr, dash, dashed, "\approx"'] & & N_3 \arrow[r, dash, dashed, "\approx_{\mathcal{E}'}"'] & N_4
\end{tikzcd}$$

Existence of the P_i is ensured by the inductive hypothesis, while the N_i are given by Theorem 4.4.19 and $\approx$-postponement (Lemmas 4.4.5 and 4.4.7). ☐

Corollary 4.4.21. *Canonical forms are unique modulo $\approx$.*

Proof. Let $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : A$, and assume

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Rightarrow_{\beta\eta} N_i \ (i = 1, 2)$$
$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_i \Downarrow A \ (i = 1, 2)$$

By Proposition 4.1.7, we conclude

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_1 =_{\beta\eta} N_2$$

and the result follows from Theorem 4.4.20 and Lemma 4.2.1 by letting $\mathcal{E}' = \cdot$. ☐

4.5 Summary

The study of η-expansion is more delicate in LF than it is in the simply-typed lambda calculus. As usual, new problems arise from having objects appearing inside types. Here, the type used in the η-expansion may contain unexpanded objects; expanding those, in turn, may introduce new expressions to be expanded, and so on. To ensure the well-founded character of expansion, we need to impose additional conditions; specifically, we require the types used in η-expansion to be normal. To avoid circularity in our definitions, we give two distinct characterizations of normality: one is syntactic, and defines as normal those terms that can be derived through a given typing judgment; the other is the traditional one, operational, that identifies normal forms as those expressions that cannot further expanded. Both of these were introduced in Section 4.1, and proved equivalent in Section 4.2.

As we started analyzing this newly introduced η-expansion relation, another feature of the language got in the way. Because of the presence of $\mathcal{E}$-conversion of types, η-expansion is not confluent in general. The best we could hope for was to prove confluence modulo an equivalence relation $\approx$ that identifies those terms obtained by $\mathcal{E}\beta\eta$-converting the types of the λ-abstracted variables. We defined this equivalence relation in Section 4.3; confluence of β-reduction and η-expansion modulo $\approx$ was shown, not without substantial effort, in Section 4.4.

Part II

Rewriting

Chapter 5

Dependence Relations

Dependence relations, introduced in Chapter 2 as part of our calculus, will play an important role in defining rewriting in LF. We dedicate this chapter to an in-depth discussion of these relations, their motivation and fundamental properties.

5.1 Main Ideas

Central in the definition of a rewriting notion is the operation of replacement of a sub-term by another. For simply-typed theories, it can be easily shown that this operation preserves the well-typedness of the overall term, provided that the replaced sub-term has the same type as the original one.

This property unfortunately does not extend to dependent types, mainly because of the application rule

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : (\Pi x : A.\ B) \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N : [N/x]B}$$

that allows the argument N to be propagated inside the type. If we were to perform a replacement inside N, giving rise to a term N', we would have to operate analogous replacements inside $[N/x]B$ to transform it into $[N'/x]B$. Moreover, if this application term was part of a larger object U, its new type might turn out to be incompatible with its use inside U.

To illustrate concretely these problems, consider the following:

Example 4. Consider the signature in Figure 5.1, representing a system to infer which natural numbers are even.

In this signature, the object

$$\mathbf{even}_+\ \mathbf{0}\ (\mathbf{0} + \mathbf{0})\ \mathbf{even}_0\ (\mathbf{even}_+\ \mathbf{0}\ \mathbf{0}\ \mathbf{even}_0\ \mathbf{even}_0)$$

is well typed, but rewriting $(\mathbf{0} + \mathbf{0}) \to \mathbf{0} : \mathbf{nat}$ we get

$$\mathbf{even}_+\ \mathbf{0}\ \mathbf{0}\ \mathbf{even}_0\ (\mathbf{even}_+\ \mathbf{0}\ \mathbf{0}\ \mathbf{even}_0\ \mathbf{even}_0)$$

which is not.

We would like to therefore restrict replacement only to the cases where it is type-safe, and specifically in

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N : [N/x]B}$$

to allow replacement inside N only when we can guarantee $x \notin \mathcal{FV}(B)$.

Of course, imposing the syntactic restriction $x \notin \mathcal{FV}(B)$ would not work, since by Lemma 2.2.5, the type of an object is unique only modulo $\mathcal{E}\beta\eta$-conversion, and in most case we would able, through β-expansion, to "sneak in" the variable x. Restricting ourselves to canonical forms, i.e. requiring

$$x \notin \mathcal{FV}(B') \text{ for all } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B' =_{\mathcal{E}\beta\eta} B \text{ such that } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B' \Downarrow \text{type}$$

$$\text{dom}(\Sigma) = \{\mathbf{nat}, \mathbf{0}, \mathbf{s}, +, \mathbf{o}, \mathbf{even}, \mathbf{proof}, \mathbf{even_0}, \mathbf{even_{ss}}, \mathbf{even_+} \mathbf{even_{s+}}\}$$

$$\begin{aligned}
&\Sigma(\mathbf{nat}) = \text{type} \\
&\Sigma(\mathbf{0}) = \mathbf{nat} \\
&\Sigma(\mathbf{s}) = \Pi x : \mathbf{nat}.\ \mathbf{nat} \\
&\Sigma(+) = \Pi x : \mathbf{nat}.\ \Pi y : \mathbf{nat}.\ \mathbf{nat}
\end{aligned}$$

$$\begin{aligned}
&\Sigma(\mathbf{o}) = \text{type} \\
&\Sigma(\mathbf{even}) = \Pi x : \mathbf{nat}.\ \mathbf{o}
\end{aligned}$$

$$\begin{aligned}
&\Sigma(\mathbf{proof}) = \Pi x : \mathbf{o}.\ \text{type} \\
&\Sigma(\mathbf{even_0}) = \mathbf{proof}\ (\mathbf{even}\ \mathbf{0}) \\
&\Sigma(\mathbf{even_{ss}}) = \Pi x : \mathbf{nat}.\ \Pi p : \mathbf{proof}(\mathbf{even}\ x).\ \mathbf{proof}(\mathbf{even}\ (\mathbf{s}\ (\mathbf{s}\ x))) \\
&\Sigma(\mathbf{even_+}) = \Pi x_1 : \mathbf{nat}.\ \Pi x_2 : \mathbf{nat}.\ \Pi p_1 : \mathbf{proof}(\mathbf{even}\ x_1). \\
&\qquad\qquad\qquad\qquad \Pi p_2 : \mathbf{proof}(\mathbf{even}\ x_2).\ \mathbf{proof}(\mathbf{even}\ (x_1 + x_2)) \\
&\Sigma(\mathbf{even_{s+}}) = \Pi x_1 : \mathbf{nat}.\ \Pi x_2 : \mathbf{nat}.\ \Pi p_1 : \mathbf{proof}(\mathbf{even}\ x_1). \\
&\qquad\qquad\qquad\qquad \Pi p_2 : \mathbf{proof}(\mathbf{even}\ x_2).\ \mathbf{proof}(\mathbf{even}\ ((\mathbf{s}\ x_1) + (\mathbf{s}\ x_2)))
\end{aligned}$$

Figure 5.1: Even Numbers

would not work either, since by Corollary 4.4.21 canonical forms are not unique, one obtainable from another by converting the types of the λ-abstracted variables.

Hence, we choose an approach more semantical in nature, making use of dependence relations. To explain the main ideas behind these, we refer back to Example 4. There, we define types **o** for formulas and **nat** for natural numbers. Objects of type **o** may contain some of type **nat**, as arguments of **even**. However, it would be "unnatural", for our intended use of this signature, to have objects of type **nat** containing formulas on their inside. Dependence relations allow us to formally state "naturality" conditions for functional dependence, and enforce these using the typing judgments. Specifically, the relations $\prec^\tau$ and $\prec^{\mathrm{t}}$ are used with the following informal meaning

$A \prec^\tau B$	objects of type A are allowed to appear inside type B
$A \prec^{\mathrm{t}} B$	objects of type A are allowed to appear inside objects of type B

We enforce these requirements, for example, in the abstraction rule

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : \text{type} \qquad \Gamma, x : A \vdash^{\prec}_{\Sigma,\varepsilon} M : B \qquad A \preceq^{\mathrm{t}} B}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \lambda x : A.\ M : \Pi x : A.\ B}$$

by preventing functional objects violating our naturality conditions to be generated.

By Definition 2.1.3, dependence relations must be also be closed under some conditions. Informally, these ensure that some of the logical consequences of the naturality choices we decided to impose are also reflected in the dependence relation. We examine some of these conditions in detail:

- $\text{head}(A_i) \prec^\tau a$ if $\Sigma(a) = \Pi x_1 : A_1.\ \Pi x_2 : A_2.\ \ldots \Pi x_n : A_n.$ type

  All base types P with $\text{head}(P) = a$ are required to satisfy the naturality requirements:

  $$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} a\ M_1\ M_2\ \ldots\ M_n : \text{type} \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_i : B_i \text{ implies } B_i \prec^\tau (a\ M_1\ M_2\ \ldots\ M_n)$$

- $A \prec^\tau A'$ if $A \prec^{\mathrm{t}} B \prec^\tau A'$ for some B

  This roughly corresponds to transitivity of containment. If objects of type B are allowed to appear inside A', and objects of type A may appear inside objects of type B, in conclusion we allow objects of type A to appear inside type A'.

- $A \prec^{\mathrm{t}} A'$ if $A \prec^{\mathrm{t}} B \prec^{\mathrm{t}} A'$ for some B

  Also corresponding to transitivity of object containment.

5.2 General Properties

The informal discussion above is made precise by the following technical results:

Lemma 5.2.1.

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A \textit{ and } N \textit{ a sub-object of } M, \textit{ then } \textit{type}(\Gamma(M, N)) \preceq^{\mathrm{t}} A$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A : K \textit{ and } N \textit{ a sub-object of } A, \textit{ then } \textit{type}(\Gamma(A, N)) \prec^{\tau} A$$

Proof. Both statement are proved simultaneously by induction on the derivations. We examine in detail a few interesting cases:

- Case

  $$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_1 : \Pi x : B.\ K \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_2 : B}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_1\ M_2 : [M_2/x]K}$$

  First, observe that $\mathrm{head}(A) = \mathrm{head}(A_1)$, so it will suffice to show $\textit{type}(\Gamma(A, N)) \prec^\tau A_1$.

  If N is a sub-object of A_1, the result follows by applying the inductive hypothesis to the derivation of

  $$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_1 : \Pi x : B.\ K$$

  If N is a sub-object of M_2, by inductive hypothesis on the derivation of

  $$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_2 : B$$

  we get $\textit{type}(\Gamma(A, N)) \preceq^{\mathrm{t}} B$.

  Let $a = \mathrm{head}(A_1)$, and

  $$\Sigma(a) = \Pi x_1 : C_1.\ \Pi x_2 : C_2.\ \ldots.\ \Pi x_n : C_n.\ \mathrm{type}$$

  For some i, we see that it must be $\mathrm{head}(C_i) = \mathrm{head}(B)$. We conclude $B \prec^\tau A_1$, and from this the result follows, since

  $$\textit{type}(\Gamma(A, N)) \preceq^{\mathrm{t}} B \prec^\tau A_1$$

- Case

  $$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A_1 : \mathrm{type} \qquad \Gamma, x : A_1 \vdash^{\prec}_{\Sigma,\varepsilon} A_2 : \mathrm{type} \qquad A_1 \preceq^{\mathrm{t}} A_2}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \Pi x : A_1.\ A_2 : \mathrm{type}}$$

  Similarly to the previous case, if N is a sub-object of A_2, we are done by inductive hypothesis on the derivation of

  $$\Gamma, x : A_1 \vdash^{\prec}_{\Sigma,\varepsilon} A_2 : \mathrm{type}$$

  since we observe $\mathrm{head}(\Pi x : A_1.\ A_2) = \mathrm{head}(A_2)$.

Otherwise N is a sub-object of A_1, and we can apply the inductive hypothesis to

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A_1 : \mathrm{type}$$

to get $type(\Gamma(A, N)) \prec^{\tau} A_1$, and hence the result:

$$type(\Gamma(A, N)) \prec^{\tau} A_1 \preceq^{\mathrm{t}} A_2$$

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_1 : \Pi x : B.\ C \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_2 : B}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_1\ M_2 : [M_2/x]C}$$

If $M = N$, the result follows trivially. Also, if N is a sub-object of M_1, we are done by inductive hypothesis on the derivation of

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_1 : \Pi x : B.\ C$$

since $\mathrm{head}([M_2/x]C) = \mathrm{head}(\Pi x : B.\ C)$.

If N is contained into M_2, by the inductive hypothesis on

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_2 : B$$

we get $type(\Gamma(M, N)) \preceq^{\mathrm{t}} B$. By Type Consistency, we get a derivation of

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} \Pi x : B.\ C : \mathrm{type}$$

and by inversion on this derivation, we find $B \preceq^{\mathrm{t}} C$. Hence

$$type(\Gamma(M, N)) \preceq^{\mathrm{t}} B \preceq^{\mathrm{t}} C$$

and we are done because $\mathrm{head}([M_2/x]C) = \mathrm{head}(C)$.

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A_1 : \mathrm{type} \qquad \Gamma x : A \vdash^{\preceq}_{\Sigma,\varepsilon} M_2 : B \qquad A_1 \preceq^{\mathrm{t}} B}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} \lambda x : A_1.\ M_2 : \Pi x : A_1.\ B}$$

Notice that $\mathrm{head}(\Pi x : A_1.\ B) = \mathrm{head}(B)$; hence, if N occurs inside M_2 we are done by applying the inductive hypothesis on the derivation

$$\Gamma, x : A_1 \vdash^{\preceq}_{\Sigma,\varepsilon} M_2 : B$$

On the other hand, if N occurs inside A_1, using the inductive assumptions on

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} A_1 : \mathrm{type}$$

we conclude $type(\Gamma(M, N)) \prec^{\tau} A_1$. Then $type(\Gamma(M, N)) \prec^{\mathrm{t}} A_1$, and we conclude the proof of this case by applying transitivity of $\prec^{\mathrm{t}}$:

$$type(\Gamma(M, N)) \prec^{\mathrm{t}} A_1 \preceq^{\mathrm{t}} B$$

□

Corollary 5.2.2.

$$\Gamma, z : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\varepsilon} M : A, z \in \mathcal{FV}(M) \textit{ implies } A_\circ \preceq^{\mathrm{t}} A$$
$$\Gamma, z : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\varepsilon} A : K, z \in \mathcal{FV}(A) \textit{ implies } A_\circ \prec^{\tau} A$$

Proof. Apply Lemma 5.2.1 to any of the free occurrences of z in the given terms. □

5.3 Summary

Dependence relations were first introduced by us in [71] as a tool to define rewriting in a non-equational setting. They will used in a similar way here, but their role, as we will see, will be greatly expanded.

We gave in this chapter, first informally in Section 5.1 and then formally in Section 5.2, the intuition that lies behind their introduction.

Chapter 6

Rewriting

In this chapter we will introduce our notion of rewriting. This is similar to the one employed in [71] for the non-equational calculus. Although, as we have seen in Chapter 4, uniqueness of canonical forms holds in a weaker form (i.e. modulo $\approx$), most of the results we proved in that paper lift to this equational version of LF.

6.1 Environments

Central in defining rewriting is the concept of contextual replacement of an expression with another. This is usually formalized in the literature by *contexts*, which are essentially "expression with a hole": the hole indicates where the replacement takes place, while the rest of the expression is left unchanged. One little notational difficulty that we encounter here is that the name "context" is already used in standard LF literature to denote something different. Hence we are forced to choose a different name for these entities. We decided for "environment", hoping that this choice will not confuse the reader too much.

Definition 6.1.1 (Environments). Environments are expressions defined by the following syntax:

$$\textit{Environments} \quad E \quad := \circ \mid E\ M \mid M\ E \mid \lambda x : A.\ E$$

Well-formed environments are constructed using the judgment:

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \quad E \text{ is an environment of type A}$$

formed according to the set of rules shown in Figure 6.1.

Remark. Our informal discussion of dependence relations in Chapter 5 motivates the side condition $A_\circ \not\prec^\tau B$ in the rule

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad A_\circ \not\prec^\tau B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M\ E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B}$$

above. This corresponds to restricting the realm of applicability of the rule to those cases where objects of type $A_\circ$ are not allowed to appear inside B. In particular, using the contrapositive of Corollary 5.2.2, $x \notin \mathcal{FV}(B)$, and hence replacement of any object $M_\circ$ for the hole $\circ$ does not modify the type of the overall expression.

Notation. We will use the letter E for environments. Given an environment E and an object $M_\circ$ we will denote by $E[\![M_\circ]\!]$ the object obtained by replacing the hole $\circ$ in E by the object $M_\circ$.

E will be said an *environment for* M provided there is a sub-object $M_\circ$ of M such that $M = E[\![M_\circ]\!]$; in this case, the sub-object $M_\circ$ is said to be *isolated by* E.

Finally, we will write $E_1 \cdot E_2$ to denote the environment obtained by composing the two environments E_1 and E_2, i.e. by replacing the hole of E_1 with E_2. Note that, even when both E_1 and E_2 are well-typed, $E_1 \cdot E_2$ needs not to be.

$$\frac{\Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} A_\circ : \mathrm{type} \qquad \Gamma_\circ \subseteq \Gamma}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \circ[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A_\circ} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A : \mathrm{type} \qquad \Gamma, x : A \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B \qquad A \preceq^{\mathrm{t}} B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : A.\ E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : \Pi x : A.\ B}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : \Pi x : A.\ B \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : A}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!]\ M : [M/x]B}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad A_\circ \not\prec^{\tau} B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M\ E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A'}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A'}$$

Figure 6.1: Environments

We list below a few basic properties of environments.

Proposition 6.1.2 (Weakening). *Let* $\Sigma \subseteq \Sigma'$, $\mathcal{E}'' \subseteq \mathcal{E} \subseteq \mathcal{E}'$, $\mathcal{E}'' \subseteq \mathcal{E}''' \subseteq \mathcal{E}'$, $\Gamma \subseteq \Gamma'$,

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \textit{ implies } \Gamma' \vdash^{\preceq}_{\Sigma',\mathcal{E}'} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$$

Proof. By induction on the derivation. Note that, unlike in Proposition 2.2.1, the statement may not hold if we increase $\prec^{\tau}$. □

Proposition 6.1.3. *If* $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$, *then* $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A : \mathrm{type}$ *and* $A_\circ \preceq^{\mathrm{t}} A$.

Proof. Both properties are proved by induction on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$. For the second one, a non-trivial case is given by the abstraction rule:

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : \Pi x : A.\ B \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad A_\circ \not\prec^{\tau} B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M\ E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B}$$

  By inductive hypothesis, we get $A_\circ \preceq^{\mathrm{t}} A$. Using Type Consistency, we conclude $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \Pi x : A.\ B : \mathrm{type}$; by inversion we see $A \preceq^{\mathrm{t}} B$, from which the result follows by transitivity of $\prec^{\mathrm{t}}$.

□

Proposition 6.1.4. *Let* $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$, *then*

$$\Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ : A_\circ \textit{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![M_\circ]\!] : A$$
$$\Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ =_{\mathcal{E}'\beta\eta} M'_\circ \textit{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![M_\circ]\!] =_{\mathcal{E}'\beta\eta} E[\![M'_\circ]\!]$$

Proof. By induction on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$. There is one subtle point to consider when dealing with one of the application rules:

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 : \Pi x : B.\ C \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B \qquad A_\circ \not\prec^{\tau} C}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1\ E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : C}$$

The inductive hypothesis gives us $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} E_2[\![M_\circ]\!] : \Pi x : B.\ C$. Hence, using the application rule we conclude

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1 : \Pi x : B.\ C \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} E_2[\![M_\circ]\!] : B}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1\ E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : [E_2[\![M_\circ]\!]/x]C}$$

We are left to show that $x \notin \mathcal{FV}(B)$, so that $[E_2[\![M_\circ]\!]/x]B = B$. Assume otherwise, using Type Consistency and inversion, we get a derivation of

$$\Gamma, x : B \vdash^{\prec}_{\Sigma,\varepsilon} C : \text{type}$$

and from this, using Corollary 5.2.2 we conclude $B \prec^{\tau} C$.

Using the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$ and Proposition 6.1.3, we also have $A_\circ \preceq^{\text{t}} B$; hence

$$A_\circ \preceq^{\text{t}} B \prec^{\tau} C$$

and therefore $A_\circ \prec^{\tau} C$, a contradiction.

$\square$

Notation. We will denote by θE the environment obtained by applying the substitution θ to all the expressions in E, i.e. the environment defined as

$$\begin{aligned}
\theta(\circ) &= \circ \\
\theta(\lambda x : A.\ E') &= \lambda z : \theta A.\ \theta([z/x]E') \qquad (z \text{ fresh}) \\
\theta(E'\ N) &= \theta E'\ \theta N \\
\theta(M\ E') &= \theta M\ \theta E'
\end{aligned}$$

Note that this definition basically extends the notion of applying a substitution to contexts; in particular, if $M = E[\![M_\circ]\!]$ then $\theta M = (\theta E)[\![\theta M_\circ]\!]$.

Proposition 6.1.5. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta : \Delta$, *then*

$$\Delta \vdash^{\prec}_{\Sigma,\varepsilon} E[\![\Delta, \Gamma_\circ \vdash \circ :]\!]A \ \textit{implies}\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta E[\![\Gamma, \theta\Gamma_\circ \vdash \circ : \theta A_\circ]\!] : \theta A$$

Proof. By a straightforward induction on the derivations, observing that, for each type C,

$$\text{head}(\theta C) = \text{head}(C)$$

and hence $A_\circ \not\prec^{\tau} B$ iff $\theta A_\circ \not\prec^{\tau} \theta B$. $\square$

When $M_\circ$ is a sub-object of M isolated by an environment E, i.e. $M = E[\![M_\circ]\!]$, if M is well-formed then $M_\circ$ is too; the same it is not true in general for E. The following gives sufficient conditions for this to happen:

Proposition 6.1.6. *Let* $M = E[\![M_\circ]\!]$, *where*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A \qquad\qquad \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\varepsilon} M_\circ : A_\circ$$

if $A_\circ \not\prec^{\tau} A$ *then*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} E[\![\Gamma(M, M_\circ) \vdash \circ : A_\circ]\!] : A$$

Proof. By induction on the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : A$. We examine the case

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_1 : \Pi x : B.\ C \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E_2[\![M_\circ]\!] : B}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_1\ E_2[\![M_\circ]\!] : [E_2[\![M_\circ]\!]/x]C}$$

By Type Consistency and inversion, $B \preceq^{\mathrm{t}} C$. Hence we see that $A_\circ \not\prec^{\tau} B$: otherwise, since

$$\mathrm{head}(C) = \mathrm{head}([E_2[\![M_\circ]\!]/x]C) = \mathrm{head}(A)$$

we would have

$$A_\circ \prec^{\tau} B \preceq^{\mathrm{t}} A$$

leading to $A_\circ \prec^{\tau} A$, a contradiction to the assumptions. From this, we see, using Corollary 5.2.2, that $x \notin \mathcal{FV}(C)$, so $[E_2[\![M_\circ]\!]/x]C = C$. Applying the inductive hypothesis to $E_2[\![M_\circ]\!]$, we obtain

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E_2[\![\Gamma(M, M_\circ) \vdash \circ : A_\circ]\!] : B$$

and therefore

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_1 : \Pi x : B.\ C \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E_2[\![\Gamma(M, M_\circ) \vdash \circ : A_\circ]\!] : B \qquad A_\circ \not\prec^{\tau} C}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_1\ E_2[\![\Gamma(M, M_\circ) \vdash \circ : A_\circ]\!] : C}$$

□

When $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![M_\circ]\!] : A$, where E is not well-formed, replacing $M_\circ$ with another object $M'_\circ$ of the same type is not guaranteed to give rise to another object of type A, or even a well-formed object at all. One notable exception is when $M_\circ$ and $M'{}_\circ$ are pairwise convertible:

Proposition 6.1.7. *Let $M = E[\![M_\circ]\!]$ and $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![M_\circ]\!] : A$,*

$$\Gamma(M, M_\circ) \vdash^{\preceq}_{\Sigma,\varepsilon} M_\circ \Rightarrow_\beta M'_\circ \textit{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![M_\circ]\!] \Rightarrow_\beta E[\![M'_\circ]\!]$$
$$\Gamma(M, M_\circ) \vdash^{\preceq}_{\Sigma,\varepsilon} M_\circ \Rightarrow_\eta M'_\circ \textit{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![M_\circ]\!] \Rightarrow_\eta E[\![M'_\circ]\!]$$

Proof. Both statements are proved by a straightforward induction on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![M_\circ]\!] : A$. □

When two environment are used to perform replacement of disjoint sub-expressions of the same object, the order in which this replacement is performed is immaterial; this simple remark is formalized by the following:

Definition 6.1.8. Given two environments E_1 and E_2 for a same object M, we say that E_1 and E_2 *are disjoint*, and write $E_1 \uparrow E_2$, if neither one is contained in the other, i.e. there is no E_0 such that $E_1 = (E_2 \cdot E_0)$ or $E_2 = (E_1 \cdot E_0)$.

Proposition 6.1.9. *Let $E_0[\![M_0]\!] = E_1[\![M_1]\!]$, $E_0 \uparrow E_1$, where*

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E_i[\![\Delta_i \vdash \circ : A_i]\!] : A\ (i = 0, 1)$$
$$\Delta_i \vdash^{\preceq}_{\Sigma,\varepsilon} M_i : A_i\ (i = 0, 1)$$

then for each object $\Delta_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M'_0 : A_0$ there is an environment $(E_1)_{E_0[\![M'_0]\!]}$,

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} (E_1)_{E_0[\![M'_0]\!]}[\![\Delta_1 \vdash \circ : A_1]\!] : A$$

such that $(E_1)_{E_0[\![M'_0]\!]}[\![M_1]\!] = E_0[\![M'_0]\!]$. Moreover, given $E_2[\![M_2]\!] = E_1[\![M_1]\!]$, and

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E_2[\![\Delta_2 \vdash \circ : A_2]\!] : A \qquad \Delta_2 \vdash^{\preceq}_{\Sigma,\varepsilon} M_2 : A_2 \qquad E_2 \uparrow E_0, E_2 \uparrow E_1$$

we also have $(E_1)_{E_0[\![M'_0]\!]} \uparrow (E_2)_{E_0[\![M'_0]\!]}$.

Proof. By induction on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E_0[\![\Delta_0 \vdash \circ : A_0]\!] : A$. $\square$

The notion of disjoint environments is useful, for example, when considering objects $[N/x]M$, in cases when the object N can be rewritten to N'. Occurrences of N in $[N/x]M$ are isolated by disjoint environments, so we can use Proposition 6.1.9 to rewrite all of them, in any order we choose, to eventually obtain $[N'/x]M$. However, for Proposition 6.1.9 to be applicable in these cases, we have to make sure that the environments are well-formed. It is easy to convince ourself that this is equivalent to checking well-formedness of all the environments isolating occurrences of x in M, and a sufficient condition for this to happen is given by the following:

Proposition 6.1.10. *Let* $\Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : A$*, with* $A_\circ \not\prec^{\tau} A$*, then* $M = M[\![x]\!]$*, where*

$$\Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} M[\![\Gamma_x \vdash \circ : A_\circ]\!] : A$$

for all occurrences of x *in* M *(for some context* Γ_x *which depends on the occurrence).*

Proof. By induction on the derivation of $\vdash^{\preceq}_{\Sigma,\mathcal{E}} \Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : A$. The only two interesting cases are

- Case

$$\frac{\Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} B : \text{type} \qquad \Gamma, x : A_\circ, \Delta, y : B \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 : C \qquad B \preceq^{\mathrm{t}} C}{\Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : B.\ M_1 : \Pi x : B.\ C}$$

  Since our definition of environment does not allow the hole to appear inside a type, we have first to check that $x \notin \mathcal{FV}(B)$. If that was the case, then we would conclude, by Corollary 5.2.2, $A_\circ \prec^{\tau} B$.

  Hence

$$A_\circ \prec^{\tau} B \preceq^{\mathrm{t}} C$$

  from which we derive $A_\circ \prec^{\tau} C$, and, since $\text{head}(\Pi x : B.\ C) = \text{head}(C)$, $A_\circ \prec^{\tau} A$, a contradiction to the assumptions.

  Therefore it must be $x \in \mathcal{FV}(M_1)$, and the result follows by inductive hypothesis.

- Case

$$\frac{\Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 : \Pi x : B.\ C \qquad \Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_2 : B}{\Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1\ M_2 : [M_2/x]C}$$

  If the occurrence of x is in M_1, then the result is trivially obtained by inductive hypothesis, since $\text{head}(\Pi x : B.\ C) = \text{head}([M_2/x]C)$.

  If the occurrence of x is in M_2, in order to apply the inductive hypothesis we have to show that $A_\circ \not\prec^{\tau} B$.

  From the derivation of

$$\Gamma, x : A_\circ, \Delta \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 : \Pi x : B.\ C$$

  we conclude, by Type Consistency and inversion, $B \preceq^{\mathrm{t}} C$.

  If $A_\circ \prec^{\tau} B$, we would then have

$$A_\circ \prec^{\tau} B \preceq^{\mathrm{t}} C$$

  from which we conclude $A_\circ \prec^{\tau} C$, and, since $\text{head}(C) = \text{head}([M_2/x]C)$, $A_\circ \prec^{\tau} A$, a contradiction.

$\square$

Remark. Note that Proposition 6.1.10 is not subsumed by Proposition 6.1.6, although they share the same assumption and have similar results: in Proposition 6.1.10 we also show that all the free occurrences of x are isolated by some environment, i.e. that x does not appear free in any of the types of the λ-abstracted variables in M.

6.2 Rewriting

We have now all the machinery necessary to state a formal definition of rewriting. Such definition will have the same structure as the one in [71], but additional care is necessary, since now canonical forms are only unique modulo λ-equivalence, and therefore we will want the definition to be invariant with respect of the particular representative chosen in each equivalence class.

Definition 6.2.1 (Higher-order Term Rewriting Systems). A rewrite rule is a pair of objects l and r such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} l \Downarrow A \text{ is a pattern} \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} r : A \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \downarrow \text{type}$$

We will denote a rewrite rule like the above by $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} l \to r : A$.

A set of rewrite rules $\mathcal{R}$ is a Higher-order Term Rewriting System (HTRS) provided that for any two rules

$$\Gamma_i \vdash^{\preceq}_{\Sigma,\mathcal{E}} l_i \to r_i : A_i \in \mathcal{R} \ (i = 1, 2)$$

we have $A_1 \not\prec^{\tau} A_2$.

Definition 6.2.2. Given a set of rules $\mathcal{R}$, we define

$$\text{heads}(\mathcal{R}) \stackrel{def}{=} \left\{ \text{head}(A) \,\middle|\, \Gamma \vdash^{\preceq}_{\Gamma,\mathcal{E}} l \to r : A \in \mathcal{R} \right\}$$

Notation. We will often write $\mathcal{R}$ for heads($\mathcal{R}$), whenever the intended meaning is clear from the context.

Furthermore, we will extend the previously defined relations over type constants to sets by existential quantification; for example

$$a \prec^{\tau} \mathcal{R} \text{ iff } \exists\, (\Gamma \vdash^{\preceq}_{\Gamma,\mathcal{E}} l \to r : A) \in \mathcal{R} \,.\; a \prec^{\tau} A$$
$$a \not\prec^{\mathrm{t}} \mathcal{R} \text{ iff } \forall\, (\Gamma \vdash^{\preceq}_{\Gamma,\mathcal{E}} l \to r : A) \in \mathcal{R} \,.\; a \not\prec^{\mathrm{t}} A$$

Under these conventions, the condition for a set of rules $\mathcal{R}$ to be a HTRS can be concisely expressed by

$$\mathcal{R} \not\prec^{\tau} \mathcal{R}$$

Remark. The requirement $\mathcal{R} \not\prec^{\tau} \mathcal{R}$ will be needed when defining critical pairs. See the remark following Definition 7.3.2 for more details.

Definition 6.2.3 (Rewriting). We say that M rewrites under the HTRS $\mathcal{R}$ to M', and write

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}} M'$$

provided that

$$\begin{array}{ll}
M = E[\![M_\circ]\!] & M' = E[\![M'_\circ]\!] \\
\Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ : A_\circ & \Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} M'{}_\circ \Downarrow A_\circ \\
\Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ =_{\beta\eta} \theta l & \Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} M'_\circ =_{\beta\eta} \theta r \\
\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A & \Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ \\
\Delta_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} l \to r : C_\circ \in \mathcal{R} &
\end{array}$$

Rewriting modulo $\mathcal{E}'$, where $\mathcal{E}' \subseteq \mathcal{E}$, is defined as

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 \to_{\mathcal{R}/\mathcal{E}'\beta\eta} M_2 \quad \text{iff} \quad \exists N_i \text{ s.t. } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_i =_{\mathcal{E}'\beta\eta} N_i \ (i = 1, 2) \text{ and } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_1 \to_{\mathcal{R}} N_2$$

6.3 Canonical Rewriting

Although the definition of rewriting modulo $\mathcal{E}' \subseteq E$ we just gave is the most natural one, it is not a very practical one to use. Too much freedom is left as to where to apply the rewrite step; rewriting may take place over an object that is not canonical, or even not β-normal. For this reason, we introduce another form of rewriting modulo an equivalence relation, where we force rewriting to take place only on canonical forms. We will also prove that the two definitions are equivalent, hence allowing us to use them interchangeably throughout the rest of this thesis.

Definition 6.3.1. We say M *rewrites canonically* modulo $\mathcal{E}'$, in symbols

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{R}_{\mathcal{E}'\beta\eta}} M'$$

if

$$\begin{array}{ll}
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx_{\mathcal{E}'} E[\![M_\circ]\!] & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (M')_{\Downarrow} \approx_{\mathcal{E}'} E[\![M'_\circ]\!] \\
\Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \Downarrow A_\circ & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M'{}_\circ \Downarrow A_\circ \\
\Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ =_{\mathcal{E}'\beta\eta} \theta l & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ =_{\mathcal{E}'\beta\eta} \theta r \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ \\
\Delta_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} l \rightarrow r : C_\circ \in \mathcal{R} &
\end{array}$$

Using Corollary 4.4.21 and Weakening, one immediately sees that $\rightarrow_{\mathcal{R}_{\mathcal{E}'\beta\eta}}$ is well-defined, i.e. it is independent from the choice of the canonical representatives $M_{\Downarrow}$ and $(M')_{\Downarrow}$.

To study the properties of canonical rewriting, it is useful to first extend some of the definitions and results of Chapter 4 to environments:

Definition 6.3.2. Canonical and atomic environments are constructed using the judgments

$$\begin{array}{ll}
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A & E \text{ is a canonical environment} \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow A & E \text{ is an atomic environment}
\end{array}$$

derived accordingly to the following inference system:

$$\frac{\Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} A_\circ \downarrow \text{type}}{\Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \circ[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A_\circ} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \downarrow \text{type}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \text{type} \qquad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow B \qquad A \preceq^{\mathrm{t}} B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow \Pi x : A.\ B}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A' \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' : \text{type}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow \Pi x : A.\ B \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E\ M[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow [M/x]B}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow \Pi x : A.\ B \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A \qquad A_\circ \not\prec^{\tau} B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow B}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A' \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' : \text{type}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow A'}$$

We state below a few basic properties of canonical and atomic environments. The proofs are easily obtained arguing by induction on the derivations, and are omitted here.

Proposition 6.3.3.

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$$

Proposition 6.3.4. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \Downarrow A_\circ$,

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![M_\circ]\!] \Downarrow A$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![M_\circ]\!] \downarrow A$$

Proposition 6.3.5. *Let* $M = E[\![M_\circ]\!]$, *where*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \Downarrow A_\circ \qquad \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} A_\circ \downarrow \mathrm{type}$$

then

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow A$$

We want now to show equivalence of $\rightarrow^*_{\mathcal{R}/\mathcal{E}'\beta\eta}$ and $\rightarrow^*_{\mathcal{R}_{\mathcal{E}'\beta\eta}}$. One direction is immediate:

Proposition 6.3.6. *We have*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{R}_{\mathcal{E}'\beta\eta}} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{R}/\mathcal{E}'\beta\eta} M'$$

Proof. Assume $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{R}_{\mathcal{E}'\beta\eta}} M'$. By Definition 6.3.1, we have

$$\begin{array}{ll}
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx_{\mathcal{E}'} E[\![M_\circ]\!] & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (M')_{\Downarrow} \approx_{\mathcal{E}'} E[\![M'_\circ]\!] \\
\Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \Downarrow A_\circ & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M'{}_\circ \Downarrow A_\circ \\
\Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ =_{\mathcal{E}'\beta\eta} \theta l & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ =_{\mathcal{E}'\beta\eta} \theta r \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ \\
\Delta_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} l \rightarrow r : C_\circ \in \mathcal{R} &
\end{array}$$

Using Lemma 4.2.3, and Propositions 6.1.4 and 4.3.2, we see

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} E[\![\theta l]\!] \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\theta l]\!] \rightarrow_{\mathcal{R}} E[\![\theta r]\!] \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' =_{\mathcal{E}'\beta\eta} E[\![\theta r]\!]$$

□

Corollary 6.3.7. *If* $\rightarrow_{\mathcal{R}/\mathcal{E}'\beta\eta}$ *is strongly normalizing, so is* $\rightarrow_{\mathcal{R}_{\mathcal{E}'\beta\eta}}$

Proof. By Proposition 6.3.6, any infinite $(\rightarrow_{\mathcal{R}_{\mathcal{E}'\beta\eta}})$-rewriting sequence is also a $(\rightarrow_{\mathcal{R}/\mathcal{E}'\beta\eta})$-rewriting one. □

Proving the other direction of the equivalence suggested by Proposition 6.3.6 is more involved, and requires some preliminary work.

Definition 6.3.8. Let $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A$ a well-formed object, we define

$$\beta(M) = \{N \mid M \rightarrow^+_\beta N\}$$

the set of β-reducts of M. Given a well-formed type A or kind K, the sets $\beta(A)$ and $\beta(K)$ are similarly defined.

Remark. By Theorem 3.3.10, β-reduction is strong normalizing on well-formed expressions, so these are always finite, and therefore it makes sense to compute their size $|\beta(M)|$, $|\beta(A)|$, and $|\beta(K)|$.

Lemma 6.3.9.

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to_{\mathcal{R}} N' \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow_{\eta} M'' \textit{ implies } \exists N \textit{ s. t. } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M' \Rightarrow^{=}_{\eta} N \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M'' \to_{\mathcal{R}} N$$

In pictures:

$$\begin{tikzcd}
M \arrow[r, "\mathcal{R}"] \arrow[d, Rightarrow, "\eta"'] & M' \arrow[d, Rightarrow, dashed, "\eta^{=}"] \\
M'' \arrow[r, dashed, "\mathcal{R}"'] & N
\end{tikzcd}$$

Proof. By definition, $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to_{\mathcal{R}} M'$ if

$$\begin{array}{ll}
M = E[\![M_\circ]\!] & M' = E[\![M'_\circ]\!] \\
\Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M_\circ : A_\circ & \Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M'_\circ \Downarrow A_\circ \\
\Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M_\circ =_{\beta\eta} \theta_\circ l & \Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M'_\circ =_{\beta\eta} \theta_\circ r
\end{array}$$

where

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad \Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} \theta : \Delta_\circ \qquad \Delta_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} l \to r : C_\circ \in \mathcal{R}$$

We argue by induction on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$, using inversion on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow_{\eta} M''$.

The only non-trivial case is when $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : \Pi x : B.\ C$, and

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M : \Pi x : B'.\ C' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} B' \Downarrow \text{type} \qquad M \text{ not a } \lambda}{\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow_{\eta} \lambda x : B'.\ (M\ x)}$$

By Uniqueness of Types we conclude

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} B =_{\mathcal{E}\beta\eta} B' \qquad \Gamma, x : B \vdash^{\preceq}_{\Sigma,\varepsilon} C =_{\mathcal{E}\beta\eta} C' \qquad B' \preceq^{\mathrm{t}} C$$

and from this the result follows, using the (well-formed) environment $E' = (\lambda x : B'.\ E\ x)$. ☐

Corollary 6.3.10. *We have*

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to^{*}_{\mathcal{R}} M' \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow^{*}_{\eta} M'' \textit{ implies } \exists N \textit{ s. t. } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M' \Rightarrow^{*}_{\eta} N \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M'' \to^{*}_{\mathcal{R}} N$$

Graphically:

$$\begin{tikzcd}
M \arrow[r, "\mathcal{R}^*"] \arrow[d, Rightarrow, "\eta^*"'] & M' \arrow[d, Rightarrow, dashed, "\eta^*"] \\
M'' \arrow[r, dashed, "\mathcal{R}^*"'] & N
\end{tikzcd}$$

Moreover, the rewrite sequences $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to^{*}_{\mathcal{R}} M'$ and $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M'' \to^{*}_{\mathcal{R}} N$ are of the same length.

Proof. By induction on the length of the sequence of rewriting steps and η-expansions, using Lemma 6.3.9. ☐

Lemma 6.3.11. *We have:*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}} M' \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow_{\beta} M'' \text{ implies } \exists N' \text{ and } N'' \text{ s. t. } \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \Rightarrow^{*}_{\beta} N' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M'' \to^{*}_{\mathcal{R}} N'' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N' \Rightarrow^{*}_{\eta} N'' \end{cases}$$

In pictures:

$$\begin{tikzcd}
M \arrow[r, "\mathcal{R}"] \arrow[dd, Rightarrow, "\beta"'] & M' \arrow[d, Rightarrow, "\beta^{*}"] \\
 & N' \arrow[dl, Rightarrow, dashed, "\eta^{*}"] \\
M'' \arrow[r, dashed, "\mathcal{R}^{*}"'] & N''
\end{tikzcd}$$

Moreover, $|\beta(N')| = |\beta(N'')|$.

Proof. By definition, $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}} M'$ if

$$\begin{array}{ll} M = E[\![M_\circ]\!] & M' = E[\![M'_\circ]\!] \\ \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ : A_\circ & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ \Downarrow A_\circ \\ \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ =_{\beta\eta} \theta_\circ l & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ =_{\beta\eta} \theta_\circ r \end{array}$$

where

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ \qquad \Delta_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} l \to r : C_\circ \in \mathcal{R}$$

The proof then proceeds by induction on the structure of the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$. As usual, we consider just a couple of representative cases:

- Case

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E_1[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : \Pi x : B.\ C \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_2 : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E_1[\![\Gamma_\circ \vdash \circ : A_\circ]\!]\ M_2 : [M_2/x]C}$$

If

$$M = E_1[\![M_\circ]\!]\ M_2 \to_\beta E_1[\![M_\circ]\!]\ M''_2 = M''$$

then the result follows trivially, letting $N' = N'' = E_1[\![M'_\circ]\!]\ M''_2$.

Otherwise, if

$$M = E_1[\![M_\circ]\!]\ M_2 \to_\beta M''_1\ M_2 = M''$$

the result follows by applying the inductive hypothesis to $E[\![M'_\circ]\!]$ and M''_1.

The only interesting case is therefore

$$M = E_1[\![M_\circ]\!]\ M_2 = (\lambda x : B'.\ M_3)\ M_2 \to_\beta [M_2/x]M_3 = M''$$

By inversion, it can be shown $E_1 = \lambda x : B'.\ E_3$, where

$$\begin{array}{l} \Gamma, x : B' \vdash^{\prec}_{\Sigma,\mathcal{E}} E_3[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : C' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : B.\ C =_{\mathcal{E}\beta\eta} \Pi : B'.\ C' \end{array}$$

and $E_3[\![M_\circ]\!] = M_3$. Note that this inversion property does not hold in general, and uses the assumption that all the rules in $\mathcal{R}$ are of base type.

Applying Proposition 6.1.5 with $\theta = (id_\Gamma, x \mapsto M_2)$, we see that the environment θE_3 is well-formed, and $(\theta E_3)[\![\theta M_\circ]\!] = [M_2/x]M_3$.

By Lemma 4.2.3

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta M'_\circ \Rightarrow^*_{\beta\eta} (\theta M'_\circ)_{\Downarrow} \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\theta M'_\circ)_{\Downarrow} \Downarrow [M_2/x]A_\circ$$

Without loss of generality, we can assume that $N''_\circ = (\theta M'_\circ)_{\Downarrow}$ is obtained by first reducing $\theta M'_\circ$ to its β-normal form $N'_\circ$. Using Proposition 6.1.7,

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : B'.\ E_3[\![M'_\circ]\!])\ M_2 \Rightarrow_\beta (\theta E_3)[\![\theta M'_\circ]\!] \Rightarrow^*_\beta (\theta E_3)[\![N'_\circ]\!]$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\theta E_3)[\![N'_\circ]\!] \Rightarrow^*_\eta (\theta E_3)[\![N''_\circ]\!]$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} [M_2/x]M_3 \rightarrow_{\mathcal{R}} (\theta E_3)[\![N''_\circ]\!]$$

Finally, since $N'_\circ$ is of base type, no new β-redexes are created by η-expanding $N' = (\theta E_3)[\![N'_\circ]\!]$ to $N'' = (\theta E_3)[\![N''_\circ]\!]$; hence $|\beta(N')| = |\beta(N'')|$.

- Case

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 : \Pi x : B.\ C \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B \qquad A_\circ \not\prec^\tau C}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1\ E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : C}$$

As before, the cases

$$M = M_1\ E_2[\![M_\circ]\!] \rightarrow_\beta M''_1\ E_2[\![M_\circ]\!] = M''$$

and

$$M = M_1\ E_2[\![M_\circ]\!] \rightarrow_\beta M_1\ M''_2 = M''$$

are easily handled.

The only non-trivial case is therefore again

$$M = (\lambda x : B'.\ M_3)\ E_2[\![M_\circ]\!] \rightarrow_\beta [E_2[\![M_\circ]\!]/x]M_3 = M''$$

where, by inversion on the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : B'.\ M_3) : \Pi x : B.\ C$, we conclude

$$\Gamma, x : B' \vdash^{\prec}_{\Sigma,\mathcal{E}} M_3 : C' \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}\beta\eta} B'$$
$$B' \preceq^{\mathrm{t}} C' \qquad\qquad \Gamma, x : B' \vdash^{\prec}_{\Sigma,\mathcal{E}} C =_{\mathcal{E}\beta\eta} C'$$

From the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B$ we obtain, by type conversion and Proposition 6.1.3, $A_\circ \preceq^{\mathrm{t}} B'$. We now see that it must be $B' \not\prec^\tau C$, because otherwise we would have

$$A_\circ \preceq^{\mathrm{t}} B' \prec^\tau C$$

and therefore $A_\circ \prec^\tau C$, a contradiction to the assumptions.

Hence $B' \not\prec^\tau C'$, and we can apply Proposition 6.1.10 to obtain disjoint environments

$$\Gamma, x : B' \vdash^{\prec}_{\Sigma,\mathcal{E}} M_3[\![\Gamma_x \vdash \circ : B]\!] : C'$$

From these, using Proposition 6.1.5 with $\theta = (id_\Gamma, x \mapsto E_2[\![M_\circ]\!])$, we get (still disjoint) environments

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\theta M_3)[\![\theta\Gamma_x \vdash \circ : \theta B']\!] : \theta C'$$

Hence, by repeated applications of Proposition 6.1.9, we show

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} [E_2[\![M_\circ]\!]/x]M_3 \rightarrow^*_{\mathcal{R}} [E_2[\![M'_\circ]\!]/x]M_3$$

and therefore the result, letting $N' = N'' = [E_2[\![M'_\circ]\!]/x]M_3$.

Corollary 6.3.12. *The following holds:*

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to^*_{\mathcal{R}} M' \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Rightarrow^*_{\beta} M'' \textit{ implies } \exists N \textit{ s. t. } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M' \Rightarrow^*_{\beta\eta} N \textit{ and } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M'' \to^*_{\mathcal{R}} N$$

Pictorially, we have the following diagram:

$$\begin{tikzcd}
M \arrow[r, "\mathcal{R}^*"] \arrow[d, Rightarrow, "\beta^*"'] & M' \arrow[d, Rightarrow, dashed, "\beta\eta^*"] \\
M'' \arrow[r, dashed, "\mathcal{R}^*"'] & N
\end{tikzcd}$$

Proof. We argue by a double induction argument: the primary induction is on $|\beta(M)|$, the secondary one on the length of the rewriting sequence. The proof is obtained from Lemma 6.3.11 and Corollary 6.3.10 by diagram chasing:

$$\begin{tikzcd}
M \arrow[rr, "\mathcal{R}"] \arrow[dd, Rightarrow, "\beta"'] & & M_1' \arrow[r, "\mathcal{R}^*"] \arrow[d, Rightarrow, dashed, "\beta^*"] & M' \arrow[d, Rightarrow, dashed, "\beta\eta^*"] \\
& & P_1' \arrow[r, dashed, "\mathcal{R}^*"'] \arrow[dl, Rightarrow, dashed, "\eta^*"] & N' \arrow[dl, Rightarrow, dashed, "\eta^*"] \\
M_1'' \arrow[r, dashed, "\mathcal{R}^*"'] \arrow[d, Rightarrow, dashed, "\beta^*"'] & P_1'' \arrow[r, dashed, "\mathcal{R}^*"'] & N'' \arrow[d, Rightarrow, dashed, "\beta\eta^*"] & \\
M'' \arrow[rr, dashed, "\mathcal{R}^*"'] & & N &
\end{tikzcd}$$

□

Proposition 6.3.13. *Let* $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to_{\mathcal{R}} M'$*, then*

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Downarrow A \textit{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M' \Downarrow A$$
$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \downarrow A \textit{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M' \downarrow A$$

Proof. By definition, $\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to_{\mathcal{R}} M'$ if there are

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad \Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} \theta : \Delta_\circ \qquad \Delta_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} l \to r : C_\circ \in \mathcal{R}$$

such that

$$\begin{array}{ll}
M = E[\![M_\circ]\!] & M' = E[\![M'_\circ]\!] \\
\Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M_\circ : A_\circ & \Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M'_\circ \Downarrow A_\circ \\
\Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M_\circ =_{\varepsilon\beta\eta} \theta_\circ l & \Gamma_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M'_\circ =_{\varepsilon\beta\eta} \theta_\circ r
\end{array}$$

By Proposition 6.3.5, from

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \Downarrow A \quad (\text{resp. } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \downarrow A)$$

we conclude

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \Downarrow A \quad (\text{resp. } \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] \downarrow A)$$

and the result follows by Proposition 6.3.4. □

Theorem 6.3.14. *We have*

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \rightarrow^*_{\mathcal{R}/\mathcal{E}'\beta\eta} M' \textit{ if and only if } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \rightarrow^*_{\mathcal{R}_{\mathcal{E}'\beta\eta}} M' \textit{ or } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M'$$

Proof. One direction is easily proven using Proposition 6.3.6.

For the other one, it suffices to show

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{R}/\mathcal{E}'\beta\eta} N \text{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \rightarrow^*_{\mathcal{R}_{\mathcal{E}'\beta\eta}} N$$

Assume $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{R}/\mathcal{E}'\beta\eta} N$. By definition, there are objects M' and N' such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}'\beta\eta} M' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M' \rightarrow_{\mathcal{R}} N' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N' =_{\mathcal{E}'\beta\eta} N$$

Without loss of generality, we can assume that $(M')_{\Downarrow}$ is obtained by first β-normalizing M, and then η-expanding all the non-saturated sub-objects. We can then apply Corollaries 6.3.12, 6.3.10, and Proposition 6.3.13, (in this order) to conclude

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} (M')_{\Downarrow} \rightarrow^*_{\mathcal{R}} N''$$

for some canonical objects N'' such that $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N'' =_{\beta\eta} N'$.

By Theorem 4.4.20

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx_{\mathcal{E}'} (M')_{\Downarrow} \qquad\qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N'' \approx_{\mathcal{E}'} N_{\Downarrow}$$

hence the result. □

We conclude this section by extending one final notion,$\mathcal{E}'\lambda$-equivalence, to environments:

Definition 6.3.15. Let $\mathcal{E}' \subseteq \mathcal{E}$, we define $\mathcal{E}'\lambda$-*equivalent environments* by means of the judgment:

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E' \quad E \text{ and } E' \text{ are } \mathcal{E}'\lambda\text{-equivalent environments}$$

built up from the rules:

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E'} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E'}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E' \approx_{\mathcal{E}'} E} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E' \quad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E' \approx_{\mathcal{E}'} E''}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E''}$$

$$\frac{}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \circ \approx_{\mathcal{E}'} \circ}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A' \quad \Gamma, x : A \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : A.\ E \approx_{\mathcal{E}'} \lambda x : A'.\ E} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A : \text{type} \quad \Gamma, x : A \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E'}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : A.\ E \approx_{\mathcal{E}'} \lambda x : A.\ E'}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E' \quad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : A}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E\ M \approx_{\mathcal{E}'} E'\ M} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \quad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E\ M \approx_{\mathcal{E}'} E\ M'}$$

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M' \quad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M\ E \approx_{\mathcal{E}'} M'\ E} \qquad \frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M : A \quad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E'}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M\ E \approx_{\mathcal{E}'} M\ E'}$$

Proposition 6.3.16. *Let* $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ : A_\circ$*, then*

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E' \textit{ implies } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![M_\circ]\!] \approx_{\mathcal{E}'} E'[\![M_\circ]\!]$$

Proof. By a simple induction on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E'$. □

6.4 Summary

The results about dependence relations proved previously are used here to formulate our definition of rewriting modulo an equational theory. In particular, dependence relations are used when defining environments (Definition 6.1.1) to rule out cases, like the one presented in Chapter 5, where replacement of sub-expressions invalidates the type of the overall expression.

We gave two definitions of rewriting; the first, in Section 6.2, is the most natural one, but not necessarily the most convenient to use in our study. The second one, canonical rewriting, introduced in Section 6.3, sets some restriction on the form of the expression that we allow to be rewritten. These restriction do not translate in a loss of generality: in the same section, we proved that the reflexive-transitive closures of these two relations coincide (Theorem 6.3.14).

Chapter 7

Confluence for Simpler Equational Theories

In studying HTRSs, two properties that are of particular significance are confluence and strong normalization. If a HTRS possesses both of these properties, equivalence of any two expressions modulo $\mathcal{E}(\mathcal{R})$ (the equational theory generated from $\mathcal{R}$ by disregarding orientation) can be effectively computed by reducing both expressions to their $\mathcal{R}$-normal forms. In this dissertation we will concentrate on the first of these two properties, confluence, although some of the results we present can be used in establishing strong normalization.

Testing confluence of a Term Rewriting System (TRS) using the definition is not practical, since it would require checking that the diagram

$$\begin{tikzcd} M \arrow[r, "\mathcal{R}^*"] \arrow[d, "\mathcal{R}^*"'] & M_1 \arrow[d, dashed, "\mathcal{R}^*"] \\ M_2 \arrow[r, dashed, "\mathcal{R}^*"'] & N \end{tikzcd}$$

holds for all possible choice of well-formed objects M, M_1, M_2.

If we assume $\mathcal{R}$ to be strongly normalizing, our task is simplified somewhat, because by Newmann's Lemma [53] it suffices to check for *local confluence*, i.e.

$$\begin{tikzcd} M \arrow[r, "\mathcal{R}"] \arrow[d, "\mathcal{R}"'] & M_1 \arrow[d, dashed, "\mathcal{R}^*"] \\ M_2 \arrow[r, dashed, "\mathcal{R}^*"'] & N \end{tikzcd}$$

Still, this requires testing infinitely many triples of objects M, M_1, M_2. In this respect, one of the most fundamental contributions to the theory of TRSs can be considered Knuth and Bendix's *Critical Pair Criterion*, which reduces the check for local confluence to a finite set of pairs of objects M_1 and M_2.

When considering confluence modulo an equivalence relation, an adaptation of this criterion is available, due to Huet [29]. However, here, critical pairs coming from overlapping rules and equations have also to be considered. There are circumstances, of course, where examining some of these additional pairs is unnecessary, like for example when $\mathcal{R}$ and $\mathcal{E}$ are defined over disjoint signatures. However, these situations are quite rare and or difficult to check for. In this chapter, we will show how dependence relations allow us to isolate a frequently-occuring class of such cases.

7.1 Simpler Equational Theories

The class of problems we will discuss in this chapter are those where the equational theory $\mathcal{E}$ is defined on lower classes of types than the HTRS $\mathcal{R}$. To make the notion of "lower class of types" more precise, we give the following definitions:

Definition 7.1.1. Let $\mathcal{E}$ an equational theory, we define

$$\mathrm{heads}(\mathcal{E}) \stackrel{def}{=} \left\{\mathrm{head}(A) \,\middle|\, \Gamma \vdash^{\preceq}_{\Sigma} M = N : A \in \mathcal{E}\right\}$$

Notation. As before with heads($\mathcal{R}$), we will often write, by abuse of notation, $\mathcal{E}$ for heads($\mathcal{E}$).

Definition 7.1.2. An equational theory $\mathcal{E}$ is said to be *simpler* than a HTRS $\mathcal{R}$ defined over it, provided that $\mathcal{R} \not\preceq^{\mathrm{t}} \mathcal{E}$.

Remark. Informally speaking, an equational theory $\mathcal{E}$ is simpler if no object rewritable by $\mathcal{R}$ can appear as a sub-object of $\mathcal{E}$.

In Chapter 6, we proved that rewriting modulo $\mathcal{E}$ is equivalent to canonical rewriting (modulo $\mathcal{E}$), and therefore these two notions could be used interchangeably. In particular, one nice feature of the latter is that it commutes with $\beta\eta$-conversion:

$$\begin{tikzcd}
M \arrow[r, "\mathcal{R}_{\mathcal{E}\beta\eta}"] \arrow[d, equal, "\beta\eta"'] & M_1 \arrow[d, equal, dashed, "\beta\eta"] \\
M_2 \arrow[r, dashed, "\mathcal{R}_{\mathcal{E}\beta\eta}"'] & N
\end{tikzcd}$$

The above diagram fails to be true in general if we replace $=_{\beta\eta}$ with $=_{\mathcal{E}\beta\eta}$. However, in the case of simpler $\mathcal{E}$ this can be shown to hold.

Proposition 7.1.3. *Let $M = E[\![M_\circ]\!]$, with*

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M' \qquad\qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A \qquad\qquad \Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ : A_\circ$$

if $A_\circ \not\preceq^{\mathrm{t}} \mathcal{E}'$ then $M' = E'[\![M'_\circ]\!]$ for some environment E' and object $M'_\circ$ such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}'} E' \qquad\qquad \Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ \approx_{\mathcal{E}'} M'_\circ$$

Proof. By induction on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A$. We examine a few cases:

- Case

$$\frac{\Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} A_\circ : \mathrm{type} \qquad \Gamma_\circ \subseteq \Gamma}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \circ[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A_\circ}$$

  Immediate from the assumptions, by choosing $E' = \circ$, $M'_\circ = M'$.

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} B : \mathrm{type} \qquad \Gamma, x : B \vdash^{\preceq}_{\Sigma,\mathcal{E}} E_1[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : C \qquad B \preceq^{\mathrm{t}} C}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : B.\ E_1[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : \Pi x : B.\ C}$$

  By inversion on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'$, one shows $M' = \lambda x : B'.M'_1$, with

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}\beta\eta} B' \qquad\qquad \Gamma, x : B' \vdash^{\preceq}_{\Sigma,\mathcal{E}} E_1[\![M_\circ]\!] \approx_{\mathcal{E}'} M'_1$$

We can therefore apply the inductive hypothesis, obtaining E_1' and $M_\circ'$ as required. Then, the result follows by letting $E' = \lambda x : B'.\ E_1'$:

$$\dfrac{\dfrac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}\beta\eta} B' \qquad \Gamma, x : B \vdash^{\prec}_{\Sigma,\mathcal{E}} E_1[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : C}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : B.\ E_1 \approx_{\mathcal{E}'} \lambda x : B'.\ E_1} \qquad \dfrac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B' : \mathrm{type} \qquad \Gamma, x : B' \vdash^{\prec}_{\Sigma,\mathcal{E}} E_1 \approx_{\mathcal{E}'} E_1'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : B'.\ E_1 \approx_{\mathcal{E}'} \lambda x : B'.\ E_1'}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : B.\ E_1 \approx_{\mathcal{E}'} \lambda x : B'.\ E_1'}$$

- Case

$$\dfrac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 : \Pi x : B.\ C \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : B \qquad A_\circ \not\preceq^{\tau} C}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1\ E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : C}$$

Arguing by inversion on the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}'} M'$, we can show $M' = M_1'\ M_2'$, where

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 \approx_{\mathcal{E}} M_1' \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E_2[\![M_\circ]\!] \approx_{\mathcal{E}} M_2'$$

and the result follows again by inductive hypothesis.
Note that the inversion property we used here is not true in general, since the derivation of

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1\ E_2[\![M_\circ]\!] \approx_{\mathcal{E}'} M'$$

might be obtained by one or more top-level applications of the $\mathcal{E}'$-conversion rule:

$$\dfrac{(\Delta \vdash^{\prec}_{\Sigma} M_\circ = M_\circ' : C_\circ) \in \mathcal{E}' \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} U =_{\beta\eta} \theta M_\circ \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} U' =_{\beta\eta} \theta M_\circ'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} U \approx_{\mathcal{E}'} U'}$$

We have to show that this is not the case. If it was, since (by Proposition 6.1.3) $A_\circ \preceq^{\mathrm{t}} C$, and

$$\mathrm{head}(C) = \mathrm{head}(\theta C_\circ) = \mathrm{head}(C_\circ)$$

we would have $A_\circ \preceq^{\mathrm{t}} C_\circ$, a contradiction to the assumption $A_\circ \not\preceq^{\mathrm{t}} \mathcal{E}'$.

$\square$

Corollary 7.1.4. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}\beta\eta} N$*, then*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}_{\mathcal{E}\beta\eta}} M' \textit{ implies } \exists N' \textit{ s.t. } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \to_{\mathcal{R}_{\mathcal{E}\beta\eta}} N' \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' =_{\mathcal{E}\beta\eta} N'$$

In pictures:

$$\begin{tikzcd} M \arrow[r, "\mathcal{R}_{\mathcal{E}\beta\eta}"] \arrow[d, equal, "\mathcal{E}\beta\eta"'] & M' \arrow[d, equal, dashed, "\mathcal{E}\beta\eta"] \\ N \arrow[r, dashed, "\mathcal{R}_{\mathcal{E}\beta\eta}"'] & N' \end{tikzcd}$$

Proof. By definition, $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}_{\mathcal{E}\beta\eta}} M'$ if

$$\begin{array}{ll} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx_{\mathcal{E}} E[\![M_\circ]\!] & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (M')_{\Downarrow} \approx_{\mathcal{E}} E[\![M_\circ']\!] \\ \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \Downarrow A_\circ & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ \Downarrow A_\circ \\ \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ =_{\mathcal{E}\beta\eta} \theta l & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ' =_{\mathcal{E}\beta\eta} \theta r \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : A_\circ]\!] : A & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ \\ \Delta_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} l \to r : C_\circ \in \mathcal{R} & \end{array}$$

Using Theorem 4.4.20 we show

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx_{\mathcal{E}} N_{\Downarrow}$$

Let E' be the environment for $N_{\Downarrow}$ obtained by applying Proposition 7.1.3, it is easily seen that $N' = E'[\![M'_\circ]\!]$ is as required. $\square$

7.2 Generalized Patterns

In what follows, the assumption that all the LHSs in a HTRS are patterns will play, as we will see, a crucial role. Unfortunately, this property is not preserved by taking sub-objects (previously bound variables may become free), and for this reason it is ill-suited to proving properties by structural induction. Therefore, we introduce the following generalization:

Definition 7.2.1. Let $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Downarrow A$ (resp. $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \downarrow A$), and $\Gamma' \subseteq \Gamma$, M is said to be a Γ'-*pattern* if every occurrence of a variable $x \in \mathrm{dom}(\Gamma')$ in M appears enclosed in sub-objects of the form $M_\circ = x\ N_1\ N_2 \ldots N_k$, where

$$\Gamma(M, M_\circ) \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ \Downarrow \mathit{type}(\Gamma(M, M_\circ))$$

and the N_i are all η-convertible to distinct variables $y_i \notin \mathrm{dom}(\Gamma')$. A sub-object $M_\circ$ of the above form is called a Γ'-*generalized variable.*

Another important notion that we will use throughout this dissertation is *stability*:

Definition 7.2.2. An object M is said to be Γ-*stable* if $M = c\ M_1\ M_2 \ldots M_k$, or $M = x\ M_1\ M_2 \ldots M_k$ with $x \notin \mathrm{dom}(\Gamma)$.

An environment E is Γ-stable if

$$E = (E_1 \cdot (M\ E_2)) \text{ implies } M \text{ is } \Gamma\text{-stable}$$

What makes stability a key property is that, as the following fact shows, all the interesting environments of a Γ'-pattern are Γ'-stable.

Proposition 7.2.3. *Assume* $M = E[\![M_\circ]\!]$, *with*

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Downarrow A \ (\textit{resp. } \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \downarrow A))\ \Gamma'\text{-}\textit{pattern}$$

for some $\Gamma' \subseteq \Gamma$, *and*

$$\Gamma(M, M_\circ) \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ \Downarrow A_\circ$$

then E *is* Γ'*-stable if and only if* $M_\circ$ *is not a proper sub-object occurrence of some* Γ'*-generalized variable in* $E[\![M_\circ]\!]$.

Proof. By a simple induction on the derivation of $E[\![M_\circ]\!]$. When dealing with the rule

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![M_\circ]\!] \downarrow A \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} A \downarrow \mathrm{type}}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E[\![M_\circ]\!] \Downarrow A}$$

we check whether $E[\![M_\circ]\!]$ is a Γ'-generalized variable. If it is, then we immediately verify that, unless $E = \circ$, E is not Γ'-stable. If $E[\![M_\circ]\!]$ is not a Γ'-generalized variable, we proceed by inductive hypothesis. $\square$

Another useful property of Γ'-stable objects and environments is that their "structure" is preserved by those substitutions which are the identity on variables outside Γ', in a sense made precise by the following:

Proposition 7.2.4. *Let*

$$\Delta, \Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \downarrow A \qquad\qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta : \Delta,$$

if M *is* Δ*-stable then there is an object* M' *such that*

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} M' \downarrow \theta A \qquad\qquad \Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta M =_{\beta\eta} M'$$

Proof. We can consider θM as the object obtained by applying the (well-formed, by the assumptions) substitution

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta, id_{\Delta'}) : \Delta, \Delta'$$

The proof then proceeds by induction on the derivation of $\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M \downarrow A$.

- Case

$$\frac{(\Delta, \Delta')(x) = A}{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} x \downarrow A}$$

By stability of M, we have $x \in \mathrm{dom}(\Delta')$, hence $(\theta, id_{\Delta'})(x) = x$, and therefore $M' = x$ is as required:

$$\frac{(\Gamma, \theta\Delta')(x) = \theta A}{\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} x \downarrow \theta A}$$

- Case

$$\frac{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M_1 \downarrow \Pi x : B.\ C \qquad \Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M_2 \Downarrow B}{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M_1\ M_2 \downarrow [M_2/x]C}$$

By inductive hypothesis we get an object M_1' such that

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M_1' \downarrow \theta(\Pi x : B.\ C) \qquad\qquad \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta M_1 =_{\beta\eta} M_1'$$

Using Lemma 4.2.3, we have also

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta M_2)_{\Downarrow} \Downarrow \theta B \qquad\qquad \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta M_2 =_{\beta\eta} (\theta M_2)_{\Downarrow}$$

so we see $M' = M_1\ (\theta M_2)_{\Downarrow}$ is as required.

□

Corollary 7.2.5. *Let $M = E[\![M_\circ]\!]$, with*

$$\begin{array}{ll} \Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} E[\![M_\circ]\!] \Downarrow A & (\textit{resp.}\ \Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} E[\![M_\circ]\!] \downarrow A) \\ (\Delta, \Delta')(M, M_\circ) \vdash^{\prec}_{\Sigma,\varepsilon} M_\circ \Downarrow A_\circ & \\ (\Delta, \Delta')(M, M_\circ) \vdash^{\prec}_{\Sigma,\varepsilon} A_\circ \Downarrow \mathrm{type} & \\ \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \theta : \Delta & \end{array}$$

if E is Δ-stable then there is an environment $(\theta E)_{\Downarrow}$ such that

$$\begin{array}{ll} \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta E)_{\Downarrow}[\![(\theta M_\circ)_{\Downarrow}]\!] \Downarrow \theta A & (\textit{resp.}\ \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta E)_{\Downarrow}[\![(\theta M_\circ)_{\Downarrow}]\!] \downarrow \theta A) \\ \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta(E[\![M_\circ]\!]) =_{\beta\eta} (\theta E)_{\Downarrow}[\![(\theta M_\circ)_{\Downarrow}]\!] & \end{array}$$

Proof. By induction on the derivation of well-typedness of $E[\![M_\circ]\!]$. We consider a few interesting cases:

- Case

$$\frac{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} B \Downarrow \mathrm{type} \qquad \Delta, \Delta', x : B \vdash^{\prec}_{\Sigma,\varepsilon} E_1[\![M_\circ]\!] \Downarrow C \qquad B \preceq^{\mathrm{t}} C}{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \lambda x : B.\ E_1[\![M_\circ]\!] \Downarrow \Pi x : B.\ C}$$

By inductive hypothesis, there is an environment $(\theta E_1)_{\Downarrow}$ such that

$$\begin{array}{l} \Gamma, \theta\Delta', x : \theta B \vdash^{\prec}_{\Sigma,\varepsilon} (\theta E_1)_{\Downarrow}[\![(\theta M_\circ)_{\Downarrow}]\!] \Downarrow \theta C \\ \Gamma, \theta\Delta', x : \theta B \vdash^{\prec}_{\Sigma,\varepsilon} \theta(E_1[\![M_\circ]\!]) =_{\beta\eta} (\theta E_1)_{\Downarrow}[\![(\theta M_\circ)_{\Downarrow}]\!] \end{array}$$

By Lemma 4.2.3, we get

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta B)_{\Downarrow} \Downarrow \text{type}$$
$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta B =_{\beta\eta} (\theta B)_{\Downarrow}$$

From this, we see that the result follows letting $(\theta E)_{\Downarrow} = \lambda x : (\theta B)_{\Downarrow}.\ (\theta E_1)_{\Downarrow}$.

- Case

$$\frac{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M \downarrow \Pi x : B.\ C \qquad \Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} E_1[\![M_\circ]\!] \Downarrow B \qquad A_\circ \not\prec^{\tau} C}{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M\ E_1[\![M_\circ]\!] \downarrow C}$$

By the assumptions, M is stable, so we can apply Proposition 7.2.4 to obtain a M' such that

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M' \downarrow \theta(\Pi x : B.\ C)$$
$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta M =_{\beta\eta} M'$$

Also, by the inductive assumptions, there is an environment $(\theta E_1)_{\Downarrow}$ such that

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta E_1)_{\Downarrow}[\![(\theta M_\circ)_{\Downarrow}]\!] \Downarrow \theta B \qquad \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta(E_1[\![M_\circ]\!]) =_{\beta\eta} (\theta E_1)_{\Downarrow}[\![(\theta M_\circ)_{\Downarrow}]\!]$$

and therefore, taking $(\theta E)_{\Downarrow} = M'\ (\theta E_1)_{\Downarrow}$, we have the result.

□

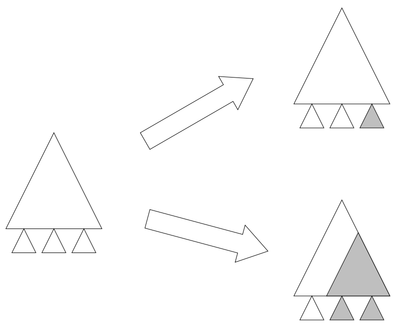
7.3 Confluence for Simpler Equational Theories

At the core of most adaptations of most of the adaptations of Knuth and Bendix's criterion there usually lies the same "key lemma". Although its statement can be quite complex, the idea behind it is quite simple: when rewriting a object instance θM, one of the following, mutually-exclusive, situations must happen:

1. rewriting takes place inside one of the sub-objects $\theta(x)$ (where $x \in \mathrm{dom}(\theta)$) introduced by the substitution;

2. the object being replaced is $\theta M'$, where M' is a non-variable sub-object of M.

Pictorially:



We present here our version of the key lemma:

Lemma 7.3.1. *Let $M' = E'[\![M'_\circ]\!]$, with*

$$\Delta, \Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \Downarrow A \ \Delta\text{-}pattern \qquad (resp.\ \Delta, \Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \downarrow A\ \Delta\text{-}pattern)$$
$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} E'[\![M'_\circ]\!] \Downarrow \theta A \qquad (resp.\ \Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} E'[\![M'_\circ]\!] \downarrow \theta A)$$
$$(\Gamma, \theta\Delta'_\circ)(M', M'_\circ) \vdash^{\preceq}_{\Sigma,\mathcal{E}} M'_\circ \Downarrow A'_\circ$$
$$(\Gamma, \theta\Delta'_\circ)(M', M'_\circ) \vdash^{\preceq}_{\Sigma,\mathcal{E}} A'_\circ \Downarrow \mathrm{type}$$
$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta : \Delta$$

such that

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta M =_{\mathcal{E}'\beta\eta} E'[\![M'_\circ]\!]$$

if $A'_\circ \not\preceq^{\mathrm{t}} \mathcal{E}'$, then there is a Δ-stable environment E such that

1. *either $M = E[\![M_\circ]\!]$, and*

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} E' \approx_{\mathcal{E}'} (\theta E)_{\Downarrow} \qquad (\Gamma, \theta\Delta'_\circ)(M', M'_\circ) \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta M_\circ =_{\mathcal{E}'\beta\eta} M'_\circ$$

2. *or $M = E[\![N]\!]$, where $N = x\ N_1\ N_2 \ldots N_k$ is a Δ-generalized variable, and there is an environment $E'_\circ$ such that*

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} E' \approx_{\mathcal{E}'} ((\theta E)_{\Downarrow} \cdot E'_\circ)$$
$$\Gamma, \theta\Delta'_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta(x) =_{\mathcal{E}'\beta\eta} \lambda y_1 : A_1 \ldots \lambda y_k : A_k.\ E'_\circ[\![M'_\circ]\!]$$

Proof. By induction on the derivation of M. We examine some representative cases:

- Case

$$\frac{\Delta, \Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} B \Downarrow \mathrm{type} \qquad \Delta, \Delta', x : B \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 \Downarrow C \qquad B \preceq^{\mathrm{t}} C}{\Delta, \Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : B.\ M_1 \Downarrow \Pi x : B.\ C}$$

  Since $\theta M = \lambda x : \theta B.\ \theta M_1$, we have

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : \theta B.\ \theta M_1 =_{\mathcal{E}'\beta\eta} E'[\![M'_\circ]\!]$$

  Using Theorem 4.4.20 and the fact that all the equations in $\mathcal{E}'$ are of base type, we easily see that $E' = \lambda x : B'.\ E'_1$, with

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta B =_{\mathcal{E}\beta\eta} B' \qquad \Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} B' \Downarrow \mathrm{type}$$
$$\Gamma, \theta\Delta', x : \theta B \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta M_1 =_{\mathcal{E}'\beta\eta} E'_1[\![M'_\circ]\!] \qquad \Gamma, \theta\Delta', x : \theta B \vdash^{\preceq}_{\Sigma,\mathcal{E}} E'_1[\![M'_\circ]\!] \Downarrow \theta C$$

  We can therefore apply the inductive hypothesis to M_1, obtaining an environment E_1 satisfying in the statement. We want to show that $E = \lambda x : B.\ E_1$ is as required.

  If E_1 satisfies case (1) of the statement, then

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : \theta B.\ E'_1 \approx_{\mathcal{E}'} \lambda x : \theta B.\ (\theta E_1)_{\Downarrow}$$

  Using the assumptions, Lemma 4.2.3, and Weakening

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} B' =_{\mathcal{E}\beta\eta} \theta B \qquad \Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta B =_{\mathcal{E}\beta\eta} (\theta B)_{\Downarrow}$$

  from which we see

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : B'.\ E'_1 \approx_{\mathcal{E}'} \lambda x : \theta B.\ E'_1 \qquad \Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : \theta B.\ (\theta E_1)_{\Downarrow} \approx_{\mathcal{E}'} \lambda x : (\theta B)_{\Downarrow}.\ (\theta E_1)_{\Downarrow}$$

  By transitivity

$$\Gamma, \theta\Delta' \vdash^{\preceq}_{\Sigma,\mathcal{E}} \lambda x : B'.\ E'_1 \approx_{\mathcal{E}'} \lambda x : (\theta B)_{\Downarrow}.\ (\theta E_1)_{\Downarrow}$$

  and hence, since $(\theta E)_{\Downarrow} = \lambda x : (\theta B)_{\Downarrow}.\ (E_1)_{\Downarrow}$, the result.

  A similar proof shows that if E_1 satisfies case (2), so does E.

- Case

$$\frac{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M \downarrow A \qquad \Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} A \downarrow \text{type}}{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M \Downarrow A}$$

There are two possibilities. If M is a Δ-generalized variable, then $M = x\ N_1\ N_2 \ldots N_k$, where $N_i \to^*_\eta y_i \in \mathrm{dom}(\Delta')$ are pairwise distinct variables. Let $\Delta'(y_i) = A_i$, we easily see

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta(x))_{\Downarrow} \approx \lambda y_1 : A_1 \ldots \lambda y_k : A_k.\ (\theta M)_{\Downarrow}$$

and therefore, using the assumptions and transitivity,

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta(x) =_{\beta\eta} \lambda y_1 : A_1 \ldots \lambda y_k : A_k.\ E'[\![M'_\circ]\!]$$

From this we see that letting $E = \circ$ and $E'_\circ = E'$, we satisfy case (2) of the statement.

If M is not a Δ-generalized variable, then M is Δ-stable. Moreover, from $\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} A \downarrow \text{type}$ we conclude $\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta A)_{\Downarrow} \downarrow \text{type}$, and hence, by inversion and type conversion,

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} E'[\![M'_\circ]\!] \downarrow \theta A,$$

and the result follows from the inductive assumptions.

- Case

$$\frac{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M_1 \downarrow \Pi x : B.\ C \qquad \Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M_2 \Downarrow B}{\Delta, \Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M_1\ M_2 \downarrow [M_2/x]C}$$

Since M is Δ-stable, so is M_1 , so by Proposition 7.2.4 there is M''_1 such that

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M''_1 \downarrow \theta(\Pi x : B.\ C) \qquad\qquad \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta M_1)_{\Downarrow} =_{\beta\eta} M''_1$$

Hence we conclude

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M''_1\ (M_2)_{\Downarrow} \downarrow \theta([M_2/x]C). \qquad\qquad \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta M =_{\beta\eta} M''_1\ (\theta M_2)_{\Downarrow}$$

Since, by the assumptions and Proposition 6.3.4,

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} E'[\![M'_\circ]\!] \downarrow \theta([M_2/x]C), \qquad\qquad \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta M =_{\beta\eta} E'[\![M'_\circ]\!]$$

we have

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M''_1\ (\theta M_2)_{\Downarrow} \approx_{\mathcal{E}'} E'[\![M'_\circ]\!]$$

We notice that both $E'[\![M'_\circ]\!]$ and $M''_1\ (\theta M_2)_{\Downarrow}$ are atomic, and that, by the condition $A'_\circ \not\preceq^{\mathrm{t}} \mathcal{E}'$, in the derivation above the $\mathcal{E}'$-conversion rule cannot be applied directly.

Under these circumstances, we are able to show that either $E' = E'_1\ M'_2$ or $E' = M'_1\ E'_2$. We assume the latter; the proof for the former is similar.

By inversion, and possibly type conversion, we get

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} E'_2[\![M'_\circ]\!] \Downarrow \theta B \qquad\qquad \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta M_2)_{\Downarrow} \approx_{\mathcal{E}'} E'_2[\![M'_\circ]\!]$$
$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} M''_1 \approx_{\mathcal{E}'} M'_1$$

and hence, by Proposition 4.3.2,

$$\Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} \theta M_1 =_{\mathcal{E}'\beta\eta} M'_1 \qquad\qquad \Gamma, \theta\Delta' \vdash^{\prec}_{\Sigma,\varepsilon} (\theta M_2) =_{\mathcal{E}'\beta\eta} E'_2[\![M'_\circ]\!]$$

By inductive hypothesis there is a canonical environment E_2 satisfying the statement. Then $E = M_1\ E_2$ is as required.

$\square$

We introduce now the definition of critical pair generated by two rules in $\mathcal{R}$:

Definition 7.3.2 ($\mathcal{R}|\mathcal{E}|\mathcal{R}$-Critical Pair). Let $\mathcal{R}$ be a HTRS, $\mathcal{E}' \subseteq \mathcal{E}$,

$$\Delta_i \vdash^{\preceq}_{\Sigma,\mathcal{E}} l_i \to r_i : C_i \in \mathcal{R} \qquad\qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_i : \Delta_i \ (i = 1, 2),$$

and E be a Δ_1-stable environment not isolating a generalized variable, and such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_1 l_1 =_{\mathcal{E}'\beta\eta} (\theta_1 E)_{\Downarrow}[\![\theta_2 l_2]\!]$$

then

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} <(\theta_1 E)_{\Downarrow}[\![\theta_2 r_2]\!], \theta_1 r_1>$$

is called a $(\mathcal{R}|\mathcal{E}'|\mathcal{R})$-*critical pair.*

A $(\mathcal{R}|\mathcal{E}'|\mathcal{R})$-critical pair $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} <M_1, M_2>$ is said to be *trivial* if there are objects N_1 and N_2 such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_i \to^{*}_{\mathcal{R}/\mathcal{E}'\beta\eta} N_i \ (i = 1, 2) \qquad\qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_1 =_{\mathcal{E}\beta\eta} N_2$$

Notation. When $\mathcal{E}' = \cdot$, $(\mathcal{R}|\mathcal{E}'|\mathcal{R})$-critical pairs will be called $(\mathcal{R}|\mathcal{R})$-critical pairs.

Remark. It requires some work to see that the definition above actually corresponds to the usual one. First, by considering a renaming substitution if necessary, we can assume in practice

$$\mathrm{dom}(\Delta_1) \cap \mathrm{dom}(\Delta_2) = \emptyset$$

and take $\theta = \theta_1 \cup \theta_2$ to be a most general $\mathcal{E}$-unifier of the two objects.

Secondly, by Proposition 7.2.3, the assumption of stability of E corresponds to the requirement of overlaps not to occur below generalized variables.

Notice that, using the assumption $\mathcal{R} \not\prec^{\tau} \mathcal{R}$ and Proposition 6.1.6, the environment E is always well-formed. Hence by Proposition 6.1.4 both objects of a critical pair are of type θC_1. Because we will be interested in HTRSs where all the critical pairs are trivial, and since $\to_{\mathcal{R}/\mathcal{E}'\beta\eta}$ requires both objects to be well typed, and of a same type, the cryptic requirement $\mathcal{R} \not\prec^{\tau} \mathcal{R}$ that we imposed in Definition 6.2.1 seems now fully justified.

We have now all the ingredients to prove the Critical Pair Criterion for simpler equational theories:

Lemma 7.3.3. *Let $\mathcal{E}' \subseteq \mathcal{E}$ be simpler than $\mathcal{R}$, then all the $(\mathcal{R}|\mathcal{E}'|\mathcal{R})$-critical pairs are trivial iff*

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}_{\mathcal{E}'\beta\eta}} N_i \ (i = 1, 2) \ \textit{implies} \ \exists N_i' \ \textit{s. t.} \ \begin{cases} \Gamma \vdash^{\preceq}_{\Sigma} N_i \to^{*}_{\mathcal{R}/\mathcal{E}'\beta\eta} N_i' \ (i = 1, 2) \\ \Gamma \vdash^{\preceq}_{\Sigma} N_1' =_{\mathcal{E}\beta\eta} N_2' \end{cases}$$

for all objects M, N_1 and N_2. In pictures, the condition above corresponds to the commutativity of the following diagram:

$$\begin{tikzcd}
M \arrow[r, "\mathcal{R}_{\mathcal{E}'\beta\eta}"] \arrow[dd, "\mathcal{R}_{\mathcal{E}'\beta\eta}"'] & N_1 \arrow[d, dashed, "(\mathcal{R}/\mathcal{E}'\beta\eta)^*"] \\
 & N_1' \arrow[d, dash, dashed, "\mathcal{E}\beta\eta"] \\
N_2 \arrow[r, dashed, "(\mathcal{R}/\mathcal{E}'\beta\eta)^*"'] & N_2'
\end{tikzcd}$$

Proof. We will start by proving the easy direction of this double implication. Assume the diagram above commutes for all choices of canonical objects, and let

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} <(\theta_1 E)_{\Downarrow}[\![\theta_2 r_2]\!], \theta_1 r_1>$$

be a $(\mathcal{R}|\mathcal{E}'|\mathcal{R})$-critical pair.

Let $M = (\theta_1 E)_{\Downarrow}[\![\theta_2 l_2]\!]$, we have

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}_{\mathcal{E}'\beta\eta}} (\theta_1 E)_{\Downarrow}[\![\theta_2 r_2]\!] \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}_{\mathcal{E}'\beta\eta}} \theta_1 r_1$$

By the assumptions, we find objects N_i' such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} (\theta_1 E)_{\Downarrow}[\![\theta_2 r_2]\!] \to^*_{\mathcal{R}/\mathcal{E}'\beta\eta} N_1' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_1 r_1 \to^*_{\mathcal{R}/\mathcal{E}'\beta\eta} N_2' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_1' =_{\mathcal{E}\beta\eta} N_2'$$

so we conclude that the given critical pair is trivial.

Conversely, assume that all $(\mathcal{R}|\mathcal{E}'|\mathcal{R})$-critical pairs are trivial, and let M, N_1, N_2 as in the statement. By unraveling the definitions, we get

$$\begin{array}{ll}
\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx_{\mathcal{E}'} E^{(i)}[\![M_\circ^{(i)}]\!] & \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} (N_i)_{\Downarrow} \approx_{\mathcal{E}'} E^{(i)}[\![N_\circ^{(i)}]\!] \\
\Gamma_\circ^{(i)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ^{(i)} \Downarrow \theta_i C_\circ^{(i)} & \Gamma_\circ^{(i)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_\circ^{(i)} \Downarrow \theta_i C_\circ^{(i)} \\
\Gamma_\circ^{(i)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ^{(i)} =_{\mathcal{E}'\beta\eta} \theta_i l_i & \Gamma_\circ^{(i)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_\circ^{(i)} =_{\mathcal{E}'\beta\eta} \theta_i r_i \\
\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E^{(i)}[\![\Gamma_\circ^{(i)} \vdash \circ : \theta_i C_\circ^{(i)}]\!] \Downarrow A & \Gamma_\circ^{(i)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_i : \Delta_\circ^{(i)} \\
\Delta_\circ^{(i)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} l_i \to r_i : C_\circ^{(i)} \in \mathcal{R} &
\end{array}$$

for $i = 1, 2$.

From the assumption $\mathcal{R} \not\preceq^{\mathrm{t}} \mathcal{E}'$ it follows $\theta_i C_\circ^{(i)} \not\preceq^{\mathrm{t}} \mathcal{E}'$, so we can apply Proposition 7.1.3 obtaining environments E^i and objects $M_\circ^i$ such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} E^{(i)} \approx_{\mathcal{E}'} E^i \qquad \Gamma_\circ^{(i)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ^{(i)} \approx_{\mathcal{E}'} M_\circ^i$$

and $M_{\Downarrow} = E^i[\![M_\circ^i]\!]$.

Since, by Propositions 4.3.2 and 6.3.16,

$$\Gamma_\circ^{(i)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ^i =_{\mathcal{E}'\beta\eta} \theta_i l_i \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} (N_i)_{\Downarrow} \approx_{\mathcal{E}'} E^i[\![N_\circ^{(i)}]\!]$$

we will assume, without loss of generality, $E^{(i)} = E^i$ and $M_\circ^{(i)} = M_\circ^i$, i.e. $M_{\Downarrow} = E^{(i)}[\![M_\circ^{(i)}]\!]$.

If $E^{(1)} \uparrow E^{(2)}$, then, by Proposition 6.1.9 and Theorem 6.3.14, we easily show

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_i \to_{\mathcal{R}/\mathcal{E}'\beta\eta} N_i' \ (i = 1, 2)$$

where $N_1' = N_2' = (E^{(1)})_{E^{(2)}[\![N_\circ^{(2)}]\!]}[\![N_\circ^{(1)}]\!]$.

Otherwise we can assume, without loss of generality, $E^{(2)} = (E^{(1)} \cdot E')$ for some well-formed environment E'. Hence we have

$$\Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_\circ^{(1)} =_{\mathcal{E}'\beta\eta} E'[\![M_\circ^{(2)}]\!]$$

and it will suffice to prove the theorem for $N_1 = N_\circ^{(1)}$ and $N_2 = E'[\![N_\circ^{(2)}]\!]$.

By transitivity,

$$\Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_1 l_1 =_{\mathcal{E}'\beta\eta} E'[\![M_\circ^{(2)}]\!]$$

so we can find an environment E satisfying the statement of Lemma 7.3.1.

Assume E satisfies case (1) of the statement; then

$$\Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} E' \approx_{\mathcal{E}'} (\theta_1 E)_{\Downarrow}$$

and therefore, by Propositions 6.1.4 and 6.3.16,

$$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1 l_1 =_{\mathcal{E}'\beta\eta} (\theta_1 E)_{\Downarrow}[\![\theta_2 l_2]\!]$$

Hence $\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} < (\theta_1 E)_{\Downarrow}[\![\theta_2 l_2]\!], \theta_1 r_1 >$ is a $(\mathcal{R}|\mathcal{E}'|\mathcal{R})$-critical pair; by the assumptions, it must be trivial, so there are N_1' and N_2' such that

$$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} (\theta_1 E)_{\Downarrow}[\![\theta_2 l_2]\!] \rightarrow^*_{\mathcal{R}/\mathcal{E}'\beta\eta} N_1' \qquad\qquad \Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1 r_1 \rightarrow^*_{\mathcal{R}/\mathcal{E}'\beta\eta} N_2'$$

and $\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1' =_{\mathcal{E}\beta\eta} N_2'$.

Since, using again Propositions 6.1.4 and 6.3.16, we can show

$$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} E'[\![N_\circ^{(2)}]\!] =_{\mathcal{E}'\beta\eta} (\theta_1 E)_{\Downarrow}[\![\theta_2 r_2]\!] \qquad\qquad \Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} N_\circ^{(1)} =_{\mathcal{E}'\beta\eta} \theta_1 r_1$$

the result follows by invariance of $\rightarrow_{\mathcal{R}/\mathcal{E}'\beta\eta}$ with respect to E'- and $\beta\eta$-conversion.

The remaining possibility is that E satisfies part (2) of the statement of Lemma 7.3.1. In this case $l_1 = E[\![U]\!]$, where $U = x\ U_1\ U_2 \ldots U_k$ is a generalized variable, and there is an environment $E_\circ$ such that

$$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} E' \approx_{\mathcal{E}} ((\theta_1 E)_{\Downarrow} \cdot E_\circ)$$
$$\Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1(x) =_{\mathcal{E}\beta\eta} \lambda y_1 : A_1 \ldots \lambda y_k : A_k.\ E_\circ[\![M_\circ^{(2)}]\!]$$

Let $\Delta_\circ^{(1)}(x) = B$, applying Proposition 6.1.3 to the (well-typed) environment $E_\circ$, we conclude $C_\circ^{(2)} \preceq^{\mathrm{t}} B$. From this we immediately see $B \not\prec^{\tau} C_\circ^{(1)}$: otherwise, we would have

$$C_\circ^{(2)} \preceq^{\mathrm{t}} B \prec^{\tau} C_\circ^{(1)}$$

implying $C_\circ^{(2)} \prec^{\tau} C_\circ^{(1)}$, a contradiction to the assumption $\mathcal{R} \not\prec^{\tau} \mathcal{R}$.

Let $l_1 = E_{U'}[\![U']\!]$ where $U' = x\ U_1'\ U_2' \ldots U_k'$ is any other occurrence of a generalized variable (with the same underlying variable x as U), we apply Proposition 6.1.6 to conclude

$$\Delta_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} E_{U'}[\![\Delta_\circ(l_1, U') \vdash \circ : type(\Delta_\circ(l_1, U'))]\!] : C_\circ^{(1)}$$

Moreover, all these environments $E_{U'}$ are disjoint, and $\Delta_\circ^{(1)}$-stable by Proposition 7.2.3.

By Corollary 7.2.5 and Proposition 6.1.6, we see that

$$\Delta_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} ((\theta_1 E_{U'})_{\Downarrow} \cdot E_\circ)[\![\Gamma_\circ^{(2)} \vdash \circ : \theta_2 C_\circ^{(2)}]\!] \Downarrow \theta_1 C_\circ^{(1)}$$

Using repeatedly Proposition 6.1.9, we show

$$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} E'[\![\theta_2 r_2]\!] \rightarrow^*_{\mathcal{R}/\beta\eta} \theta_1' l_1 \rightarrow_{\mathcal{R}} \theta_1' r_1$$

where $\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1' : \Delta_\circ^{(1)}$ is defined as:

$$\theta_1'(z) = \begin{cases} \theta_1(z) & z \neq x \\ \lambda y_1 : A_1 \ldots y_k : A_k.\ E_\circ[\![N_\circ^{(2)}]\!] & z = x \end{cases}$$

A similar argument, applied to r_1 instead of l_1, shows

$$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1 r \rightarrow^*_{\mathcal{R}/\beta\eta} \theta_1' r$$

from which we see that $N_1' = N_2' = \theta_1' r$ are as required. $\square$

Theorem 7.3.4 (Critical Pair Criterion for Simpler Equational Theories).
Let $\rightarrow_{\mathcal{R}/\mathcal{E}\beta\eta}$ be strongly normalizing, with $\mathcal{E}$ simpler than $\mathcal{R}$. Then all $(\mathcal{R}|\mathcal{E}|\mathcal{R})$-critical pairs are trivial iff rewriting modulo $\mathcal{E}$ is Church-Rosser, i.e.

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \rightarrow^{*}_{\mathcal{R}/\mathcal{E}\beta\eta} M_i \ (i = 1, 2) \textit{ implies } \exists N \textit{ s. t. } M_i \rightarrow^{*}_{\mathcal{R}/\mathcal{E}\beta\eta} N \ (i = 1, 2)$$

for all objects M, M_1, and M_2; or, in pictures, iff the following diagram commutes:

$$\begin{tikzcd}
M \arrow[r, "(\mathcal{R}/\mathcal{E}\beta\eta)^*"] \arrow[d, "(\mathcal{R}/\mathcal{E}\beta\eta)^*"'] & M_1 \arrow[d, dashed, "(\mathcal{R}/\mathcal{E}\beta\eta)^*"] \\
M_2 \arrow[r, dashed, "(\mathcal{R}/\mathcal{E}\beta\eta)^*"'] & N
\end{tikzcd}$$

Proof. Assume $\rightarrow_{\mathcal{E}\beta\eta}$ is Church-Rosser, then all the $(\mathcal{R}|\mathcal{E}|\mathcal{R})$-critical pairs are trivial by Lemma 7.3.3 and Proposition 6.3.6.

Conversely, assume all $(\mathcal{R}|\mathcal{E}|\mathcal{R})$-critical pairs are trivial; by Theorem 6.3.14, it suffices to prove that $\rightarrow_{\mathcal{R}_{\mathcal{E}\beta\eta}}$ is Church-Rosser. From the assumptions and Corollary 6.3.7 it follows that $\rightarrow_{\mathcal{R}_{\mathcal{E}\beta\eta}}$ is strongly normalizing. Hence, by Newman's Lemma, it suffices to show that canonical rewriting is locally Church-Rosser, i.e. that the diagram

$$\begin{tikzcd}
M \arrow[r, "\mathcal{R}_{\mathcal{E}\beta\eta}"] \arrow[d, "\mathcal{R}_{\mathcal{E}\beta\eta}"'] & M' \arrow[d, dashed, "\mathcal{R}^*_{\mathcal{E}\beta\eta}"] \\
M'' \arrow[r, dashed, "\mathcal{R}^*_{\mathcal{E}\beta\eta}"'] & N
\end{tikzcd}$$

commutes. We use again Theorem 6.3.14 again to transform the problem to the diagram

$$\begin{tikzcd}
M \arrow[r, "\mathcal{R}_{\mathcal{E}\beta\eta}"] \arrow[d, "\mathcal{R}_{\mathcal{E}\beta\eta}"'] & N_1 \arrow[d, dashed, "(\mathcal{R}/\mathcal{E}\beta\eta)^*"] \\
N_2 \arrow[r, dashed, "(\mathcal{R}/\mathcal{E}\beta\eta)^*"'] & U
\end{tikzcd}$$

whose commutativity is ensured, under the current assumptions, by Lemma 7.3.3 □

7.4 Internal Rewriting

From the proof of 7.3.3 we can extract a result that may simplify the task of determining if a given object is a $(\rightarrow_{\mathcal{R}_{\mathcal{E}\beta\eta}})$-normal form. We introduce first the following definition:

Definition 7.4.1. We say that M *rewrites internally* modulo $\mathcal{E}$ to N under $\mathcal{R}$, and write

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{R}[\mathcal{E}]} N$$

if and only if

$$\begin{array}{ll} M_{\Downarrow} = E[\![M_\circ]\!] & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_{\Downarrow} \approx E[\![N_\circ]\!] \\ \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \Downarrow \theta C_\circ & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} N_\circ \Downarrow \theta C_\circ \\ \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ =_{\mathcal{E}\beta\eta} \theta l & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} N_\circ =_{\beta\eta} \theta r \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : \theta C_\circ]\!] \Downarrow A & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ \\ \Delta_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} l \to r : C_\circ \in \mathcal{R} & \end{array}$$

Remark. From the definition, it is easy to see that $(\to_{\mathcal{R}[\mathcal{E}]})$-rewriting involves only $\mathcal{E}$-pattern matching with the LHS of a rule, instead of $\mathcal{E}$-conversion of the entire object, as in the case of $\to_{\mathcal{R}_{\mathcal{E}\beta\eta}}$ and $\to_{\mathcal{R}/\mathcal{E}\beta\eta}$.

Lemma 7.4.2. *Let $\mathcal{E}$ be simpler then $\mathcal{R}$, then*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to^{+}_{\mathcal{R}_{\mathcal{E}\beta\eta}} N \text{ implies } \exists N' \text{ s.t. } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to^{+}_{\mathcal{R}[\mathcal{E}]} N' \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N' =_{\mathcal{E}\beta\eta} N$$

Proof. It will suffice to prove the statement for one-step canonical rewriting, i.e.

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}_{\mathcal{E}\beta\eta}} N_1$$

The result will then follow arguing by induction on the length of the rewriting sequence, and using invariance of canonical rewriting with respect to conversion.

By Definition 6.3.1,

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \to_{\mathcal{R}_{\mathcal{E}\beta\eta}} N_1$$

if and only if

$$\begin{array}{ll} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx_{\mathcal{E}} E[\![M_\circ]\!] & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (N_1)_{\Downarrow} \approx_{\mathcal{E}} E[\![N_\circ]\!] \\ \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \Downarrow \theta C_\circ & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} N_\circ \Downarrow \theta C_\circ \\ \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ =_{\mathcal{E}\beta\eta} \theta l & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} N_\circ =_{\mathcal{E}\beta\eta} \theta r \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E[\![\Gamma_\circ \vdash \circ : \theta C_\circ]\!] \Downarrow A & \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ \\ \Delta_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} l \to r : C_\circ \in \mathcal{R} & \end{array}$$

Using the assumption $\mathcal{R} \not\preceq^{\mathrm{t}} \mathcal{E}$, we conclude $\theta C_\circ \not\preceq^{\mathrm{t}} \mathcal{E}$. We can therefore apply Proposition 7.1.3 obtaining E' and $M'_\circ$ such that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} E \approx_{\mathcal{E}} E' \qquad \Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \approx_{\mathcal{E}} M'_\circ$$

and $M_{\Downarrow} = E'[\![M'_\circ]\!]$.

From Propositions 4.3.2 and 6.3.16 it follows

$$\Gamma^{(i)}_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ =_{\mathcal{E}\beta\eta} \theta l \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (N_1)_{\Downarrow} \approx_{\mathcal{E}} E'[\![(\theta r)_{\Downarrow}]\!]$$

so we see that, by letting $N'_1 = E'[\![(\theta r)_{\Downarrow}]\!]$, we have

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{R}[\mathcal{E}]} N'_1 \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 =_{E\beta\eta} N'_1$$

□

Theorem 7.4.3. *Let $\mathcal{E}$ simpler than $\mathcal{R}$, and assume $\to_{\mathcal{E}/\beta\eta}$ strongly normalizing. Then all the $\mathcal{R}|\mathcal{E}|\mathcal{R}$-critical pairs are trivial if and only if $\to_{\mathcal{R}[\mathcal{E}]}$ is confluent modulo $\mathcal{E}$- and $\beta\eta$-conversion, i.e.*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 =_{\mathcal{E}\beta\eta} M_2 \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i \to^{*}_{\mathcal{R}[\mathcal{E}]} M'_i \ (i=1,2) \text{ implies } \exists N_1, N_2 \text{ s. t.} \begin{cases} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_i \to^{*}_{\mathcal{R}[\mathcal{E}]} N_i \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 =_{\mathcal{E}\beta\eta} N_2 \end{cases}$$

for all objects M_i and M_i'.

Graphically, confluence of $\rightarrow_{\mathcal{R}[\mathcal{E}]}$ modulo $\mathcal{E}$- and $\beta\eta$-conversion corresponds to commutativity of the following diagram:

$$\begin{tikzcd}
 & M_1 \arrow[r, "\mathcal{R}[\mathcal{E}]^*"] \arrow[dl, dash, "\mathcal{E}\beta\eta"'] & M_1' \arrow[dd, dashed, "\mathcal{R}[\mathcal{E}]^*"] \\
M_2 \arrow[dd, "\mathcal{R}[\mathcal{E}]^*"'] & & \\
 & & N_1' \arrow[dl, dashed, dash, "\mathcal{E}\beta\eta"] \\
M_2' \arrow[r, dashed, "\mathcal{R}[\mathcal{E}]^*"'] & N_2 &
\end{tikzcd}$$

Proof. In light of Theorem 7.3.4, it suffices to show that $\rightarrow_{\mathcal{R}[\mathcal{E}]}$ is confluent modulo $\mathcal{E}$- and $\beta\eta$-conversion if and only if $\rightarrow_{\mathcal{R}/\mathcal{E}\beta\eta}$ is Church-Rosser.

Assume $\rightarrow_{\mathcal{R}[\mathcal{E}]}$ confluent modulo $\mathcal{E}$- and $\beta\eta$-conversion, and

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow^*_{\mathcal{R}/\mathcal{E}\beta\eta} M_i \ (i = 1, 2)$$

Using Theorem 6.3.14 and Lemma 7.4.2 we conclude that, for some objects M_i',

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow^*_{\mathcal{R}[\mathcal{E}]} M_i' \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i' =^*_{\mathcal{E}\beta\eta} M_i \ (i = 1, 2)$$

Hence there are objects N_i such that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i' \rightarrow^*_{\mathcal{R}[\mathcal{E}]} N_i \ (i = 1, 2) \text{ and } \ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 =_{\mathcal{E}\beta\eta} N_2$$

and this gives us the result, since it is readily verified that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i \rightarrow^*_{\mathcal{R}/\mathcal{E}\beta\eta} N_i \ (i = 1, 2)$$

Conversely, assume $\rightarrow_{\mathcal{R}/\mathcal{E}\beta\eta}$ is Church-Rosser, and furthermore

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 =_{\mathcal{E}\beta\eta} M_2 \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i \rightarrow^*_{\mathcal{R}[\mathcal{E}]} M_i' \ (i = 1, 2)$$

By definition,

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 \rightarrow^*_{\mathcal{R}/\mathcal{E}\beta\eta} M_i' \ (i = 1, 2)$$

hence there is an object N such that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i' \rightarrow^*_{\mathcal{R}/\mathcal{E}\beta\eta} N \ (i = 1, 2)$$

Using Theorem 6.3.14 and Lemma 7.4.2 again, we get objects N_1 and N_2 such that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i' \rightarrow^*_{\mathcal{R}[\mathcal{E}]} N_i \ (i = 1, 2) \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_i =_{\mathcal{E}\beta\eta} N \ (i = 1, 2)$$

which concludes the proof. □

7.5 Completion by Stages

In Definition 6.2.1 we define well-formedness of a HTRS $\mathcal{R}$ by the condition $\mathcal{R} \not\prec^{\tau} \mathcal{R}$, and we proceed developing our theory in Chapter 7 and 8 relying on the assumption that the HTRSs we deal with are well-formed. The reader will wonder how important is this assumption in our constructions, and if it is possible to relax it under some special circumstances. The answer is affirmative, and in what follows we will describe a technique by us named "completion by stages" to establish confluence of some non-well-formed HTRSs.

Let $\mathcal{R}$ be a strong normalizing HTRS such that $\mathcal{R} \not\prec^{\tau} \mathcal{R}$, assume $\mathcal{R}$ can be partitioned into the disjoint union of HTRSs $\mathcal{R}_i$:

$$\mathcal{R} = \mathcal{R}_1 \sqcup \mathcal{R}_2 \sqcup \cdots \sqcup \mathcal{R}_n$$

verifying the following properties:

1. $\mathcal{R}_i$ is well-formed, i.e. $\mathcal{R}_i \not\prec^{\tau} \mathcal{R}_i$;
2. $\mathcal{R}_i \not\preceq^{\mathrm{t}} \mathcal{R}_j$ for all $j < i$

For any such $\mathcal{R}$ we can establish confluence as follows. First, observe that each of the $\mathcal{R}_i$ is strong normalizing, since $\mathcal{R}$ is. We apply Theorem 7.4.3 to establish confluence of $\mathcal{R}_1$ modulo $\mathcal{E}_1 = \cdot$. Let $\mathcal{E}(\mathcal{R}_1)$ the equational theory generated by $\mathcal{R}_1$,

$$\mathcal{E}(\mathcal{R}_1) = \{\, \Gamma \vdash^{\prec}_{\Sigma} l = r : A \mid (\Gamma \vdash^{\prec}_{\Sigma} l \to r : A) \in \mathcal{R}_1 \}$$

it is easily seen to be well-formed. Moreover, equality of expressions modulo $\mathcal{E}(\mathcal{R}_1)$ is decidable (comparing normal forms). To establish confluence of $\mathcal{R}_1 \sqcup \mathcal{R}_2$ we apply again the Critical Pair Criterion, this time to R_2, and working modulo $\mathcal{E}_2 = \mathcal{E}(\mathcal{R}_1)$.

In general, define

$$\mathcal{R}_{<1} = \emptyset \qquad\qquad \mathcal{R}_{<(i+1)} = R_i \sqcup \mathcal{R}_{<i}$$

and

$$\mathcal{E}_i = \{\Gamma \vdash^{\prec}_{\Sigma} l = r : A \mid (\Gamma \vdash^{\prec}_{\Sigma} l = r : A) \in \mathcal{R}_{<i}\}$$

In the i-th step, we try to establish to establish confluence of $\mathcal{R}_{<(i+1)}$ under the assumption that $\mathcal{R}_{<i}$ has already proven confluent. We do so by using the Critical Pair Criterion given in Chapter 7 to establish confluence of $\mathcal{R}_{i+1}$ modulo $\mathcal{E}_i$.

What makes this technique applicable to a vast class of cases is the remark that the statement of Theorem 7.4.3 does not require any particular restriction on the syntactic structure of the equational theory, and specifically does not require both sides of the equations to be higher-order patterns. This is important, since very few interesting HTRSs in the literature happen to have pattern right-hand-sides.

Example 5. To give an idea of how frequent cases tackled by this technique are, the following example is taken from [2], with a few modifications to adapt it to a dependent type setting.

Consider the signature

$$\Sigma = \{\mathbf{nat}, \mathbf{0}, \mathrm{s}, \mathbf{double}, \mathbf{list}, \mathbf{nil}, \mathbf{cons}, \mathbf{map}, \mathbf{doublelist}\}$$

$$\begin{aligned}
&\Sigma(\mathbf{nat}) = \text{type}\\
&\Sigma(\mathbf{0}) = \mathbf{nat}\\
&\Sigma(\mathrm{s}) = \Pi n : \mathbf{nat}.\ \mathbf{nat}\\
&\Sigma(\mathbf{double}) = \Pi n : \mathbf{nat}.\ \mathbf{nat}\\
&\Sigma(\mathbf{list}) = \Pi n : \mathbf{nat}.\ \text{type}\\
&\Sigma(\mathbf{nil}) = \mathbf{list}\ \mathbf{0}\\
&\Sigma(\mathbf{cons}) = \Pi m : \mathbf{nat}.\ \Pi n : \mathbf{nat}.\ \Pi n : \mathbf{list}\ m.\ \mathbf{list}\ (\mathrm{s}\ m)\\
&\Sigma(\mathbf{map}) = \Pi m : \mathbf{nat}.\ \Pi f : (\Pi n : \mathbf{nat}.\ \mathbf{nat}).\ \Pi n : (\mathbf{list}\ m).\ \mathbf{list}\ m\\
&\Sigma(\mathbf{doublelist}) = \Pi m : \mathbf{nat}.\ \Pi n : (\mathbf{list}\ m).\ \mathbf{list}\ m
\end{aligned}$$

It can be easily verified that this signature is well-typed under the dependence relation

$$\prec^{\mathrm{t}} = \prec^{\tau} = \{(\mathbf{nat}, \mathbf{list})\}$$

We define on Σ the HTRS $\mathcal{R}$ with rules:

$$\left.\begin{array}{ll}\Gamma_0 & \vdash^{\prec}_{\Sigma} \mathbf{double}\ \mathbf{0} \rightarrow \mathbf{0} : \mathbf{nat} \\ \Gamma_1 & \vdash^{\prec}_{\Sigma} \mathbf{double}\ (\mathbf{s}\ x) \rightarrow \mathbf{s}\ (\mathbf{s}\ x) : \mathbf{nat}\end{array}\right\}\mathcal{R}_1$$

$$\left.\begin{array}{ll}\Gamma_2 & \vdash^{\prec}_{\Sigma} \mathbf{map}\ \mathbf{0}\ F\ \mathbf{nil} \rightarrow \mathbf{nil} : \mathbf{list}\ \mathbf{0} \\ \Gamma_3 & \vdash^{\prec}_{\Sigma} \mathbf{map}\ (\mathbf{s}\ m)\ F\ (\mathbf{cons}\ m\ y\ L) \rightarrow \mathbf{cons}\ m\ (Fy)\ (\mathbf{map}\ m\ F\ L) : \mathbf{list}\ (\mathbf{s}\ m) \\ \Gamma_4 & \vdash^{\prec}_{\Sigma} \mathbf{doublelist}\ m\ L \rightarrow \mathbf{map}\ m\ \mathbf{double}\ L\end{array}\right\}\mathcal{R}_2$$

where

$$\begin{array}{ll}\Gamma_0 = \cdot & \Gamma_3 = F : \Pi x : \mathbf{nat}.\ \mathbf{nat}, m : \mathbf{nat}, y : \mathbf{nat}, L : \mathbf{list}\ m \\ \Gamma_1 = x : \mathbf{nat} & \Gamma_4 = m : \mathbf{nat}, L : \mathbf{list}\ m \\ \Gamma_2 = F : \Pi x : \mathbf{nat}.\ \mathbf{nat} & \end{array}$$

This is proven strong normalizing using the techniques described in the cited paper. $\mathcal{R}$ is clearly not well-formed in the sense of Definition 6.2.1, but it can be decomposed into the disjoint sum of two well-formed HTRSs $\mathcal{R}_1$ and $\mathcal{R}_2$. Since $\mathcal{R}_2 \not\preceq^{\mathrm{t}} R_1$, $\mathcal{R}$ can be proved confluent using completion by stages.

7.6 Summary

The Critical Pair Criterion reduces the problem of establishing confluence of a strongly normalizing HTRS $\mathcal{R}$ to checking joinability of special pairs of objects called critical pairs. In the case of confluence modulo an equational theory $\mathcal{E}$, some of these pairs come, as we will see in detail in the next chapter, from overlapping rules in $\mathcal{R}$ with equations in $\mathcal{E}$. However, there are special circumstances under which examinations of these pairs is unneeded. In this chapter we studied a large class of problems of these kind, which could be characterized by the fact that $\mathcal{E}$ is defined on more primitive types than $\mathcal{R}$. This idea was made precise by the definition of simpler equational theory in Section 7.1.

In section 7.2, we introduced some technical tools used in the analysis of critical pairs. These focus on a generalization of the notion of higher-order patterns which is preserved by considering sub-objects. We used this machinery in Section 7.3 to prove our specialized version of the Critical Pair Lemma; we will see that it also plays an important role use in the development of the general version of the Critical Pair Criterion, described in the next chapter.

In Section 7.4 we presented the same results from a different perspective. We introduced the third and last rewriting notion of this dissertation; this is even more restrictive than canonical rewriting, and therefore more suitable to practical implementations. Most importantly, we showed that, in the case of simpler equational theories, these two are closely related, as we showed in Lemma 7.4.2.

Section 7.5 wrapped up the subject by presenting a technique, completion by stages, that is often useful to circumvent some of the restrictions we imposed in Definition 6.2.1 on the constructions of well-formed HTRSs.

Chapter 8

Confluence for General Equational Theories

In this chapter we will deal with the general case, when the type classes of the equational theory $\mathcal{E}$ could be the same as the classes of the HTRS $\mathcal{R}$ being considered. As we will see, we will still be able to prove similar results to those presented in Chapter 7, provided that additional critical pairs, coming from the interaction between the rewriting system $\mathcal{R}$ and the equational theory $\mathcal{E}$, are considered. In doing this, our strategy will consist in treating the equational theory as a bi-directional HTRS. Some further assumptions will also be needed:

Notation. Throughout this chapter, we will assume the following on the equational theories $\mathcal{E}$ and HTRSs $\mathcal{R}$ considered:

- for each equation $\Gamma_\circ \vdash^{\prec}_{\Sigma} M_\circ = N_\circ : A_\circ$, both objects $M_\circ$ and $N_\circ$ are (higher-order) patterns;
- we have $\mathcal{R} \not\prec^{\tau} \mathcal{E}$.

These restrictions can be seen as the analogues of those we formulated for HTRSs in Definition 6.2.3.

8.1 One-Step $\mathcal{E}$-Conversion

We start by introducing a different, more "rewrite-like", formulation of congruence modulo $\beta\eta$- and $\mathcal{E}$-conversion. The relation $=_{\mathcal{E}\beta\eta}$ introduced in chapter 2 is ill-suited for our purposes, as it is intrinsically transitive, and therefore capable of performing several equational steps in parallel. We will need a more fine-grained notion, where only one $\mathcal{E}$-conversion step is performed at a time. This motivates the following:

Definition 8.1.1. We define the judgments

$$\begin{array}{ll} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{E}} M' & M\ \mathcal{E}\text{-converts to } M' \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \to_{\mathcal{E}} A' & A\ \mathcal{E}\text{-converts to } A' \end{array}$$

derived according to the set of rules listed in Figure 8.1

As usual, we will start by listing a few basic properties of this new judgment. The proofs, unless stated otherwise, will be obtained by a simple induction on the derivations.

Proposition 8.1.2.

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{E}} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx_{\mathcal{E}} M'$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \to_{\mathcal{E}} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \approx_{\mathcal{E}} A'$$

$$\frac{(\Delta \vdash^{\prec}_{\Sigma} M_\circ = N_\circ : A_\circ) \in \mathcal{E} \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\beta\eta} \theta M_\circ \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N =_{\beta\eta} \theta N_\circ \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \Downarrow \theta A_\circ}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} N}$$

$$\frac{(\Delta \vdash^{\prec}_{\Sigma} N_\circ = M_\circ : A_\circ) \in \mathcal{E} \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\beta\eta} \theta M_\circ \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N =_{\beta\eta} \theta N_\circ \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \Downarrow \theta A_\circ}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} N}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow_{\mathcal{E}} A' \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} M : B}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M \rightarrow_{\mathcal{E}} \lambda x : A'.\ M}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : A.\ M \rightarrow_{\mathcal{E}} \lambda x : A.\ M'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M' \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : A}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N \rightarrow_{\mathcal{E}} M'\ N} \qquad \frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \rightarrow_{\mathcal{E}} N'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M\ N \rightarrow_{\mathcal{E}} M\ N'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow_{\mathcal{E}} A' \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} B : \mathrm{type}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ B \rightarrow_{\mathcal{E}} \Pi x : A'.\ B}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : \mathrm{type} \quad \Gamma, x : A \vdash^{\prec}_{\Sigma,\mathcal{E}} B \rightarrow_{\mathcal{E}} B'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \Pi x : A.\ M \rightarrow_{\mathcal{E}} \Pi x : A.\ M'}$$

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \quad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A\ M \rightarrow_{\mathcal{E}} A\ M'}$$

Figure 8.1: One-step $\mathcal{E}$-conversion

Proposition 8.1.3 (Weakening). *Let* $\prec \subseteq \prec'$, $\Sigma \subseteq \Sigma'$, $\mathcal{E} \subseteq \mathcal{E}'$, , $\Gamma \subseteq \Gamma'$,

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M' \text{ implies } \Gamma' \vdash^{\prec'}_{\Sigma',\mathcal{E}'} M \rightarrow_{\mathcal{E}'} M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow_{\mathcal{E}} A' \text{ implies } \Gamma' \vdash^{\prec'}_{\Sigma',\mathcal{E}'} A \rightarrow_{\mathcal{E}'} A'$$

Proposition 8.1.4. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} C =_{\mathcal{E}\beta\eta} C' : \mathrm{type}$,

$$\Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M' \text{ implies } \Gamma, x : C', \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M'$$
$$\Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow_{\mathcal{E}} A' \text{ implies } \Gamma, x : C', \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow_{\mathcal{E}} A'$$

Proposition 8.1.5. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : C$,

$$\Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M' \text{ implies } \exists U', U'' \text{ s.t. } \begin{cases} \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]M \rightarrow^{*}_{\mathcal{E}} U'' \\ \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]M' \Rightarrow^{*}_{\beta} U' \\ \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} U' \Rightarrow^{*}_{\eta} U'' \end{cases}$$

$$\Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow_{\mathcal{E}} A' \text{ implies } \exists D', D'' \text{ s.t. } \begin{cases} \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]A \rightarrow^{*}_{\mathcal{E}} D'' \\ \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]A' \Rightarrow^{*}_{\beta} D' \\ \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} D' \Rightarrow^{*}_{\eta} D'' \end{cases}$$

provided that M *and* A *are well-formed expressions.*

Moreover, $|\beta(U')| = |\beta(U'')|$ *and* $|\beta(D')| = |\beta(D'')|$.

Proof. By induction on the derivations of $\Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M'$ and $\Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow_{\mathcal{E}} A'$. The only non-trivial case is given by the $\mathcal{E}$-conversion step:

- Case

$$\frac{\begin{array}{lll}(\Delta \vdash^{\prec}_{\Sigma} M_\circ = M'_\circ : A_\circ) \in \mathcal{E} & \Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_\circ : \Delta & \\ \Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\beta\eta} \theta_\circ M_\circ & \Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M' =_{\beta\eta} \theta_\circ M'_\circ & \Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \Downarrow \theta_\circ A_\circ\end{array}}{\Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M'}$$

Let $\theta = (id_\Gamma, x \mapsto N, id_{\Gamma'})$, under our assumptions $\Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma} \theta : \Gamma, x : C, \Gamma'$.
Using Proposition 2.2.2

$$\begin{array}{l}\Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma} [N/x]M =_{\beta\eta} (\theta \cdot \theta_\circ)M_\circ \\ \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma} [N/x]M' =_{\beta\eta} (\theta \cdot \theta_\circ)M'_\circ \\ \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma} (\theta \cdot \theta_\circ) : \Delta\end{array}$$

By Lemma 4.2.3,

$$\Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma} \theta M' \Rightarrow^{*}_{\beta\eta} (\theta M')_{\Downarrow} \qquad\qquad \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma} (\theta M')_{\Downarrow} \Downarrow (\theta \cdot \theta_\circ)A_\circ$$

and we can furthermore assume that $U'' = (\theta M')_{\Downarrow}$ is obtained by first β-reducing $\theta M' = [N/x]M'$ to its normal form U', and then η-expanding U'.
Hence, using Proposition 4.1.7,

$$\begin{array}{l}\Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma} [N/x]M' \Rightarrow^{*}_{\beta} U' \Rightarrow^{*}_{\eta} U'' \\ \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma} [N/x]M \rightarrow^{*}_{\mathcal{E}} U''\end{array}$$

□

Proposition 8.1.6. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N : C$ *and* $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \rightarrow_{\mathcal{E}} N'$

$$\begin{array}{c}\Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ implies } \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]M \rightarrow^{*}_{\mathcal{E}} [N'/x]M \\ \Gamma, x : C, \Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ implies } \Gamma, [N/x]\Gamma' \vdash^{\prec}_{\Sigma,\mathcal{E}} [N/x]A \rightarrow^{*}_{\mathcal{E}} [N'/x]A\end{array}$$

The notion of $\mathcal{E}$-conversion given by Definition 8.1.1 is not, in general, symmetric. It is, however, when considered modulo $\beta\eta$-conversion:

Proposition 8.1.7.

$$\begin{array}{c}\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \rightarrow_{\mathcal{E}/\beta\eta} M \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow_{\mathcal{E}} A' \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' \rightarrow_{\mathcal{E}/\beta\eta} A\end{array}$$

Proof. By induction on the derivations of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M'$ and $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A : K$. The only interesting case is:

- Case

$$\frac{\begin{array}{lll}(\Delta \vdash^{\prec}_{\Sigma} N_\circ = M_\circ : A_\circ) \in \mathcal{E} & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta & \\ \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\beta\eta} \theta M_\circ & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N =_{\beta\eta} \theta N_\circ & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \Downarrow \theta A_\circ\end{array}}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} N}$$

By Lemma 4.2.3, we obtain

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \Downarrow \theta A_\circ \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} =_{\beta\eta} M$$

and therefore $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} =_{\beta\eta} \theta M_\circ$, so we see that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N \rightarrow_{\mathcal{E}} M_{\Downarrow} \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} =_{\beta\eta} M$$

□

Corollary 8.1.8. *We have*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow^{*}_{\mathcal{E}/\beta\eta} M' \textit{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \rightarrow^{*}_{\mathcal{E}/\beta\eta} M$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow^{*}_{\mathcal{E}/\beta\eta} A' \textit{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' \rightarrow^{*}_{\mathcal{E}/\beta\eta} A$$

provided that M and A are well-formed expressions.

Proof. By induction on the length of the conversion sequences, using Proposition 8.1.7. □

Theorem 8.1.9. *The following holds for all well-formed expressions M and A*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}\beta\eta} M' \textit{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow^{*}_{\mathcal{E}/\beta\eta} M'$$
$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A' \textit{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \rightarrow^{*}_{\mathcal{E}/\beta\eta} A'$$

Proof. One direction is given by Propositions 8.1.2 and 4.3.2; the other is obtained by induction on the derivations of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}\beta\eta} M'$ and $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A =_{\mathcal{E}\beta\eta} A'$. We examine the most interesting cases:

- Case

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' =_{\mathcal{E}\beta\eta} M}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\mathcal{E}\beta\eta} M'}$$

  By inductive hypothesis we get $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \rightarrow^{*}_{\mathcal{E}/\beta\eta} M$, and the result follows by Corollary 8.1.8.

- Case

$$\frac{\Delta \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ = M'_\circ : A \in \mathcal{E} \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta M_\circ =_{\mathcal{E}\beta\eta} \theta M'_\circ}$$

  By Lemma 4.2.3 we have

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\theta M'_\circ)_{\Downarrow} =_{\beta\eta} \theta M'_\circ \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\theta M'_\circ)_{\Downarrow} \Downarrow \theta A$$

  and from this the result follows, since

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta M_\circ \rightarrow_{\mathcal{E}} (\theta M'_\circ)_{\Downarrow}$$

□

8.2 Canonical One-Step $\mathcal{E}$-Conversion

While the alternative formulation of $=_{\mathcal{E}\beta\eta}$ as reflexive transitive closure of $\rightarrow_{\mathcal{E}/\beta\eta}$ is more convenient, still it is far from ideal, since $\beta\eta$-conversion gets in the way, making its use quite cumbersome. For this reason we introduce yet another notion, which overcome these difficulties by applying $\mathcal{E}$-conversion only to objects in canonical form:

Definition 8.2.1. We say that the well-typed object M *$\mathcal{E}$-converts canonically modulo $\beta\eta$*, and write

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}_{\beta\eta}} N$$

provided that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx M' \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \rightarrow_{\mathcal{E}} N' \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_{\Downarrow} \approx N'$$

for some objects M' and N' such that $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \approx N'$ does not hold.

The similarity in notation between canonical $\mathcal{E}$-conversion $\rightarrow_{\mathcal{E}_{\beta\eta}}$ and canonical rewriting $\rightarrow_{\mathcal{R}_{\beta\eta}}$ (see Definition 6.3.1) is far from being accidental. As a matter of fact, both relations serve similar purposes, and share most properties.

Proposition 8.2.2. *The following holds:*

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \rightarrow_{\mathcal{E}_{\beta\eta}} N \textit{ implies } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \rightarrow_{\mathcal{E}/\beta\eta} N$$

Proof. It follows immediately from Definition 8.2.1, Lemma 4.2.3, and Proposition 4.3.2, since

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M =_{\beta\eta} M' \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M' \rightarrow_{\mathcal{E}} N' \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} N' =_{\beta\eta} N$$

□

To prove that $\rightarrow^*_{\mathcal{E}/\beta\eta}$ implies $\rightarrow^*_{\mathcal{E}_{\beta\eta}}$, we will follow the same outline used in Chapter 6 for $\rightarrow_{\mathcal{R}/\mathcal{E}\beta\eta}$ and $\rightarrow_{\mathcal{R}_{\mathcal{E}\beta\eta}}$, i.e. we will show first commutativity of $\rightarrow_{\mathcal{E}}$ with respect to β-reduction and η-expansion.

Lemma 8.2.3.

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \rightarrow_{\mathcal{E}} M' \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \Rightarrow_{\eta} M'' \textit{ implies } \exists N \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M' \Rightarrow^{=}_{\eta} N \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M'' \rightarrow_{\mathcal{E}} N$$
$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \rightarrow_{\mathcal{E}} A' \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \Rightarrow_{\eta} A'' \textit{ implies } \exists B \textit{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A' \Rightarrow^{=}_{\eta} B \textit{ and } \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A'' \rightarrow_{\mathcal{E}} B$$

provided that M and A are well-typed expressions.

In pictures, the following diagrams commute:

$$\begin{tikzcd} M \arrow[r, "\mathcal{E}"] \arrow[d, Rightarrow, "\eta"'] & M' \arrow[d, Rightarrow, "\eta^{=}"] \\ M'' \arrow[r, dashed, "\mathcal{E}"'] & N \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "\mathcal{E}"] \arrow[d, Rightarrow, "\eta"'] & A' \arrow[d, Rightarrow, "\eta^{=}"] \\ A'' \arrow[r, dashed, "\mathcal{E}"'] & B \end{tikzcd}$$

Proof. By induction on the derivations of $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \rightarrow_{\mathcal{E}} M'$ and $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} A \rightarrow_{\mathcal{E}} A'$. Most cases are trivial, as they follow through from the inductive hypothesis. Two interesting ones are:

- Case

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1 \rightarrow_{\mathcal{E}} M_1' \qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_2 : B}{\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1\ M_2 \rightarrow_{\mathcal{E}} M_1'\ M_2}$$

  If $\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M \stackrel{\mathcal{I}}{\Rightarrow}_{\eta} M''$, then either $M'' = M_1''\ M_2$, with

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_1 \Rightarrow_{\eta} M_1''$$

  in which case the result follows by inductive hypothesis, or $M'' = M_1\ M_2''$, with

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M_2 \Rightarrow_{\eta} M_2''$$

  and $N = M_1'\ M_2''$ is as required.

  Otherwise, we must have $M'' = \lambda x : B.\ M\ x$, where

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} M : \Pi x : B.\ C \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\varepsilon} B \Downarrow \mathrm{type}$$

  Using this assumptions, we can construct a derivation of

$$\Gamma \vdash^{\prec}_{\Sigma,\varepsilon} \lambda x : B.\ M\ x \rightarrow_{\mathcal{E}} \lambda x : B.\ M'\ x$$

  thus the result follows letting $N = \lambda x : B.\ M'\ x$.

- Case

$$\frac{(\Delta \vdash_{\Sigma}^{\prec} M_{\circ} = N_{\circ} : C) \in \mathcal{E} \quad \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} \theta : \Delta \quad \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M =_{\beta\eta} \theta M_{\circ} \quad \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M' =_{\beta\eta} \theta N_{\circ} \quad \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M' \Downarrow \theta C}{\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M \rightarrow_{\mathcal{E}} M'}$$

We have $\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M'' =_{\beta\eta} M$, and by transitivity we get

$$\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M'' \rightarrow_{\mathcal{E}} M'$$

so we see that in this case $N = M'$.

$\square$

Corollary 8.2.4. *For all well-formed objects* $\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M : A$

$$\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M \rightarrow_{\mathcal{E}}^{*} M' \text{ and } \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M \Rightarrow_{\eta}^{*} M'' \text{ implies } \exists N \text{ s. t. } \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M' \Rightarrow_{\eta}^{*} N \text{ and } \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M'' \rightarrow_{\mathcal{E}}^{*} N$$

In pictures, the following diagram commutes:

$$\begin{tikzcd}
M \arrow[r, "\mathcal{E}^*"] \arrow[d, Rightarrow, "\eta^*"'] & M' \arrow[d, Rightarrow, dashed, "\eta^*"] \\
M'' \arrow[r, dashed, "\mathcal{E}^*"'] & N
\end{tikzcd}$$

Moreover, the rewrite sequence $\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M \rightarrow_{\mathcal{E}}^{*} M'$ *is of the same length as* $\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M'' \rightarrow_{\mathcal{E}}^{*} N$.

Proof. By a simple induction on the length of the η-expansion sequence, using Lemma 8.2.3. $\square$

Lemma 8.2.5. *We have:*

$$\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M \rightarrow_{\mathcal{E}} M' \text{ and } \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M \Rightarrow_{\beta} M'' \text{ implies } \exists N' \text{ and } N'' \text{ s. t. } \begin{cases} \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M' \Rightarrow_{\beta}^{*} N' \\ \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M'' \rightarrow_{\mathcal{E}}^{*} N'' \\ \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} N' \Rightarrow_{\eta}^{*} N'' \end{cases}$$

$$\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} A \rightarrow_{\mathcal{E}} A' \text{ and } \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} A \Rightarrow_{\beta} A'' \text{ implies } \exists B' \text{ and } B'' \text{ s. t. } \begin{cases} \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} A' \Rightarrow_{\beta}^{*} B' \\ \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} A'' \rightarrow_{\mathcal{E}}^{*} B'' \\ \Gamma \vdash_{\Sigma,\varepsilon}^{\prec} B' \Rightarrow_{\eta}^{*} B'' \end{cases}$$

In pictures, the following diagrams commute:

$$\begin{tikzcd}
M \arrow[rr, "\mathcal{E}"] \arrow[dd, Rightarrow, "\beta"'] & & M' \arrow[d, Rightarrow, dashed, "\beta^*"] \\
& & N' \arrow[dl, Rightarrow, dashed, "\eta^*"] \\
M'' \arrow[r, dashed, "\mathcal{E}^*"'] & N'' &
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[rr, "\mathcal{E}"] \arrow[dd, Rightarrow, "\beta"'] & & A' \arrow[d, Rightarrow, dashed, "\beta^*"] \\
& & B' \arrow[dl, Rightarrow, dashed, "\eta^*"] \\
A'' \arrow[r, dashed, "\mathcal{E}^*"'] & B'' &
\end{tikzcd}$$

Moreover, $|\beta(N')| = |\beta(N'')|$ *and* $|\beta(B')| = |\beta(B'')|$.

Proof. By induction on the structure of the derivations of $\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} M \rightarrow_{\mathcal{E}} M'$ and $\Gamma \vdash_{\Sigma,\varepsilon}^{\prec} A \rightarrow_{\mathcal{E}} A'$. We analyze in detail a few representative cases:

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 \to_{\mathcal{E}} M_1' \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_2 : C''}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1\ M_2 \to_{\mathcal{E}} M_1'\ M_2}$$

  If

$$M = M_1\ M_2 \to_\beta M_1\ M_2'' = M''$$

  then the result follows trivially, letting $N' = N'' = M_1'\ M_2''$.
  Otherwise, if

$$M = M_1\ M_2 \to_\beta M_1''\ M_2 = M''$$

  the result follows by applying the inductive hypothesis to M_1' and M_1''.
  The only interesting case is therefore

$$M = M_1\ M_2 = (\lambda x : C.\ M_3)\ M_2 \to_\beta [M_2/x]M_3 = M''$$

  By inversion on the derivation of $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 \to_{\mathcal{E}} M_1'$ it follows that either $M_1' = \lambda x : C'.\ M_3$, with

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} C \to_{\mathcal{E}} C'$$

  or $M_1' = \lambda x : C.\ M_3'$, with

$$\Gamma, x : C' \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_3 \to_{\mathcal{E}} M_3'$$

  The former is easily handled by type conversion and Substitution. For the latter, from the derivation of well-typedness of M we get

$$\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} C =_{\mathcal{E}\beta\eta} C'$$

  so by type conversion $\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_2 : C'$, and we can apply Proposition 8.1.5, obtaining objects N' and N'' satisfying the statement:

$$\begin{array}{l}\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} [M_2/x]M_3 \Rightarrow^*_{\mathcal{E}} N'' \\ \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} (\lambda x : C'.\ M_3')\ M_2 \Rightarrow_\beta [M_2/x]M_3' \Rightarrow^*_\beta N' \\ \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} N' \Rightarrow^*_\eta N''\end{array}$$

- Case

$$\frac{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 : \Pi x : C.D \qquad \Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_2 \to_{\mathcal{E}} M_2'}{\Gamma \vdash^{\preceq}_{\Sigma,\mathcal{E}} M\ M_2 \to_{\mathcal{E}} M\ M_2'}$$

  As before, the cases

$$M = M_1\ M_2 \to_\beta M_1''\ M_2 = M''$$

  and

$$M = M_1\ M_2 \to_\beta M_1\ M_2'' = M''$$

  are easily handled.
  The remaining one is

$$M = (\lambda x : C'.\ M_3)\ M_2 \to_\beta [M_2/x]M_3 = M''$$

where, by inversion on the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (\lambda x : C'.\ M_3) : \Pi x : C.D$ (and possibly type conversion), we conclude

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_2 : C' \qquad\qquad \Gamma, x : C' \vdash^{\prec}_{\Sigma,\mathcal{E}} M_3 : D$$

We can therefore apply Proposition 8.1.6, concluding

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} [M_2/x]M_3 \Rightarrow^*_{\mathcal{E}} [M_2'/x]M_3$$

From this, the result follows by letting $N' = N'' = [M_2'/x]M_3$.

□

Corollary 8.2.6. *We have:*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow^*_{\mathcal{E}} M' \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Rightarrow^*_{\beta} M'' \text{ implies } \exists N \text{ s. t. } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \Rightarrow^*_{\beta\eta} N \text{ and } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M'' \rightarrow^*_{\mathcal{E}} N$$

Pictorially, the following diagram commutes:

$$\begin{tikzcd} M \arrow[r, "\mathcal{E}^*"] \arrow[d, Rightarrow, "\beta^*"'] & M' \arrow[d, Rightarrow, dashed, "\beta\eta^*"] \\ M'' \arrow[r, dashed, "\mathcal{E}^*"'] & N \end{tikzcd}$$

Proof. The proof is the same as the one of Corollary 6.3.12, where we replace $\rightarrow_{\mathcal{R}}$ by $\rightarrow_{\mathcal{E}}$, Lemma 6.3.11 by 8.2.5, and Corollary 6.3.10 by 8.2.4. □

Proposition 8.2.7. *Let* $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M'$, *then*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \Downarrow A \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \Downarrow A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \Downarrow \mathrm{type} \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' \Downarrow \mathrm{type}$$

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \downarrow A \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M' \downarrow A \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A \downarrow K \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} A' \downarrow K$$

Proof. By induction on the derivation of M and A. We look only at one case:

- Case

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B \Downarrow \mathrm{type} \qquad \Gamma, x : B \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 \Downarrow C}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : B.\ M_1 \Downarrow \Pi x : B.\ C}$$

We inspect the last inference used in the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \rightarrow_{\mathcal{E}} M'$. By type considerations, we see it cannot be neither one of the $\mathcal{E}$-conversion rules. So either $M' = \lambda x : B'.\ M_1$, with

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B \rightarrow_{\mathcal{E}} B' \qquad \Gamma, x : B \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 : C}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : B.\ M_1 \rightarrow_{\mathcal{E}} \lambda x : B'.\ M_1}$$

or $M' = \lambda x : B.\ M_1'$, with

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B : \mathrm{type} \qquad \Gamma, x : B \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 \rightarrow_{\mathcal{E}} M_1'}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : B.\ M_1 \rightarrow_{\mathcal{E}} \lambda x : B.\ M_1'}$$

Assume the former; by inductive hypothesis

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B' \Downarrow \mathrm{type}$$

By Propositions 8.1.2 and 4.3.2,

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B =_{\mathcal{E}\beta\eta} B'$$

so using Proposition 4.1.5,

$$\Gamma, x : B' \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 \Downarrow C$$

and hence the result.

□

Theorem 8.2.8. *We have*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to^{*}_{\mathcal{E}/\beta\eta} M' \text{ iff } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to^{*}_{\mathcal{E}_{\beta\eta}} M' \text{ or } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\beta\eta} M'$$

Proof. One direction is easily proven using Proposition 8.2.2. To prove the other, it suffices to show

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{E}} N \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to^{*}_{\mathcal{E}_{\beta\eta}} N$$

By invariance of $\to_{\mathcal{E}_{\beta\eta}}$ with respect to λ-conversion, we can assume that $M_{\Downarrow}$ is obtained by β-normalizing and then η-expanding. By using Corollaries 8.2.6 and 8.2.4, and Proposition 8.2.7 (in this order), we see

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{E}} N \text{ implies } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \to^{*}_{\mathcal{E}} N'$$

for some canonical N' such that $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N' \approx N_{\Downarrow}$. Hence, by Definition 8.2.1

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to^{*}_{\mathcal{E}_{\beta\eta}} N \text{ or } \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M =_{\beta\eta} N$$

□

The following results will allow us to give $\to_{\mathcal{E}}$ and $\to_{\mathcal{E}_{\beta\eta}}$ formulations that are closer in form to those given for $rewrite\mathcal{R}$ and $\to_{\mathcal{R}_{\beta\eta}}$ in Definitions 6.2.3 and 6.3.1.

Lemma 8.2.9. *Let M and M' well-formed, and such that $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \approx M'$ does not hold, then*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{E}} M'$$

if and only if there is an environment E such that

$$\begin{array}{ll} M = E[\![M_\circ]\!] & M' = E[\![M'_\circ]\!] \\ \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ =_{\beta\eta} \theta N_\circ & \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ =_{\beta\eta} \theta N'_\circ \\ \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} N'_\circ \Downarrow \theta A_\circ & \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ \end{array}$$

where $(\Delta \vdash^{\prec}_{\Sigma} N_\circ = N'_\circ : A_\circ) \in \mathcal{E}$ or $(\Delta \vdash^{\prec}_{\Sigma} N'_\circ = N_\circ : A_\circ) \in \mathcal{E}$.

Proof. Both directions are easily proved by induction on the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A$. One subtle point of this proof is that in the derivation of $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{E}} M'$ the rule

$$\frac{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} B \to_{\mathcal{E}} B' \qquad \Gamma, x : B \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 : C}{\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : B.\ M_1 \to_{\mathcal{E}} \lambda x : B'.\ M_1}$$

is never used, since from Propositions 8.1.2 and 4.3.2 it follows

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \lambda x : B.\ M_1 \approx \lambda x : B'.\ M_1$$

Hence the $\mathcal{E}$-conversion step cannot happen inside the type of one of the λ-abstracted variable, and therefore it can be isolated by an environment E. □

Remark. Note that the environment E in Lemma 8.2.9 needs not to be (and in most cases will not be) well-formed.

Corollary 8.2.10. *Let $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A$, then*

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M \to_{\mathcal{E}_{\beta\eta}} M'$$

if and only if M and M' are not pairwise $\beta\eta$-convertible and there is an environment E such that

$$\begin{array}{ll} \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_{\Downarrow} \approx E[\![M_\circ]\!] & \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} (M')_{\Downarrow} \approx E[\![M'_\circ]\!] \\ \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ \Downarrow \theta A_\circ & \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ \Downarrow \theta A_\circ \\ \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} M_\circ =_{\beta\eta} \theta N_\circ & \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} M'_\circ =_{\beta\eta} \theta N'_\circ \\ \Gamma(M, M_\circ) \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta : \Delta_\circ & \end{array}$$

where $(\Delta \vdash^{\prec}_{\Sigma} N_\circ = N'_\circ : A_\circ) \in \mathcal{E}$ or $(\Delta \vdash^{\prec}_{\Sigma} N'_\circ = N_\circ : A_\circ) \in \mathcal{E}$.

Proof. Immediate from Definition 8.2.1 and Corollary 8.2.10. □

8.3 Confluence for General Equational Theories

We will now introduce two new types of critical pairs, coming from overlapping rules and equations:

Definition 8.3.1 (Equational Critical Pairs). Let $\mathcal{R}$ be a HTRS, and

$$
\begin{array}{l}
(\Delta_1 \vdash^{\prec}_{\Sigma,\mathcal{E}} l_1 = r_1 : C_1) \in \mathcal{E} \text{ or } (\Delta_1 \vdash^{\prec}_{\Sigma,\mathcal{E}} r_1 = l_1 : C_1) \in \mathcal{E} \\
\Delta_2 \vdash^{\prec}_{\Sigma,\mathcal{E}} l_2 \to r_2 : C_2 \in \mathcal{R} \\
\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_i : \Delta_i \qquad (i = 1, 2)
\end{array}
$$

If there is a Δ_1-stable environment $E_1[\![\Gamma_\circ \vdash \circ : A_\circ]\!]$ not isolating a generalized variable, and such that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1 l_1 =_{\beta\eta} (\theta_1 E_1)_{\Downarrow}[\![\theta_2 l_2]\!]$$

then

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} < (\theta_1 E)_{\Downarrow}[\![\theta_2 r_2]\!], \theta_1 r_1 >$$

is a $(\mathcal{R}|\mathcal{E})$-*critical pair.*

Similarly, given a Δ_2-stable environment $E_2[\![\Gamma_\circ \vdash \circ : A_\circ]\!]$ that does not isolate a generalized variable, and

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_2 l_2 =_{\beta\eta} (\theta_2 E_2)_{\Downarrow}[\![\theta_1 l_1]\!]$$

then

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} < (\theta_2 E)_{\Downarrow}[\![\theta_1 r_1]\!], \theta_2 r_2 >$$

is called a $(\mathcal{E}|\mathcal{R})$-*critical pair.*

Let $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} < M_1, M_2 > : A$ be either a $(\mathcal{E}|\mathcal{R})$- or a $(\mathcal{R}|\mathcal{E})$-critical pair, it is said to be *trivial* if there are objects N_1 and N_2 such that

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i \to^{*}_{\mathcal{R}/\beta\eta} N_i \ (i = 1, 2) \qquad\qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 =_{\mathcal{E}\beta\eta} N_2$$

The key result of this chapter will be the following Lemma, due to G. Huet:

Lemma 8.3.2. *Let $\to_S$ be a symmetric relation, $=_S$ its reflexive-transitive closure, and $\to_R$ another relation such that $\to_{R/S}$ is strong normalizing. Then $\to_R$ is confluent modulo $=_S$, i.e.*

$$M_1 =_S M_2 \textit{ and } M_i \to^{*}_{R} M'_i \ (i = 1, 2) \textit{ implies } \exists N_1, N_2 \textit{ s.t. } M_i \to^{*}_{R} N_i \ (i = 1, 2) \textit{ and } N_1 =_S N_2$$

for all objects M_i and M'_i, if and only if the conditions (A) and (B) are satisfied:

(A) For all objects M, M_1, M_2,

$$M \to_R M_i \ (i = 1, 2) \textit{ implies } \exists N_1, N_2 \textit{ s.t. } M_i \to^{*}_{R} N_i \ (i = 1, 2) \textit{ and } N_1 =_S N_2$$

(B) For all objects M, M_1, M_2,

$$M \to_R M_1 \textit{ and } M \to_S M_2 \textit{ implies } \exists N_1, N_2 \textit{ s.t. } M_i \to^{*}_{R} N_i \ (i = 1, 2) \textit{ and } N_1 =_S N_2$$

Proof. A proof for a very general notion of rewriting, which encompasses ours, can be found in [29]. □

Confluence of $\to_R$ modulo $=_S$ is actually a stronger property than $\to_{R/S}$ being Church-Rosser, as the following shows:

Proposition 8.3.3. *Let $\to_S$ be a symmetric relation, and $=_S$ its reflexive, transitive closure. Furthermore, assume $\to_R$ is confluent modulo $=_S$ and $\to_{R/S}$ strong normalizing. Then $\to_{R/S}$ is Church-Rosser.*

Proof. By Newman's lemma, it suffices to show that $\to_{R/S}$ is locally Church-Rosser, i.e.

$$M \to_{R/S} N_i \ (i = 1, 2) \text{ implies } \exists N \text{ s.t. } N_i \to^*_{R/S} N \ (i = 1, 2)$$

or, in pictures:

$$\begin{tikzcd}
M \arrow[r, "R/S"] \arrow[d, "R/S"'] & N_1 \arrow[d, dashed, "(R/S)^*"] \\
N_2 \arrow[r, dashed, "(R/S)^*"'] & N
\end{tikzcd}$$

Then the proof is obtained by a simple diagram chasing:

$$\begin{tikzcd}
M \arrow[r, dash, "S"] \arrow[d, dash, "S"'] & M_1 \arrow[r, "R"] \arrow[dl, dash, "S"] & M_1' \arrow[r, dash, "S"] \arrow[d, dashed, "R^*"] & N_1 \arrow[ddd, dashed, "(R/S)^*"] \\
M_2 \arrow[d, "R"'] & & N_1' \arrow[dl, dashed, dash, "S"'] \arrow[ddr, dashed, dash, "S"] & \\
M_2' \arrow[r, dashed, "R^*"] \arrow[d, "S"'] & N_2' \arrow[drr, dashed, dash, "S"] & & \\
N_2 \arrow[rrr, dashed, "(R/S)^*"'] & & & N
\end{tikzcd}$$

The objects M_1, M_2, M_1', and M_2' are obtained by definition of $\to_{R/S}$; N_1' and N_2' by confluence of $\to_R$ modulo $=_S$; finally, we can $N = N_1'$ or $N = N_2'$ at will. $\square$

Lemma 8.3.4. *Let $\mathcal{R}$ be a HTRS, $\to_{\mathcal{R}_{\beta\eta}}$ and $\to_{\mathcal{E}_{\beta\eta}}$ satisfy condition (A) of Lemma 8.3.2 if and only if all the $(\mathcal{R}|\mathcal{R})$-critical pairs are trivial.*

Proof. It follows immediately from Lemma 7.3.3 (letting $\mathcal{E}' = \cdot$), and Theorem 6.3.14. $\square$

The analogue of Lemma 8.3.4 requires one additional restriction on the HTRS.

Definition 8.3.5. A well-typed object $\Gamma \vdash^{\prec}_{\Gamma,\mathcal{E}} M : A$ is said to be *$\mathcal{E}$-linear* if each variable $x \in \mathrm{dom}(\Gamma)$ such that $\mathcal{E} \preceq^{\mathrm{t}} \Gamma(x)$ appears free at most once in M.

A HTRS R is called *left $\mathcal{E}$-linear* if

$$(\Gamma \vdash^{\prec}_{\Gamma,\mathcal{E}} M : A) \in \mathcal{R} \text{ implies } l \ \mathcal{E}\text{-linear}$$

Lemma 8.3.6. *Let $\mathcal{R}$ be a left $\mathcal{E}$-linear HTRS, $\to_{\mathcal{R}_{\beta\eta}}$ and $\to_{\mathcal{E}_{\beta\eta}}$ satisfy condition (B) of Lemma 8.3.2 if and only if all the $(\mathcal{E}|\mathcal{R})$- and $(\mathcal{R}|\mathcal{E})$-critical pairs are trivial.*

Proof. Assume $\to_{\mathcal{R}_{\beta\eta}}$ and $\to_{\mathcal{E}_{\beta\eta}}$ satisfy condition (B) of Lemma 8.3.2, and let

$$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} <(\theta_1 E)_{\Downarrow}[\![\theta_2 r_2]\!], \theta_1 r_1>$$

be a $(\mathcal{E}|\mathcal{R})$-critical pair.

Define $M = (\theta_1 E)_\Downarrow[\![\theta_2 l_2]\!]$, we have, by the definitions and Corollary 8.2.10,

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to_{\mathcal{E}_{\beta\eta}} (\theta_1 E)_\Downarrow[\![\theta_2 r_2]\!] \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to_{\mathcal{R}_{\beta\eta}} \theta_1 r_1$$

By applying condition (B) of Lemma 8.3.2, we find objects P_i such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} (\theta_1 E)_\Downarrow[\![\theta_2 r_2]\!] \to^*_{\mathcal{R}_{\beta\eta}} U_1 \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} \theta_1 r_1 \to^*_{\mathcal{R}_{\beta\eta}} U_1 \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} U_1 \to^*_{\mathcal{E}_{\beta\eta}} U_2$$

Using Theorems 6.3.14 and 8.2.8, we see then

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} (\theta_1 E)_\Downarrow[\![\theta_2 r_2]\!] \to^*_{\mathcal{R}/\beta\eta} U_1 \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} \theta_1 r_1 \to^*_{\mathcal{R}/\beta\eta} U_1 \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} U_1 =^*_{\mathcal{E}\beta\eta} U_2$$

so the given critical pair is trivial. A similar proof shows that, under these same assumptions, all the $(\mathcal{R}|\mathcal{E})$-critical pairs are also trivial.

Conversely, assume that all the $(\mathcal{R}|\mathcal{E})$- and $(\mathcal{E}|\mathcal{R})$-critical pairs are trivial, and moreover

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to_{\mathcal{E}_{\beta\eta}} M_1 \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M \to_{\mathcal{R}_{\beta\eta}} M_2$$

for some objects M, M_1, and M_2.

By Corollary 8.2.10, there is a (not necessarily well-typed) environment $E^{(1)}$ such that

$$\begin{array}{ll}
\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_\Downarrow \approx E^{(1)}[\![M^{(1)}_\circ]\!] & \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} (M_1)_\Downarrow \approx E^{(1)}[\![N^{(1)}_\circ]\!] \\
\Gamma^{(1)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M^{(1)}_\circ \Downarrow \theta_1 C^{(1)}_\circ & \Gamma^{(1)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} N^{(1)}_\circ \Downarrow \theta_1 C^{(1)}_\circ \\
\Gamma^{(1)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M^{(1)}_\circ =_{\beta\eta} \theta_1 l_1 & \Gamma^{(1)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} N^{(1)}_\circ =_{\beta\eta} \theta_1 r_1 \\
\Gamma^{(1)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} \theta_1 : \Delta^{(1)}_\circ &
\end{array}$$

where either $(\Delta^{(1)}_\circ \vdash^{\preceq}_{\Sigma} l_1 = r_1 : C^{(1)}_\circ) \in \mathcal{E}$; or $(\Delta^{(1)}_\circ \vdash^{\preceq}_{\Sigma} r_1 = l_1 : C^{(1)}_\circ) \in \mathcal{E}$; we will assume the former.

Also, from Definition 6.3.1 we get

$$\begin{array}{ll}
\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} M_\Downarrow \approx E^{(2)}[\![M^{(2)}_\circ]\!] & \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} (M_2)_\Downarrow \approx E^{(2)}[\![N^{(2)}_\circ]\!] \\
\Gamma^{(2)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M^{(2)}_\circ \Downarrow \theta_2 C^{(2)}_\circ & \Gamma^{(2)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} N^{(2)}_\circ \Downarrow \theta_2 C^{(2)}_\circ \\
\Gamma^{(2)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M^{(2)}_\circ =_{\beta\eta} \theta_2 l_2 & \Gamma^{(2)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} N^{(2)}_\circ =_{\beta\eta} \theta_2 r_2 \\
\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E^{(2)}[\![\Gamma^{(2)}_\circ \vdash \circ : \theta_2 C^{(2)}_\circ]\!] \Downarrow A & \Gamma^{(2)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} \theta_2 : \Delta^{(2)}_\circ \\
\Delta^{(2)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} l_2 \to r_2 : C^{(2)}_\circ \in \mathcal{R} &
\end{array}$$

Applying Proposition 7.1.3 with $\mathcal{E}' = \cdot$, we obtain environments E^i and objects $M^i_\circ$ such that

$$\Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} E^{(i)} \approx E^i \qquad \Gamma^{(i)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M^{(i)}_\circ \approx_{\mathcal{E}'} M^i_\circ$$

and $M_\Downarrow = E^i[\![M^i_\circ]\!]$, for $i = 1, 2$.

From Propositions 4.3.2 and 6.3.16, it follows

$$\Gamma^{(i)}_\circ \vdash^{\preceq}_{\Sigma,\varepsilon} M^i_\circ =_{\mathcal{E}'\beta\eta} \theta_i l_i \qquad \Gamma \vdash^{\preceq}_{\Sigma,\varepsilon} (M_i)_\Downarrow \approx_{\mathcal{E}'} E^i[\![N^{(i)}_\circ]\!]$$

hence, without loss of generality, we can assume $E^{(i)} = E^i$ and $M^{(i)}_\circ = M^i_\circ$, and therefore

$$M_\Downarrow = E^{(i)}[\![M^{(i)}_\circ]\!]$$

We have three mutually disjoint cases to examine:

- Case $E^{(1)} \uparrow E^{(2)}$

  If $E^{(1)}$ and $E^{(2)}$ are disjoint environment, it is not difficult to see that they will have a common reduct:

  $$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 \rightarrow_{\mathcal{R}_{\beta\eta}} U \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_2 \rightarrow_{\mathcal{E}_{\beta\eta}} U$$

  hence, taking $N_1 = U$ and $N_2 = M_2$, we can show, using symmetry of $\rightarrow_{\mathcal{E}_{\beta\eta}}$,

  $$\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M_i \rightarrow^{*}_{\mathcal{R}_{\beta\eta}} N_i \ (i = 1, 2) \qquad \Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 \rightarrow^{*}_{\mathcal{E}_{\beta\eta}} N_2$$

  which concludes this case.

- Case $E^{(2)} = E^{(1)} \cdot E'$

  It will suffice to prove that condition (B) holds for $M_1 = N_\circ^{(1)}$ and $M_2 = E'[\![N_\circ^{(2)}]\!]$.

  By induction on the derivation of we easily show that E' is in this case well-formed. Since

  $$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1 l_1 =_{\beta\eta} E'[\![M_\circ^{(2)}]\!]$$

  there is an environment E satisfying the statement of Lemma 7.3.1 for $E' = \cdot$.

  If E satisfies case (1) of the statement, then

  $$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} E' \approx (\theta_1 E)_{\Downarrow}$$

  and therefore, through Propositions 6.1.4 and 6.3.16, we show

  $$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1 l_1 =_{\beta\eta} (\theta_1 E)_{\Downarrow}[\![\theta_2 l_2]\!]$$

  and $\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} < (\theta_1 E)_{\Downarrow}[\![\theta_2 l_2]\!], \theta_1 r_1 >$ is a $(\mathcal{R}|\mathcal{E})$-critical pair.

  Under the current working assumptions, it must be trivial, so there are N_1 and N_2 such that

  $$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} (\theta_1 E)_{\Downarrow}[\![\theta_2 l_2]\!] \rightarrow^{*}_{\mathcal{R}/\beta\eta} N_1 \qquad \Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1 r_1 \rightarrow^{*}_{\mathcal{R}/\beta\eta} N_2 \qquad \Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 =_{\mathcal{E}\beta\eta} N_2$$

  Using Theorems 6.3.14, 8.1.9 and 8.2.8, we show

  $$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} M_2 \rightarrow^{*}_{\mathcal{R}_{\beta\eta}} N_1 \qquad \Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} M_1 \rightarrow^{*}_{\mathcal{R}_{\beta\eta}} N_2 \qquad \Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} N_1 \rightarrow^{*}_{\mathcal{E}_{\beta\eta}} N_2$$

  If E satisfies part (2) of the statement of Lemma 7.3.1, then $l_1 = E[\![U_\circ]\!]$, where $U_\circ = x\ U_1\ U_2 \ldots U_k$ is a generalized variable, and

  $$\Gamma_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} E' \approx ((\theta_1 E)_{\Downarrow} \cdot E_\circ)$$
  $$\Gamma_\circ \vdash^{\prec}_{\Sigma,\mathcal{E}} \theta_1(x) =_{\beta\eta} \lambda y_1 : A_1 \ldots \lambda y_k : A_k.\ E_\circ[\![M_\circ^{(2)}]\!]$$

  for some (well-typed) environment $E_\circ$.

  Let $\Delta_\circ^{(1)}(x) = B$, applying Proposition 6.1.3 to the $E_\circ$, we get $C_\circ^{(2)} \preceq^{\mathrm{t}} B$. From this we immediately see $B \not\prec^{\tau} C_\circ^{(1)}$: otherwise, we would have

  $$C_\circ^{(2)} \preceq^{\mathrm{t}} B \prec^{\tau} C_\circ^{(1)}$$

  implying $C_\circ^{(2)} \prec^{\tau} C_\circ^{(1)}$, a contradiction to the assumption $\mathcal{R} \not\prec^{\tau} \mathcal{E}$.

  Let $l_1 = E_{U'}[\![U']\!]$ where $U' = x\ U_1'\ U_2' \ldots U_k'$ is any other occurrence of a generalized variable (with the same underlying variable x as U), we apply Proposition 6.1.6 to conclude

  $$\Delta_\circ^{(1)} \vdash^{\prec}_{\Sigma,\mathcal{E}} E_{U'}[\![\Delta_\circ^{(1)}(l_1, U') \vdash \circ : type(\Delta_\circ(l_1, U'))]\!] : C_\circ^{(1)}$$

Moreover, all these environments $E_{U_\circ}$ are disjoint, and $\Delta_\circ^{(1)}$-stable by Proposition 7.2.3.

By Corollary 7.2.5 and Proposition 6.1.6, we see that

$$\Delta_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} ((\theta_1 E_{U'})_{\Downarrow} \cdot E_\circ)[\![\Gamma_\circ^{(2)} \vdash \circ : \theta_2 C_\circ^{(2)}]\!] \Downarrow \theta_1 C_\circ^{(1)}$$

Using repeatedly Proposition 6.1.9, we show

$$\Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} E'[\![\theta_2 r_2]\!] \to^*_{\mathcal{R}/\beta\eta} \theta'_1 l_1$$

where $\Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta'_1 : \Delta_\circ^{(1)}$ is defined as:

$$\theta'_1(z) = \begin{cases} \theta_1(z) & z \neq x \\ \lambda y_1 : A_1 \ldots y_k : A_k.\ E_\circ[\![N_\circ^{(2)}]\!] & z = x \end{cases}$$

By applying a similar argument to r_1 we show

$$\Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_1 r \to^*_{\mathcal{R}/\beta\eta} \theta'_1 r$$

Hence

$$\Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_1 \to^*_{\mathcal{R}/\beta\eta} \theta'_1 l_1 \qquad \Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} M_2 \to^*_{\mathcal{R}/\beta\eta} \theta'_1 r_1 \qquad \Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta'_1 l_1 \to_{\mathcal{E}} \theta'_1 r_1$$

and we are done by Theorem 6.3.14.

- Case $E^{(1)} = E^{(2)} \cdot E'$

  The proof is similar to that of the previous case. Like before, we apply Lemma 7.3.1 to

$$\Gamma_\circ^{(1)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_2 l_2 =_{\beta\eta} E'[\![M_\circ^{(1)}]\!]$$

  obtaining E.

  If E satisfies case (1) of the statement of Lemma 7.3.1, the result is easily seen to follow from the fact that all the $(\mathcal{E}|\mathcal{R})$-critical pairs are trivial.

  Otherwise E satisfies case (2), and $l_2 = E[\![U_\circ]\!]$, where $U = x\ U_1\ U_2 \ldots U_k$ is a generalized variable; moreover

$$\Gamma_\circ^{(2)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} E' \approx_{\mathcal{E}} ((\theta_2 E)_{\Downarrow} \cdot E_\circ)$$
$$\Gamma_\circ \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_1(x) =_{\mathcal{E}\beta\eta} \lambda y_1 : A_1 \ldots \lambda y_k : A_k.\ E_\circ[\![M_\circ^{(1)}]\!]$$

  for some (well-typed) environment $E_\circ$.

  By linearity of l_2, U is the only sub-object where x appears free. Hence $(\theta_2 E)_{\Downarrow} = (\theta'_2 E)_{\Downarrow}$, where

$$\theta'_2(z) = \begin{cases} \theta_2(z) & z \neq x \\ \lambda y_2 : A_1 \ldots y_k : A_k.\ E_\circ[\![N_\circ^{(1)}]\!] & z = x \end{cases}$$

  and therefore $E'[\![N_\circ^{(1)}]\!] = (\theta'_2 l_1)_{\Downarrow}$.

  Finally, using Lemma 8.2.9, one shows

$$\Gamma_\circ^{(2)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} \theta_2 r_2 \to^*_{\mathcal{E}} \theta'_2 r_2$$

  so we can conclude

$$\Gamma_\circ^{(2)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} E'[\![N_\circ^{(1)}]\!] \to_{\mathcal{R}} \theta'_2 r_2 \qquad \Gamma_\circ^{(2)} \vdash^{\preceq}_{\Sigma,\mathcal{E}} N_\circ^{(2)} \to^*_{\mathcal{E}/\beta\eta} \theta'_2 r_2$$

  which is what we wanted to show.

$\square$

Theorem 8.3.7 (Critical Pair Criterion). *Let $\mathcal{R}$ be a left $\mathcal{E}$-linear HTRS such that $\to_{\mathcal{R}/\mathcal{E}\beta\eta}$ is strongly normalizing, then $\to_{\mathcal{R}/\beta\eta}$ is confluent modulo $\mathcal{E}$- and $\beta\eta$-conversion, i.e.*

$$M_1 =_{\mathcal{E}\beta\eta} M_2 \text{ and } M_i \to^*_{\mathcal{R}/\beta\eta} M_i' \ (i = 1, 2) \text{ implies } \exists N_1, N_2 \text{ s.t. } M_i \to^*_{\mathcal{R}/\beta\eta} N_i \ (i = 1, 2) \text{ and } N_1 =_{\mathcal{E}\beta\eta} N_2$$

for all objects M_i, and M_i', if and only if all the $(\mathcal{R}|\mathcal{R})$-, $(\mathcal{E}|\mathcal{R})$-, $(\mathcal{R}|\mathcal{E})$-critical pairs are trivial.

Proof. In light of Theorems 6.3.14, 8.1.9 and 8.2.8, confluence of $\to_{\mathcal{R}/\beta\eta}$ is confluent modulo $\mathcal{E}\beta\eta$-conversion is equivalent to confluence of $\to_{\mathcal{R}_{\beta\eta}}$ modulo $\to^*_{\mathcal{E}_{\beta\eta}}$. The result then follows from Lemmas 8.3.2, 8.3.4, and 8.3.6. $\square$

8.4 Summary

In this chapter we presented the general version of the Critical Pair Criterion. The key idea here was is to treat $\mathcal{E}$-conversion as a symmetric, rewrite-like relation. For this purpose we introduced the notions of one-step $\mathcal{E}$-conversion and canonical $\mathcal{E}$-conversion in Sections 8.1 and 8.2. We can see these as the analogues of the concepts defined in Sections 6.2 and 6.3.

In Section 8.3 we defined new critical pairs, coming from superposing rules to equations ($(\mathcal{E}|\mathcal{R})$-critical pairs) and equations to rules ($(\mathcal{R}|\mathcal{E})$-critical pairs). Finally, we proved that, as long as all these new critical pairs, as well as the old $(\mathcal{R}|\mathcal{R})-$ ones we defined in the last chapter, are shown to be trivial, a strongly normalizing HTRS $\mathcal{R}$ is confluent modulo $\mathcal{E}$.

Chapter 9

Directions of Future Research

The results presented can be considered just the first step in the study of rewriting in the context of rich type systems. There are numerous directions toward which the research started here can be expanded; we list a few of them.

9.1 Implementations

One of our intents in pursuing this research was to justify equational extensions of the LF Calculus and its implementations. Many interesting applications of Twelf would greatly benefit from having algebraic manipulation of expressions during type reconstruction. A version of Twelf implementing some of the ideas presented here has already been implemented and it is currently being tested. A rewriting module for Twelf, capable of normalizing terms with respect to a given HTRS, as well as checking the critical pairs of the HTRS itself, is also currently being designed.

9.2 Other Confluence Criteria

There are still several systems which fall outside the scope of the results presented in this dissertation, as they do not happen to satisfy neither of the conditions on which the results of Chapters 7 and 8 relied upon.

Systems not dealt with by our results include the frequently-occurring cases of associative, and associative-commutative operators. These have been thoroughly studied for the first-order case [58]. Although the theoretical machinery we developed during our study should greatly facilitate the task of lifting these results to our calculus, there might be some additional problems that need to be tackled. For example, first-order binary operators often translate in n-ary ones (where $n > 2$) in LF, where the extra arguments are needed to correctly specify the type of the object at hand. Developing results for these cases may require providing the means for detecting these extra arguments, and separate them from the main ones.

Another interesting class of systems are orthogonal HTRSs, studied by Nipkow for the simply-typed calculus [55]. Confluence for orthogonal systems can be established even in cases where strong normalization cannot be attained. This seems promising for dealing with HTRS which contain "catch all" rules like the one presented at the end of Section 1.2, which are by our definitions not normalizing (not even in a weak sense).

9.3 Narrowing

The implementation of the equational extension of Twelf [58], we mentioned in Section 9.1 uses a constraint-solving approach. We endow the type reconstruction engine with decisions procedures for subsets of some interesting domains (e.g. rational arithmetic). The equational problems that can be formulated in the language but cannot be solved by these procedures are delayed as constraints until more information is available. While our approach is certainly efficient, it is not very flexible, since it does not allow the user to define his/her own equational theories or modify the existing ones. Rewriting-based languages like Maude [10] offer this ability, under the (mild) assumption that the equational theories introduced can be oriented into confluent, strongly normalizing ones. We believe the way to offer similar capabilities in Twelf is through higher-order narrowing. Narrowing strategies for the simply-typed lambda calculus have been studied by Prehofer [65, 66]; his work should be a good starting point in pursuing this line of research.

9.4 Termination

One area which holds promise of benefiting from the use of dependence relations, and therefore deserves further investigation, is the study of termination of HTRSs. A conspicuous number of techniques have been proposed in these last few years, some based on syntactic orderings [2, 39, 33], similar to or extending the first-order Recursive Path Ordering (RPO) [14], some others based on interpretations over well-founded structures [34, 68].

Among the higher-order extensions of RPO, it is deserving of mention some recent work by Jouannaud and Rubio [33]. Their technique is the first, to our knowledge, to fully take advantage of information coming from the type structure, and this seems to help it achieve better results than the other proposed methods. Their definition of Higher-Order RPO requires, in addition to the standard well-founded quasi-order over the symbols of the signature, another quasi-order over types, defined as follows:

Definition 9.4.1. The *quasi-ordering on types* $\leq$ satisfies the following properties:

1. its strict part $<$ is well-founded
2. $\leq$ is *compatible with the term structure:*
   for all ground canonical terms $\Gamma \vdash^{\prec}_{\Sigma,\mathcal{E}} M : A$ if N is contained in M, $\Gamma(M, N) \vdash^{\prec}_{\Sigma,\mathcal{E}} N : type(\Gamma(M, N))$, we have $type(\Gamma(M, N)) \leq A$

We have the following:

Proposition 9.4.2. *Let $\prec = (\prec^{\tau}, \prec^{\mathrm{t}})$ be a dependence relation defined over a signature Σ, $\preceq^{\mathrm{t}}$ is a quasi-ordering on types, in the sense of Definition 9.4.1.*

Proof. First, observe $\preceq^{\mathrm{t}}$ is by definition reflexive and transitive, hence a quasi-order. Its strict part is well-founded, since defined on a finite set (the set of type family constants of Σ). Finally, its compatibility with the term structure has been shown in Lemma 5.2.1. ☐

We believe that orderings like this that take advantage of the type structure of terms have a clear advantage over purely syntactic methods which are defined over untyped lambda-terms. Their main drawback, that is the need for an additional well-founded quasi-order to be specified, is eliminated here since this can be automatically obtained by computing minimal dependence relations. It might be worthwile investigating if some of the other orderings mentioned earlier can be modified to include quasi-orderings on types, and, if so, to evaluate the advantage in doing so.

Part III

Applications

Chapter 10

A Change in Notation

In this third and last part of the dissertation we present some applications of the theory to various fields of logic and theoretical computer science. Before we proceed, however, we feel necessary to make substantial changes to the notation we use. In presenting these examples, we will adopt a syntax which is very close to the one used in Twelf. There are many reasons for this change; we list here a couple.

One motive to this change is to emphasize that the concepts presented so far lend themselves to be implemented on a computer. Not only implementation of our ideas is possible, but often turns out to be necessary, given the large number of critical pairs that need to be considered for non-trivial systems. We will borrow the syntax for rewriting rules from ElfRW [20], a rewriting module for Elf written by Pfenning and Gehrke in 1995. ElfRW was designed to perform rewriting and critical pair checking of rewriting modulo an empty equational theory, using the theoretical results presented by us in [71]. A new module for Twelf, the successor of Elf, capable of rewriting modulo some special equational theories, is currently being designed.

A second reason for this notational change is that it will allow us to take advantage of the many syntax-sugaring features that Twelf offers to make our presentation more comprehensible. One particularly annoying characteristic of LF is the proliferation of arguments of higher-order constants. The presence of a fair degree of redundancy in LF signatures was already pointed out in [44]. The Twelf interpreter alleviates this problem by allowing the user to keep some of the arguments of functional objects implicit. These implicit arguments are kept hidden from the user, and the type reconstruction algorithm automatically generates them when parsing a signature. Keeping redundant information implicit benefits us, since we are able to write complex signature concisely, and the reader, allowing him/her to concentrate on the crucial point of the definitions we give without being distracted by irrelevant items.

10.1 Twelf Syntax

Twelf signatures are usually stored as text-only files, without any additional formatting or font information. For this reason, the syntax in which they are written differ from the one we adopted so far. In Figure 10.1, we give a list of conversion rules between the two notations.

Two features that we will use intensively will be the ability to write operators in prefix, postfix, and infix

Twelf	LF
`[x:` $\underline{A}$ `]` $\underline{B}$	$\lambda x : A.\ M$
`c` $\underline{M_1} \ldots \underline{M_n}$	$c\ M_1\ \ldots\ M_n$
$\underline{A}$ `->` $\underline{B}$	$\Pi x : A.\ B \quad (x \notin \mathcal{FV}(B))$
`{x:` $\underline{A}$ `}` $\underline{B}$	$\Pi x : A.\ B \quad (x \in \mathcal{FV}(B))$
`a` $\underline{M_1} \ldots \underline{M_n}$	$a\ M_1\ \ldots\ M_n$

Figure 10.1: Twelf/LF Syntax Conversion

notation, and, as we mentioned earlier, the possibility of treating some arguments as implicit. Figure 10.2 show how both these features are used; the corresponding signature in our old notation can be seen in Figure 10.3.

As can also be noticed by looking at Figure 10.2, we will borrow from ElfRW the ability to tag rewrite rules with identifiers; as a matter of fact, we will also extend this notation to equations.

A description of the Twelf system beyond what presented so far is beyond the scope of this dissertation. For more information, we recommend the interested reader to consult [63].

```
obj   : type.
mor   : obj -> obj -> type.

id    : {O : obj} mor C O O.                     %prefix 10 id.
*     : mor O2 O3 -> mor O1 O2 -> mor O1 O3.    %infix right 5 *.

idl   : id O2 * F => F.
idr   : F * id O1 => F.
assoc : (F * G) * H => F * (G * H).
```

Figure 10.2: Signature and HTRS (New Notation)

$$\mathrm{dom}(\Sigma) = \{\mathbf{obj}, \mathbf{morph}, \mathbf{id}, \mathbf{o}\}$$

$$\begin{aligned}
\Sigma(\mathbf{obj}) &= \mathrm{type}\,. \\
\Sigma(\mathbf{morph}) &= \Pi x_1 : \mathbf{obj}.\ \Pi x_2 : \mathbf{obj}.\ \mathrm{type} \\
\Sigma(\mathbf{id}) &= \Pi x_1 : \mathbf{obj}.\ \mathbf{morph}\ x_1\ x_1 \\
\Sigma(\mathbf{o}) &= \Pi x_1 : \mathbf{obj}.\ \Pi x_2 : \mathbf{obj}.\ \Pi x_3 : \mathbf{obj}.\ \Pi y_1 : \mathbf{morph}\ x_2\ x_3.\ \Pi y_2 : \mathbf{morph}\ x_1\ x_2.\ \mathbf{morph}\ x_1\ x_3
\end{aligned}$$

Figure 10.3: Signature (Old Notation)

Chapter 11

Applications: Concurrent Processes

We show how the rewriting technique developed by us can be used to represent Milner's action structures and process calculus. This is not meant to be a tutorial on these concepts; we assume familiarity of the reader with [50], on which most of the material presented in this chapter is based.

Furthermore, this can be considered more a case study than a demonstrating example: we will present some negative results, i.e. cases where the tools we developed were not sufficient to tackle the problem at hand. We believe that presenting failures (as well as successes) of our theory is useful, as they stress out its current limits, and suggest directions of further research.

11.1 Action Structures

Definition 11.1.1. Syntactically, a static action structure is built from a set of *names* X and basic actions Π. We will use letters x, y, z, u, v, w to denote names, and π for basic actions. Actions are formed from these sets according to the following BNF:

$$\textit{Actions} \quad a \quad := \pi \mid \mathrm{id} \mid a \cdot b \mid a \otimes b \mid \mathrm{ab}_x\, a$$

In what follows, we use letters a, b, c, d to signify actions. Given an monoid of *arities* $(M, \star, \epsilon)$, freely generated by a set of *prime arities* Σ, and an assignment ρ of names to prime arities and of basic actions to ordered pairs of arities, such assignment is extended to generic actions by the following inference system:

$$\frac{\rho(\pi) = (m, n)}{\pi : m \to n} \qquad \frac{}{\mathrm{id}_m : m \to m}$$

$$\frac{a : k \to l \quad b : l \to m}{a \cdot b : k \to m} \qquad \frac{a : k \to m \quad b : l \to n}{a \otimes b : k \star l \to m \star n}$$

$$\frac{a : l \to m \quad \rho(x) = k}{\mathrm{ab}_x\, a : k \star l \to k \star m}$$

(above, we subscripted the identity action id by its arity; we will recur to this expedient often, to improve clarity). Finally, we require actions to satisfy the following congruences:

$$\begin{aligned}
& a \cdot \mathrm{id} = a = \mathrm{id} \cdot a && a \cdot (b \cdot c) = (a \cdot b) \cdot c \\
& a \otimes \mathrm{id}_\epsilon = a = \mathrm{id}_\epsilon \otimes a && a \otimes (b \otimes c) = (a \otimes b) \otimes c \\
& \mathrm{id}_m \otimes \mathrm{id}_n = \mathrm{id}_{m \star n} && (a \cdot b) \otimes (c \cdot d) = (a \otimes c) \cdot (b \otimes d) \\
& \mathrm{ab}_x\, \mathrm{id} = \mathrm{id} && \mathrm{ab}_x (a \cdot b) = (\mathrm{ab}_x\, a) \cdot (\mathrm{ab}_x\, b)
\end{aligned}$$

To formalize static action structures in LF, we start by giving a signature for the monoid of the arities:

```
arity : type.
prime : type.

0 : arity.
‘ : prime -> arity.              %prefix 15 ‘.
+ : arity -> arity -> arity.     %infix right 10.
```

We also introduce an equational theory $\mathcal{E}_M$ for arities, where $\mathcal{E}_M = \mathcal{E}(\mathcal{R}_M)$ is the equational theory generated by the strongly normalizing and confluent HTRS given by the following rules:

```
ar1 : 0 + M => M.
ar2 : M + 0 => M.
ar3 : (K + L) + M => K + (L + M).
```

The well-formed signature Σ_M and equational theory $\mathcal{E}_M$ are then extended to accommodate static action structures, by adding new constants

```
name   : prime -> type.
action : arity -> arity -> type

id : {M : arity} action M M.                                %prefix 15 id.
;  : action K L -> action L M -> action K M.                %infix right 10 ;.
*  : action K M -> action L N -> action (K + L) (M + N).    %infix right 10 *.
ab : (name W -> action L M) -> action (‘ W + L) (‘ W + M).  %prefix 15 ab.
```

and equations

```
c1  : A ; id M == A.
c2  : id M ; A == A.
c3  : (A ; B) ; C == A ; (B ; C).
p1  : A * id 0 == A.
p2  : id 0 * A == A.
p3  : (A * B) * C == A * (B * C).
pf1 : id M * id N == id (M + N).
pf2 : (A ; B) * (C ; D) == (A * C) ; (B * D).
af1 : ab ([x:name W] id M) == id (‘ W + M).
af2 : ab ([x:name W] (A x) ; (B x)) == ab ([x:name W] A x) ; ab ([x:name W] B x).
```

We will call these new signature and equational theory Σ_A and $\mathcal{E}_A$, respectively.

We would be tempted to try to show, in analogy to what we did for $\mathcal{E}_M$, that $\mathcal{E}_A = \mathcal{E}(\mathcal{R}_A)$, by carefully choosing an orientation for the equations that make them into a confluent (modulo $\mathcal{E}_M$) and terminating HTRS. Unfortunately, this turns out not to be the case. Consider the subset of zero-ary actions:

$$A_{\epsilon,\epsilon} = \{a : m \to n \mid m = n = \epsilon\}$$

It is easily shown that $A_{\epsilon,\epsilon}$ is closed under $\cdot$ and $\otimes$. Hence, if $X = \emptyset$, this forms a (sub-) action structure. Over this subset, the operators $\cdot$ and $\otimes$ are easily seen to collapse into a single, associative-commutative one:

$$a \cdot b = (\mathrm{id} \otimes a) \cdot (b \otimes \mathrm{id}) = (\mathrm{id} \cdot b) \otimes (a \cdot \mathrm{id}) = b \otimes a = (b \cdot \mathrm{id}) \otimes (\mathrm{id} \cdot a) = (b \otimes \mathrm{id}) \cdot (\mathrm{id} \otimes a) = b \cdot a$$

Although this may clash with our intuitive notion of $\cdot$ representing sequentiality, the above fact will be crucial, as we will see, in representing CCSs as action structures.

Definition 11.1.2. A (dynamic) action structure consists of an action structure $A(\Pi, X, M, \rho)$ together with a action preorder $\searrow$ called *reaction* such that

- each operator preserves reaction, e.g. $a \searrow a'$ implies $a \cdot b \searrow a' \cdot b$ and $b \cdot a \searrow b \cdot a'$;

- id is inactive, i.e. id $\searrow a$ implies $a = \mathrm{id}$;
- reaction preserves arities, i.e. $a \searrow a'$ and $a : m \to n$ implies $a' : m \to n$.

Given their reductional nature, reactions will be modeled in LF as a HTRS $\mathcal{R}_R$, whose confluence (modulo $\mathcal{E}_A$) will have to be checked using the theory developed in Chapter 8.

11.2 Action structures for CCS

In Synchronous CCS [48], basic actions are formed taking positive and negative instances from a set of *atomic particles* Π_0. More formally, we have the following:

$$
\begin{aligned}
&X = \emptyset \\
&\Pi = \{\pi \mid \pi \in \Pi_0\} \cup \{\overline{\pi} \mid \pi \in \Pi_0\} \\
\\
&M = \{\epsilon\} \\
&\rho(\pi) = \rho(\overline{\pi}) = (\epsilon, \epsilon) \text{ for all } \pi \in \Pi_0
\end{aligned}
$$

For every $\pi \in \Pi_0$, instances of π of opposite sign react canceling each other:

$$\pi \otimes \overline{\pi} \searrow \mathrm{id}$$

A LF signature for this action structure is given by defining a new type for the particles in Π, and two "signing" operators:

```
particle : type.

pos : particle -> action 0 0.    %prefix 15 pos.
neg : particle -> action 0 0.    %prefix 15 neg.
```

Additionally, each $\pi \in \Pi_0$ will be represented by an object constant of type `particle`.

Since all the actions lie inside $A_{\epsilon,\epsilon}$, as we showed before, it can be proved that $\otimes$ and $\cdot$ coincide for this action structure, and are both commutative. Therefore the equational theory $\mathcal{E}_A$ in this case simplifies to the one of a commutative monoid:

```
ccs1 : A * B == B * A : action 0 0.
ccs2 : (A * B) * C == A * (B * C).
ccs3 : (id 0) * A == A.
```

Reactions are formalized by the single-rule HTRS $\mathcal{R}_R$

```
ccsr : pos M * neg M => id 0.
```

Unfortunately, the theoretical results proven in Chapter 8 are not strong enough to tackle theories with associative-commutative operators. However, in this specific instance we observe that, since $X = \emptyset$ and $M = \{\epsilon\}$, the action structure for CCS can be described adequately by a first order theory. Hence we can apply the Peterson-Stickel method [58] to show that $\mathcal{R}_R$ is indeed confluent modulo $\mathcal{E}_A$.

Action structures for Pure CCS [49] are defined similarly, but require the addition of a special particle τ. In the Pure CCS case, particles of opposite sign react producing τ, rather than the identity:

$$\pi \otimes \overline{\pi} \searrow \tau$$

The encoding into LF, and proof of confluence of the resulting system, are handled like before.

11.3 Process Calculus

Action structures can be used in more than one way to model concurrent processes. The first way is to use them to model each single transition of a process. A second approach is to model the whole process, as well as its individual transitions, as action structures. We take the first of these two approaches here, following [50].

Definition 11.3.1. Processes are defined over an action structure A, and formed according to the following syntax:

$$\begin{array}{lll} \textit{Processes} & P := & a \cdot P \mid P \otimes P \mid (x)P \\ & & {}_{\bullet}P \mid \mathbf{0} \mid P + P \mid \partial P \mid !P \end{array}$$

We will use letters P, Q, R to denote processes. Each process is assigned an arity, according to the following rules:

$$\frac{a : m \to n \quad P : n}{a \cdot P : m} \qquad \frac{P : m \quad Q : n}{P \otimes Q : m \star n} \qquad \frac{\rho(x) = k \quad P : m}{(x)P : k \star m}$$

$$\frac{P : m}{{}_{\bullet}P : m} \qquad \frac{}{\mathbf{0} : m} \qquad \frac{P : m \quad Q : m}{P + Q : m} \qquad \frac{P : m}{\partial P : m} \qquad \frac{P : \epsilon}{!P : \epsilon}$$

Finally, processes must satisfy the following rules of structural congruence:

$$\begin{array}{cc} id \cdot P = P & a \cdot (b \cdot P) = (a \cdot b) \cdot P \\ (x)(a \cdot P) = (\mathrm{ab}_x\, a) \cdot (x)P & (a \cdot P) \otimes (b \cdot Q) = (a \otimes b) \cdot (P \otimes Q) \end{array}$$

In formalizing the process calculus in LF, we choose, rather than overloading some of the already defined symbols, to prepend "#" to all the newly introduced constants:

```
proc : arity -> type.

#;  : action M N -> proc N -> proc M.
#*  : proc M -> proc N -> proc (M + N).
#ab : (name W -> proc M) -> proc (‘ W + M).
##  : proc M -> proc M.
#0  : proc M.
#+  : proc M -> proc M -> proc M.
#d  : proc M -> proc M.
#!  : proc 0 -> proc 0.
```

Structural congruence can be expressed by the following HTRS $\mathcal{R}_p$:

```
pc1 : (id M) #; P => P.
pc2 : A #; (B #; P) => (A ; B) #; P.
pc3 : (A #; P) #* (B #; Q) => (A * B) #; (P #* Q).
pc4 : #ab ([x:name W] (A x) #; (P x)) => ab ([x:name W] A x) #; #ab ([x:name W] P x).
```

The proof of confluence of $\mathcal{R}_P$ is a nice application of the theory developed in this thesis, and therefore it is worth presenting it in some detail.

As we mentioned in Chapter 5, when typing a signature Σ, it is important to also determine what is the minimal dependence relation under which this is possible. In this case, it turns out to be the following:

$$\begin{array}{l} \texttt{prime}, \texttt{arity} \prec^{\tau} \texttt{action}, \texttt{proc} \\ \texttt{prime} \prec^{\tau} \texttt{name} \\ \texttt{prime} \prec^{\mathrm{t}} \texttt{arity}, \texttt{name} \prec^{\mathrm{t}} \texttt{action} \prec^{\mathrm{t}} \texttt{proc} \end{array}$$

By the kinds assigned to action and proc, it must be $\texttt{arity} \prec^{\tau} \texttt{action}, \texttt{proc}$. For similar reasons, we also have $\texttt{prime} \prec^{\tau} \texttt{name}$. Typing the embedding $'$ of primes into arities introduces the pair $\texttt{prime} \prec^{\mathrm{t}} \texttt{arity}$. Hence

$$\texttt{prime} \prec^{\mathrm{t}} \texttt{arity} \prec^{\tau} \texttt{action}, \texttt{proc}$$

and therefore $\texttt{prime} \prec^{\tau} \texttt{action}, \texttt{proc}$.

Finally, the sequential composition #; of an action and a process forces $\texttt{action} \prec^{\mathrm{t}} \texttt{proc}$, and the two abstractions ab and #ab also require $\texttt{name} \prec^{\mathrm{t}} \texttt{action}, \texttt{proc}$. Since $\prec^{\tau} \subseteq \prec^{\mathrm{t}}$, we conclude

$$\texttt{prime} \prec^{\mathrm{t}} \texttt{arity}, \texttt{name} \prec^{\mathrm{t}} \texttt{action} \prec^{\mathrm{t}} \texttt{proc}$$

In Section 9.4 we mentioned how dependence relations can be used in the Recursive Path Ordering described in [33] to establish strong normalization. In this case, it can be verified that using the quasi-ordering on types $\prec^{\mathrm{t}}$ above, together with the symbol ordering

$$\texttt{\#;} < \texttt{\#*} < \texttt{\#ab}$$

is sufficient to prove strong normalization of $\mathcal{R}_P$.

Once $\mathcal{R}_P$ has been shown terminating, we turn once again to the dependence relation $\prec$, to observe that $\texttt{proc} \not\preceq^{\mathrm{t}} \texttt{action}$. Hence $\mathcal{R}_P \not\preceq^{\mathrm{t}} \mathcal{E}_A$, and we can use Theorem 7.3.4 to establish confluence by examining all the $(\mathcal{R}_P|\mathcal{E}_A|\mathcal{R}_P)$-critical pairs. There are fifteen of them, and they all turn out to be trivial by applying congruences in $\mathcal{E}_A$.

The proof of termination and confluence sketched above represent an elegant, independent proof to Propositions 5.7 and 5.8 in [50].

Definition 11.3.2 (Dynamic Process Calculus). Computation in the Process Calculus is performed by a *reduction preorder* $\searrow$ defined as the union of an *action preorder* $\overset{a}{\searrow}$ and a control preorder $\overset{c}{\searrow}$.

The action preorder is responsible for computation at the atomic level, i.e. inside actions:

$$a \cdot P \overset{a}{\searrow} a' \cdot P \text{ iff } a \searrow a'$$

The control preorder regulates computation of processes:

$$\begin{array}{cccc} {\bullet}P \otimes {\bullet}Q \overset{c}{\searrow} {\bullet}(P \otimes Q) & P+Q \overset{c}{\searrow} P & \partial P \overset{c}{\searrow} P & !P \overset{c}{\searrow} P \otimes !P \\ (x){\bullet}P \overset{c}{\searrow} {\bullet}(x)P & P+Q \overset{c}{\searrow} Q & \partial P \overset{c}{\searrow} {\bullet}\partial P & !P \overset{c}{\searrow} {\bullet}!P \end{array}$$

In LF, the action preorder can be adequately represented by the HTRS $\mathcal{R}_R$ defined before. Let $\mathcal{E}_P = \mathcal{E}_A \cup \mathcal{E}(\mathcal{R}_P)$, we introduce the following HTRS $\mathcal{R}_C$ for the control preorder:

```
cp1 : (## P) #* (## Q) => ## (P #* Q).
cp2 : #ab ([x:name W] ## (P x)) => ## (#ab [x:name W] P x).
cp3 : P #+ Q => P.
cp4 : P #+ Q => Q.
cp5 : #d P => P.
cp6 : #d P => ## (#d P).
cp7 : #! P => P #* (#! P).
cp8 : #! P => ## (#! P).
```

In general, $\mathcal{R}_C$ is neither normalizing (not even in a weak sense), nor confluent; this is as expected, since it is designed to model the behavior of non-deterministic, and possibly non-terminating, processes.

Chapter 12

Applications: Categories

In this chapter we will show how categorical notions can be expressed by HTRSs. In doing so, the work done by W. Gehrke in [19] serves as our inspiration. However, the encoding he uses turns out not to be adequate in some cases: the HTRSs he obtains, albeit confluent and strong-normalizing, in some instances do not represent the categorical structures they intend to.

The encoding we will present here is adequate. However, the problem of orienting higher-order equations into rewrite rules becomes here more delicate: fewer, and different choices than the ones presented in [19] are possible, if confluence and termination are to be attained.

12.1 Categories

Definition 12.1.1. A category is specified by the following data:

- A collection of entities called *objects*.
- A collection of entities called *morphisms*. To each morphism we associate two objects: a *source* and a *target*. We will write $f : a \to b$ to indicate that the morphism f has source a and target b.
- A binary composition operator, that, given morphisms $f : a \to b$ and $g : b \to c$, give raise to the morphism $(g \cdot f) : a \to c$. Composition is associative, i.e. for all morphisms f, g, h

  $$(h \cdot g) \cdot f = h \cdot (g \cdot f)$$

  whenever the composition above is defined.
- For each object a, an *identity* morphisms $\mathrm{id}_a : a \to a$. Identity morphisms act as unit element of composition, i.e. given any morphism $f : a \to b$, the following holds:

  $$\mathrm{id}_b \cdot f = f = f \cdot \mathrm{id}_a$$

In what follows, we will have to deal with definitions that involve more than a single category. Hence we choose a LF representation that introduces a type for categories, allowing objects and morphisms to be indexed by the categories they belong to:

```
cat   : type.

obj   : cat -> type.
mor   : {C : cat}  obj C -> obj C -> type.

id    : {O : obj C} mor C O O.                     %prefix 10 id.
*     : mor C O2 O3 -> mor C O1 O2 -> mor C O1 O3.  %infix right 5 *.
```

The associativity and unitary congruences can be expressed by the following HTRS:

```
idl   : id O2 * F => F.
idr   : F * id O1 => F.
assoc : (F * G) * H => F * (G * H).
```

We will use the symbols Σ_{cat} and $\mathcal{R}_{\mathrm{cat}}$ to denote the signature and HTRS just introduced.

Two important remarks about these signature and equational theory, which will hold true for all the other ones we will present in this chapter, are the following:

1. The rules are essentially first-order, with dependent-types annotations that restrict their applicability.
2. Term constants require a large number of auxiliary arguments; these are needed to specify the type of the main ones. For example, composition of morphism is represented by a LF functional object requiring six arguments; this does not agree with our intuitive notion of composition, which requires just two.

We use these observations to outline a different, and more effective, strategy to prove strong normalization. This goes through the definition of a map $|\cdot|$ from canonical LF terms to first-order untyped ones, such that

$$\Gamma \vdash^{\prec}_{\Sigma} M \to_{\mathcal{R}/\beta\eta} N : A \text{ implies } |M| \to_{|\mathcal{R}|} |N|$$

where $|\mathcal{R}|$ is the TRS obtained by applying the transformation $|\cdot|$ to the HTRS $\mathcal{R}$. Using such a map, the problem of establishing strong normalization for $\mathcal{R}$ can be reduced to proving it for $|\mathcal{R}|$, which can be done using a wider array of (first-order) techniques.

We show that one map satisfying the above requirements is given the following:

$$\begin{aligned} |\lambda x : A.\ M| &= \lambda(|M|[\star/x]) \\ |c\ M_1\ M_2\ \ldots\ M_n| &= c(|M_1|, |M_2|, \ldots, |M_n|) \\ |x\ M_1\ M_2\ \ldots\ M_n| &= @_{n+1}(x, |M_1|, |M_2|, |M_n|) \end{aligned}$$

where $\lambda, \star, @_i (i \geq 0)$ are new function symbols required by the translation, and disjoint from those in $\mathrm{dom}(\Sigma)$.

By Remark 1, all the free variables of rules in $\mathcal{R}$ are of base type, and hence rewriting does not create new β-redexes. This being the case, it suffices to show than the map we defined satisfies

$$\Gamma \vdash^{\prec}_{\Sigma} M \to_{\mathcal{R}} N : A \text{ implies } |M| \to_{|\mathcal{R}|} |N|$$

which is easily proved by induction on the environment E used in the rewriting step.

Although this map is as required, it is not optimal: as we noted in Remark 2, some of the arguments required by the LF constants serve no other purpose than to carry type information, and are therefore unneeded in the translation, where types have been stripped off. How to sort out which arguments should be carried in the translation? Dependence relations come to the rescue, once again: from Proposition 6.1.3 it follows that we can safely discard arguments of type A whenever $\mathtt{morph} \not\preceq^{\mathrm{t}} A$.

Hence, in an optimized version of the translation we have

$$\begin{aligned} |c\ M_1\ M_2\ \ldots\ M_n| &= c(|M_{k_1}|, |M_{k_2}|, \ldots, |M_{k_m}|) && 1 \leq k_1 < k_2 < \cdots < k_m \leq n \\ |x\ M_1\ M_2\ \ldots\ M_n| &= @_{n+1}(x, |M_{k_1}|, |M_{k_2}|, \ldots, |M_{k_m}|) && 1 \leq k_1 < k_2 < \cdots < k_m \leq n \end{aligned}$$

where the k_i are the indices of the parameters that are "needed", in the sense of the criterion described above.

In the specific case of Σ_{cat} and $\mathcal{R}_{\mathrm{cat}}$, a minimal dependence relation $\prec$ is given by

$$\mathtt{cat} \prec^{\tau} \mathtt{obj} \prec^{\tau} \mathtt{morph} \qquad\qquad \mathtt{cat} \prec^{\mathrm{t}} \mathtt{obj} \prec^{\mathrm{t}} \mathtt{morph}$$

This determines the following first-order translation:

$$\begin{aligned} |\mathtt{id}\ \mathcal{C}\ a| &= \mathtt{id} \\ |*\ \mathcal{C}\ a\ b\ c\ f\ g| &= f * g \end{aligned}$$

for which the rules $|R_{\mathrm{cat}}|$ take the familiar shape of those of a free monoid $(M, *, \mathtt{id})$.

12.2 Functors

Definition 12.2.1. Let $\mathcal{C}$ and $\mathcal{D}$ two categories, a functor F from $\mathcal{C}$ to $\mathcal{D}$ (in symbols: $F : \mathcal{C} \to \mathcal{D}$) is specified by:

- An operation taking objects a in $\mathcal{C}$ to objects $F\ a$ in $\mathcal{D}$.
- An operation taking morphism $f : a \to b$ in $\mathcal{C}$ to morphisms $F\ f : F\ a \to F\ b$ in $\mathcal{D}$

such that

$$F\ \mathrm{id}_a = \mathrm{id}_{F\ a} \qquad\qquad F\ (g \cdot f) = (F\ g) \cdot (F\ f)$$

We represent functors as a new type **fun**, dependent on two categories. The operations mentioned in the definition are represented by two object constants **funObj** and **funMorph**, parametric on a functor. We define the signature Σ_{fun}, obtained extending Σ_{fun} with constants:

```
fun    : cat -> cat -> type.

funObj : fun C D -> obj C -> obj D.
funMor : {FF : fun C D} mor C O1 O2 -> mor D (funObj FF O1) (funObj FF O2).
```

The properties that **funMorph** is required to satisfy are expressed by the HTRS $\mathcal{R}_{\text{fun}}$ given by the union of $\mathcal{R}_{\text{cat}}$ and the following set of rules:

```
fid : funMor FF (id O) |> id (funObj FF O).
f*1 : (funMor FF F) * (funMor FF G) |> funMor FF (F * G).
f*2 : (funMor FF F) * (funMor FF G) * H |> funMor FF (F * G) * H.
```

This HTRS can be proved to be terminating using the technique illustrated before. In particular, $|\mathcal{R}_{\text{fun}}|$, is easily verified to be as follows:

$$\begin{aligned}
x * \mathtt{id} &\to x \\
\mathtt{id} * x &\to x \\
(x * y) * z &\to x * (y * z) \\
\mathtt{funMorph}(\mathtt{id}) &\to \mathtt{id} \\
\mathtt{funMorph}(x) * \mathtt{funMorph}(y) &\to \mathtt{funMorph}(x * y) \\
\mathtt{funMorph}(x) * (\mathtt{funMorph}(y) * z) &\to \mathtt{funMorph}(x * y) * z
\end{aligned}$$

This first order TRS can be shown terminating by using, for example, the polynomial interpretation used by Gehrke in [19]:

$$\begin{aligned}
\rho(\mathtt{id}) &= 1 \\
\rho(x * y) &= \rho(x)\rho(y) + \rho(x) \\
\rho(\mathtt{funMorph}(x)) &= \rho(x) + 1
\end{aligned}$$

Since $|\mathcal{R}_{\text{fun}}|$ is strongly normalizing, so is $\mathcal{R}_{\text{fun}}$, and therefore we can apply either one of the Critical Pair Criteria we presented in Chapters 7 and 8 ($\mathcal{E} = \emptyset$ in this case, and $\mathcal{R}_{\text{fun}}$ is left-linear). All critical pairs are trivial, from which confluence of the system follows.

12.3 Natural Transformations

Definition 12.3.1. Let $\mathcal{C}, \mathcal{D}$ be categories, and $F, G : \mathcal{C} \to \mathcal{D}$ two functors. A natural transformation η from F to G, written $\eta : F \to G$, is specified by an operation which assigns to each object a in $\mathcal{C}$ a morphism $\eta_a : F\ a \to G\ a$ in D such that, for any morphism $f : a \to b$ in $\mathcal{C}$ we have

$$(G\ f) \cdot \eta_a = \eta_b \cdot (F\ f)$$

i.e. the following diagram commutes:

$$\begin{tikzcd}
F\,a \arrow[r, "\eta_a"] \arrow[d, "Ff"'] & G\,a \arrow[d, "Gf"] \\
F\,b \arrow[r, "\eta_b"'] & G\,b
\end{tikzcd}$$

In representing natural transformations, we follow the same approach used for functors: we introduce a new type **nat**, dependent on two functors (as well as the two categories these functors are defined upon), and we express the operation $a \mapsto \eta_a$ as with a constant **natObj**:

```
nat    : fun C D -> fun C D -> type.

natObj : nat FF GF -> {O : obj C} mor D (funObj FF O) (funObj GF O).
```

We call Σ_{nat} this extension of the signature Σ_{fun} used for functors. Two new rules are added to $\mathcal{R}_{\text{fun}}$, forming a new HTRS R_{nat}:

```
n1    : (funMor GF (F : fun O1 O2)) * (natObj FN O1) =>
            (natObj FN O2) * (funMor FF F).
n2    : (funMor GF (F : fun O1 O2)) * (natObj FN O1) * H =>
            (natObj FN O2) * (funMor FF F) * H.
```

This is shown terminating and confluent by the same techniques used before.

12.4 Adjunctions

Definition 12.4.1. Let $L : \mathcal{C} \to \mathcal{D}$ and $L : \mathcal{D} \to \mathcal{C}$ two functors, we say that L is *left adjoint* to R or that R is *right adjoint* to L if

- There is a bijection ζ between morphisms $L\,a \to b$ in $\mathcal{D}$ and morphisms $a \to R\,b$ in $\mathcal{C}$ (for all objects a in $\mathcal{C}$ and b in $\mathcal{D}$).
- ζ is *natural in a and b*, i.e given $f : a' \to a$ in $\mathcal{C}$, and $g : L\,a \to b$ and $h : b \to b'$ in $\mathcal{D}$, we have

$$\zeta(h \cdot g \cdot (L\,f)) = (R\,h) \cdot \zeta(g) \cdot f \qquad (*)$$

Trying to translate this definition into a signature Σ_{adj} and a HTRS $\mathcal{R}_{\text{adj}}$, it is useful to clarify what symbols and equations need to be used to describe an adjunction. First, observe that the (*) can be equivalently expressed by the two equations

$$\zeta(h \cdot g) = (R\,h) \cdot \zeta(g) \qquad\qquad \zeta(g \cdot (L\,f)) = \zeta(g) \cdot f$$

Also, ζ is a bijection; hence, this implicitly assumes the existence of a map ζ^{-1}, and the equations

$$\zeta^{-1}(\zeta(f)) = f \qquad\qquad \zeta(\zeta^{-1}(g)) = g$$

to hold for all $f : L\,a \to b$ and $g : a \to R\,b$ in $\mathcal{D}$ (with a in $\mathcal{C}$ and b in $\mathcal{D}$).

Not all the equations and symbols above, however, are strictly needed, as shown by the following result, due to Curien [13]:

Lemma 12.4.2. *An adjunction can be completely characterized by ϵ, L, and ζ, where*

$$\epsilon_a = \zeta^{-1}(\mathrm{id}_{R\,a})$$

The missing components can be expressed as follows:

$$R\,f = \zeta(f \cdot \epsilon_a)$$
$$\zeta^{-1}(g) = \epsilon_b \cdot L\,g$$

Using this result, we will give a representation that

- will not use ζ^{-1}, and therefore will not represent this map in the signature;
- will not contain references to the functor R other than when it is used to act on objects in $\mathcal{D}$.

The signature Σ_{adj} will extend Σ_{nat} with the new constants

```
adj : fun C D -> fun D C -> type.

eps  : adj (L : fun C D) R -> {Y : obj D} morph D (funObj L (funObj R Y)) Y.
zeta : adj (L : fun C D) R -> morph D (funObj L X) Y -> morph C X (funObj R Y).
```

As usual, we will treat some of the arguments of the operators as implicit. The HTRS $\mathcal{R}_{\mathrm{adj}}$ similarly extends $\mathcal{R}_{\mathrm{nat}}$, adding new rules

```
zeta1 : zeta (A : adj L R) ((eps A O2) * funMorph L F) => F.
zeta2 : zeta (A : adj L R) (eps A O) => id (funObj R O).
zeta3 : zeta (A : adj L R) F * G => zeta A (F * funMorph L G).

eps1 : (eps (A : adj L R) O2) * (funMorph L (zeta A F)) => F.
eps2 : (eps (A : adj L R) O2) * (funMorph L (zeta A F)) * H => F * H.
```

A simple (but long and tedious) check reveals $\mathcal{R}_{\mathrm{adj}}$ to be stongly normalizing and confluent.

Chapter 13

Intuitionistic Natural Deduction and Sequent Calculus

In this chapter we will present a process for converting between Natural Deduction and Sequent Calculus. The two directions of this conversion will both be implemented by confluent, strongly-normalizing HTRSs. One obstacle we will find in doing this is that the translation into sequent derivations requires the use of the cut rule. For this reason, we will also present a third system, which will transform any sequent derivation into a cut-free one; this is based on the work of F. Pfenning, who in [61] describes a representation of the cut-elimination theorem in the LF logical framework.

In pictures, the content of this chapter can be summarized as follows:

$$\begin{tikzcd}
 & \boxed{\text{Natural Deduction}} \arrow[dr, bend left, "\mathcal{R}_{SQ}"] & \\
\boxed{\text{Sequent Calculus}} \arrow[ur, bend left, "\mathcal{R}_{ND}"] & & \boxed{\text{Sequent Calculus + Cut}} \arrow[ll, "\mathcal{R}_{cut}"]
\end{tikzcd}$$

13.1 Natural Deduction

Our first task is to fix a representation for logical formulas. We will restrict ourselves to the propositional calculus: while Twelf can easily accommodate for first-order logic, the meta-representation of inference rules involving quantifiers force us to consider some non-pattern expressions, which lie outside the (current) range of applicability of our tools. However, there is very little loss of generality in doing so: most of the proof-theoretical issues and concepts in the transformations we describe are well-represented by the the propositional fragment.

We will consider the full set of logical connectives and constants:

$$\textit{Formulas} \quad A \quad := \quad P \mid A_1 \wedge A_2 \mid A_1 \supset A_2 \mid A_1 \vee A_2 \mid \neg A \mid \bot \mid \top$$

A Twelf signature for these, shown in Figure 13.1, is composed by a single type `o` for formulas, as well as object constants for each connective.

Natural Deduction [64] tries to reproduce in a formal context the process of logical inference used in most mathematical proofs. Judgments derived using this calculus are of the form

$$\vdash A$$

stating that the formula A is provable; its proof may rely on many outstanding assumptions, that have not been "discharged" during its development.

```
o : type.

true  : o.
false : o.

not : o -> o.             %prefix 10 not
and : o -> o -> o.        %infix right 4 and
or  : o -> o -> o.        %infix right 4 or
imp : o -> o -> o.        %infix right 4 imp
```

Figure 13.1: A Signature for Propositional Formulas

The rules of inference of the for intuitionistic Natural Deduction can be neatly organized in two groups, *introduction* and *elimination* rules. Each connective and logical constant admits at most one rule of each group; moreover, introduction and elimination rules for any given connective are complementary, in the sense that any immediately consecutive application of the two is redundant, and can be safely removed from the proof. The complete list of rules in the calculus is given in Figure 13.2.

In Twelf, we introduce a new type `nd`, dependent on objects of type `o`, to represent the judgment $\vdash A$. Rules of the calculus are implemented as functional objects, accepting (representations of) the antecedents as arguments, producing (a representation of) the consequent. For example, the rule

$$\frac{\vdash A \wedge B}{\vdash B}{\scriptstyle \wedge E_2}$$

translates into the object

```
andE2   : nd (A and B) -> nd B.
```

The complete signature is given in Figure 13.3. A proof of the adequacy of this representation can be found, for example, in [60].

13.2 Sequent Calculus

Gentzen's Sequent Calculus [21] has been an important tool for proof theoretical investigations and applications of logic in computer science. Although several formulations have been given over time, sequents are commonly found expressed as judgments of the form

$$\Gamma \longrightarrow \Delta$$

signifying that the formulas in Γ entail those in Δ. The intuitionistic fragment is elegantly isolated by requiring Δ to be a singleton formula C.

Rules of the calculus can be grouped into five classes. A *initial rule* states that any formula immediately entail itself:

$$\frac{}{A \longrightarrow A}{\scriptstyle I}$$

We have also *left* and *right rules* for each logical symbol, to perform steps of logical inference on the formulas on the left (i.e. Γ) or on the right (Δ), respectively, of the sequent.

Another class is formed by the so-called *structural rules*, which allow us to rearrange the order of Γ and Δ, as well as regulate the addition and removal of formulas from them. For intuitionistic logic, a common choice of structural rules are *exchange*, *weakening*, and *contraction*:

$$\frac{\Gamma_1, B, A, \Gamma_2 \longrightarrow C}{\Gamma_1, A, B, \Gamma_2 \longrightarrow C}{\scriptstyle ex} \qquad \frac{\Gamma_1 \longrightarrow C}{\Gamma_1, \Gamma_2 \longrightarrow C}{\scriptstyle wk} \qquad \frac{\Gamma, A, A \longrightarrow C}{\Gamma, A \longrightarrow C}{\scriptstyle ct}$$

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\wedge I \qquad\qquad \frac{\vdash A \wedge B}{\vdash A}\wedge E_1 \quad \frac{\vdash A \wedge B}{\vdash B}\wedge E_2$$

$$\frac{\begin{array}{c}\overline{\vdash A}^{\,u}\\ \vdots \\ \vdash B\end{array}}{\vdash A \supset B}\supset I^u \qquad\qquad \frac{\vdash A \qquad \vdash A \supset B}{\vdash B}\supset E$$

$$\frac{\vdash A}{\vdash A \vee B}\vee I_1 \quad \frac{\vdash B}{\vdash A \vee B}\vee I_2 \qquad \frac{\vdash A \vee B \qquad \begin{array}{c}\overline{\vdash A}^{\,u}\\ \vdots \\ \vdash C\end{array} \qquad \begin{array}{c}\overline{\vdash B}^{\,v}\\ \vdots \\ \vdash C\end{array}}{\vdash C}\vee E^{u,v}$$

$$\frac{\begin{array}{c}\overline{\vdash A}^{\,u}\\ \vdots \\ \vdash p\end{array}}{\vdash \neg A}\neg I^p \qquad\qquad \frac{\vdash \neg A \qquad \vdash A}{\vdash C}\neg E$$

$$\frac{}{\vdash \top}\top I$$

$$\frac{\vdash \bot}{\vdash C}\bot E$$

Figure 13.2: Natural Deduction Rules

```
nd : o -> type.

andI    : nd A -> nd B -> nd (A and B).
andE1   : nd (A and B) -> nd A.
andE2   : nd (A and B) -> nd B.
impI    : (nd A -> nd B) -> nd (A imp B).
impE    : nd A -> nd (A imp B) -> nd B.
orI1    : nd A -> nd (A or B).
orI2    : nd B -> nd (A or B).
orE     : nd (A or B) -> (nd A -> nd C) -> (nd B -> nd C) -> nd C.
notI    : ({p : o} nd A -> nd p) -> nd (not A).
notE    : nd (not A) -> nd A -> nd C.
trueI   : nd (true).
falseE  : nd (false) -> nd C.
```

Figure 13.3: Signature for Natural Deduction Rules

Finally, a rule commonly found in presentations of the sequent calculus is the *cut rule*:

$$\frac{\Gamma \longrightarrow \Delta_1, A \qquad \Gamma, A \longrightarrow \Delta_2}{\Gamma \longrightarrow \Delta_1, \Delta_2}\,cut$$

This is a derived rule, in the sense that derivations that use cut can be shown to always have "cut-free" correspondents (cut-elimination).

In picking a specific flavor of Sequent Calculus, our main criterion is ease of representation. Logical frameworks based on Hereditary Harrop Formulas or LF are traditionally biased toward Natural Deduction, and have some difficulties in representing sequents. Structural rules seem to be the culprit, and therefore efficient formulations of these is highly desirable. Multi-sets, where available, allow one to avoid using exchange explicitly, leading to a calculus with just weakening and contraction. The system that we will use here actually manages to eliminate these two rules too, by embedding the first into an extended version of the initial rule, and distributing the second over the left rules. By doing so we obtain G_3, a formulation for the Sequent Calculus first introduced by Kleene in [35] to obtain a decision procedure for the propositional fragment. The rules for G_3, taken from [61], are shown in Figure 13.4. The right rule for negation, $\neg R^p$, requires the use of a propositional parameter p; in order for the rule to be applicable, we require this not to appear in neither Γ nor A.

Pfenning's clever idea in [61] was to represent G_3 in Twelf using LF contexts to implement the (multi-sets) Γ of left formulas. Hence, we introduce two new types, `hyp` and `conc`, both dependent on objects of type `o`, and we represent a Sequent Calculus derivation as a proof object

$$x_1 : \texttt{hyp}\ \lceil A_1 \rceil, \ldots, x_n : \texttt{hyp}\ \lceil A_n \rceil \vdash_{\Sigma}^{\preceq} M : \texttt{conc}\ \lceil C \rceil$$

(above, we indicated with $\lceil A \rceil$ the Twelf representation of the formula A). As in the case of Natural Deduction, rules of inference are encoded by functional objects which map premises into conclusions. Cases where a formula appears in the left side of an antecedent but not in the consequent are handled by lambda-abstraction. The Twelf signature for Sequent Calculus is given in Figure 13.5.

13.3 From Natural Deduction to Sequent Calculus

The key idea in translating natural deduction proofs into sequent calculus is to translate a derivation $\mathcal{D}$ of $\vdash C$ with undischarged assumptions $\Gamma = A_1, A_2, \ldots, A_n$ into a sequent calculus derivation $\mathcal{G}$ of $\Gamma \longrightarrow C$. In pictures:

$$\left[\begin{array}{c}\mathcal{D}\\ \vdots \\ \vdash C\end{array}\right]_{\Gamma} \quad \Longrightarrow_{\mathcal{R}} \quad \begin{array}{c}\mathcal{G}\\ \vdots \\ \Gamma \longrightarrow C\end{array}$$

Introduction rules translate directly into right rules. For example:

$$\left[\dfrac{\begin{array}{c}\mathcal{D}_1\\ \vdots \\ \vdash C_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \vdots \\ \vdash C_2\end{array}}{\vdash C_1 \wedge C_2}\wedge I\right]_{\Gamma} \quad \Longrightarrow_{\mathcal{R}} \quad \dfrac{\left[\begin{array}{c}\mathcal{D}_1\\ \vdots \\ \vdash C_1\end{array}\right]_{\Gamma} \qquad \left[\begin{array}{c}\mathcal{D}_2\\ \vdots \\ \vdash C_2\end{array}\right]_{\Gamma}}{\Gamma \longrightarrow C_1 \wedge C_2}\wedge R$$

For elimination rules, the antecedent usually becomes the cut formula of the sequent derivation:

$$\left[\dfrac{\begin{array}{c}\mathcal{D}\\ \vdots \\ \vdash A \wedge B\end{array}}{\vdash A}\wedge E_1\right]_{\Gamma} \quad \Longrightarrow_{\mathcal{R}} \quad \dfrac{\left[\begin{array}{c}\mathcal{D}\\ \vdots \\ \vdash A \wedge B\end{array}\right]_{\Gamma} \qquad \dfrac{\overline{\Gamma, A \wedge B, A \longrightarrow A}\,I}{\Gamma, A \wedge B \longrightarrow A}\wedge L_1}{\Gamma \longrightarrow A}\,cut$$

$$\frac{}{\Gamma, A \longrightarrow A}I$$

$$\frac{\Gamma \longrightarrow A \qquad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\wedge R \qquad \frac{\Gamma, A \wedge B, A \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\wedge L_1 \qquad \frac{\Gamma, A \wedge B, B \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\wedge L_2$$

$$\frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B}\supset R \qquad \frac{\Gamma, A \supset B \longrightarrow A \qquad \Gamma, A \supset B, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C}\supset L$$

$$\frac{\Gamma \longrightarrow A}{\Gamma \longrightarrow A \vee B}\vee R_1 \qquad \frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow A \vee B}\vee R_2 \qquad \frac{\Gamma, A \vee B, A \longrightarrow C \qquad \Gamma, A \vee B, B \longrightarrow C}{\Gamma, A \vee B \longrightarrow C}\vee L$$

$$\frac{\Gamma, A \longrightarrow p}{\Gamma \longrightarrow \neg A}\neg R^p \qquad \frac{\Gamma, \neg A \longrightarrow A}{\Gamma, \neg A \longrightarrow C}\neg L$$

$$\frac{}{\Gamma \longrightarrow \top}\top R$$

$$\frac{}{\Gamma, \bot \longrightarrow C}\bot L$$

$$\frac{\Gamma \longrightarrow A \qquad \Gamma, A \longrightarrow C}{\Gamma \longrightarrow C}cut$$

Figure 13.4: Sequent Calculus Rules

```
hyp  : o -> type.
conc : o -> type.

axiom  : hyp A -> conc A.
andl1  : (hyp A -> conc C)
         -> (hyp (A and B) -> conc C).
andl2  : (hyp B -> conc C)
         -> (hyp (A and B) -> conc C).
andr   : conc A
         -> conc B
         -> conc (A and B).
impl   : conc A
         -> (hyp B -> conc C)
         -> (hyp (A imp B) -> conc C).
impr   : (hyp A -> conc B)
          -> conc (A imp B).
orl    : (hyp A -> conc C)
          -> (hyp B -> conc C)
          -> (hyp (A or B) -> conc C).
orr1   : conc A
         -> conc (A or B).
orr2   : conc B
         -> conc (A or B).
notl   : conc A
         -> (hyp (not A) -> conc C).
notr   : ({p : o} hyp A -> conc p)
         -> conc (not A).
truer  : conc (true).
falsel : (hyp (false) -> conc C).

cut : {A : o} conc A
      -> (hyp A -> conc C)
      -> conc C.
```

Figure 13.5: Signature for Sequent Calculus Rules

When we look at the Twelf representations of both calculi, we realize that, in implementing this transformation of a HTRS, we have to be more subtle. The derivation $\mathcal{D}$ is encoded in LF as an object M such that

$$x_1 : \mathbf{nd}\ A_1, \ldots, x_n : \mathtt{nd}\ A_n \vdash^{\prec}_{\Sigma} M : \mathtt{nd}\ C$$

On the other hand, $\mathcal{G}$ is encoded by N, where

$$y_1 : \mathtt{hyp}\ \lceil A_1 \rceil, \ldots, y_n : \mathtt{hyp}\ \lceil A_n \rceil \vdash^{\prec}_{\Sigma} N : \mathtt{conc}\ \lceil C \rceil$$

The translation will then have to be expressed by two complementary operators:

```
trSQ   : nd A -> conc A.
trSQ~1 : hyp A -> nd A.
```

and the term N we are looking for will be obtained by reducing

$$y_1 : \mathtt{hyp}\ \lceil A_1 \rceil, \ldots, y_n : \mathtt{hyp}\ \lceil A_n \rceil \vdash^{\prec}_{\Sigma} (\mathtt{trSQ}\ M)[x_i/(\mathtt{trSQ\tilde{}1}\ y)] : \mathtt{conc}\ \lceil C \rceil$$

into a `trSQ`- and `trSQ~1`-free term.

The complete HTRS is given in Figure 13.6. Note that, since these transformations are given by induction on the last inference used in the derivation, they are strongly-normalizing and trivially confluent (there are no critical overlaps).

13.4 From Sequent Calculus to Natural Deduction

Translating back from sequent calculus to natural deduction is done in a fundamentally similar way. Given a derivation $\mathcal{G}$ of $\Gamma \longrightarrow C$, we would like to obtain a proof $\mathcal{D}$ of $\vdash C$ with undischarged assumptions $\Gamma = A_1, \ldots, A_n$. However, this principle does not scale well to considering left rules; proceeding, as before, in a bottom-up manner, we see that the left rules introduce new formulas in the set Γ. Although it is possible to treat these as fresh assumptions, they are actually logical consequences of pre-existing formulas in Γ. A better representation for the transformation makes use of an assignment $\Gamma := \overline{\mathcal{I}}$ that maps each of the assumptions A_i in Γ into a natural deduction derivation $\mathcal{I}_i$:

$$\left[\begin{array}{c}\mathcal{G}\\ \vdots \\ \Gamma \longrightarrow C\end{array}\right]_{\Gamma:=\overline{\mathcal{I}}} \quad \Longrightarrow_{\mathcal{R}} \quad \begin{array}{c}\mathcal{D}\\ \vdots \\ \vdash C\end{array}$$

In the case of left rules, we extend this assignment to the newly introduced formula by giving a derivation of it, which we construct making use of the other derivations the assignment provides us. To illustrate how this works, we give here the conversion rules for $\wedge L_1$ and $\wedge R$:

$$\left[\begin{array}{c}\mathcal{G}\\ \vdots \\ \dfrac{\Gamma, A \wedge B, A \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}{\scriptstyle \wedge L_1}\end{array}\right]_{\Gamma:=\overline{\mathcal{I}},\, A\wedge B:=\mathcal{I}_0} \quad \Longrightarrow_{\mathcal{R}} \quad \left[\begin{array}{c}\mathcal{G}\\ \vdots \\ \Gamma, A \wedge B, A \longrightarrow C\end{array}\right]_{\Gamma:=\overline{\mathcal{I}},\, A\wedge B:=\mathcal{I}_0,\, A:=\dfrac{\stackrel{\mathcal{I}_0}{\vdash A\wedge B}}{\vdash A}{\scriptstyle \wedge E_1}}$$

$$\left[\begin{array}{c}\begin{array}{cc}\mathcal{G}_1 & \mathcal{G}_2\\ \vdots & \vdots\end{array}\\ \dfrac{\Gamma \longrightarrow C_1 \qquad \Gamma \longrightarrow C_2}{\Gamma \longrightarrow C_1 \wedge C_2}{\scriptstyle \wedge R}\end{array}\right]_{\Gamma:=\overline{\mathcal{I}}} \quad \Longrightarrow_{\mathcal{R}} \quad \dfrac{\left[\begin{array}{c}\mathcal{G}_1\\ \vdots \\ \Gamma \longrightarrow C_1\end{array}\right]_{\Gamma:=\overline{\mathcal{I}}} \qquad \left[\begin{array}{c}\mathcal{G}_2\\ \vdots \\ \Gamma \longrightarrow C_2\end{array}\right]_{\Gamma:=\overline{\mathcal{I}}}}{\vdash C_1 \wedge C_2}{\scriptstyle \wedge I}$$

Representing this transformation in LF is done as before, by defining two complementary operators:

```
tr_var : trSQ (trSQ~1 H) => axiom H.

tr_andI : trSQ (andI D1 D2) => andr (trSQ D1) (trSQ D2).

tr_andE1 : trSQ (andE1 D)
           => cut (A and B) (trSQ D) ([h : hyp (A and B)] andl1 axiom h).

tr_andE2 : trSQ (andE2 D)
           => cut (A and B) (trSQ D) ([h : hyp (A and B)] andl2 axiom h).

tr_impI : trSQ (impI ([d : nd A] D d))
          => impr ([h : hyp A] trSQ (D (trSQ~1 h))).

tr_impE : trSQ (impE D1 D2)
          => cut (A imp B) (trSQ D2)
                 ([h : hyp (A imp B)] impl (trSQ D1) (axiom) h).

tr_orI1 : trSQ (orI1 D1) => orr1 (trSQ D1).

tr_orI2 : trSQ (orI2 D2) => orr2 (trSQ D2).

tr_orE  : trSQ (orE D ([d1 : nd A] D1 d1) ([d2 : nd B] D2 d2))
          => cut (A or B) (trSQ D)
                 ([h : hyp (A or B)] orl ([h1 : hyp A] trSQ (D1 (trSQ~1 h1)))
                                         ([h2 : hyp B] trSQ (D2 (trSQ~1 h2))) h).

tr_notI   : trSQ (notI ([p : o] [d : nd A] D p d))
            => notr ([p : o] [h : hyp A] trSQ (D p (trSQ~1 h))).

tr_notE   : trSQ (notE D1 D2)
            => cut (not A) (trSQ D1) ([h : hyp (not A)] notl (trSQ D2) h).

tr_trueI  : trSQ (trueI) => truer.

tr_falseE : trSQ (falseE D) => cut (false) (trSQ D) falsel.
```

Figure 13.6: Natural Deduction to Sequent Calculus HTRS

```
tr_axiom : trND (axiom (trND~1 G)) => G.

tr_andl1 : trND (andl1 ([h1 : hyp A] D1 h1) (trND~1 G))
           => trND (D1 (trND~1 (andE1 G))).

tr_andl2 : trND (andl2 ([h2 : hyp A] D2 h2) (trND~1 G))
           => trND (D2 (trND~1 (andE2 G))).

tr_andr : trND (andr D1 D2) => andI (trND D1) (trND D2).

tr_impl : trND (impl D1 ([h2 : hyp B] D2 h2) (trND~1 G))
          => trND (D2 (trND~1 (impE (trND D1) G))).

tr_impr : trND (impr ([h : hyp A] D h)) => impI ([d : nd A] trND (D (trND~1 d))).

tr_orl : trND (orl ([h1 : hyp A] D1 h1) ([h2 : hyp B] D2 h2) (trND~1 G))
         => orE G ([d1 : nd A] trND (D1 (trND~1 d1)))
                  ([d2 : nd B] trND (D2 (trND~1 d2))).

tr_orr1 : trND (orr1 D1) => orI1 (trND D1).

tr_orr2 : trND (orr2 D2) => orI2 (trND D2).

tr_notl : trND (notl D (trND~1 G)) => notE G (trND D).

tr_notr : trND (notr ([p : o] [h : hyp A] D p h))
          => notI ([p : o] [d : nd A] trND (D p (trND~1 d))).

tr_truer : trND (truer) => trueI.

tr_falsel : trND (falsel (trND~1 G)) => falseE G.
```

Figure 13.7: Sequent Calculus to Natural Deduction HTRS

```
trND   : conc A -> nd A.
trND~1 : nd A -> hyp A.
```

The inverse operator `trND~1` takes here a more active role than before, since we use it to implement the assignment described before. The non-overlapping, strongly confluent HTRS that we obtain is listed in Figure 13.7.

13.5 Cut Elimination

The missing piece to our construction is the representation of a cut-elimination theorem as a confluent HTRS. In this section, we will adopt the proof presented by Pfenning in [61], turning it into a set of rewrite rules. Using an existing proof as a starting point has the advantage of freeing us from the burden of providing a termination ordering for the HTRS we produce: the same induction principle used by the proof will suffice in showing strong normalization. We are left to examine problems of confluence, which, however, given our background, we are better equipped to handle.

In constructing our HTRS, we start from the following:

Theorem 13.5.1 (Admissibility of Cut). *Let $\mathcal{D}$ and $\mathcal{G}$ be cut-free derivations of $\Gamma \longrightarrow A$ and $\Gamma, A \longrightarrow C$, respectively, then there exists a cut-free derivation of $\Gamma \longrightarrow C$.*

This result allow us to see every instance of cut as a syntactic shorthand for a larger, cut-free derivation. From this perspective, a HTRS can be easily formulated as a list of directed definitional equalities, replacing the shorthand with the proof it stands for. Places in the proof where the inductive hypothesis is invoked on sub-derivations of $\mathcal{D}$ or $\mathcal{G}$ translate in our notation into (recursive) applications of cut, to be eventually eliminated by repeated applications of the rewriting rules.

The proof of Theorem 13.5.1, and correspondingly, our higher-order rules, can be divided into four classes

1. Initial Conversions

   At least one of the derivation consists of a single application of the initial rule;

2. Essential Conversions: in both derivation, the cut formula is also the *principal formula* of the last inference step, i.e. the one that the inference rule operates upon;

3. Left Commutative Conversions: the cut formula is non-principal in $\mathcal{D}$;

4. Right Commutative Conversions: the cut formula is non-principal in $\mathcal{G}$.

Translating the Pfenning's proof into rewrite rules, we obtain the system shown in Figures 13.9-13.12.

Since the four classes above are not mutually exclusive, the system has critical overlaps. Unfortunately, not all of these turn out to be trivial: specifically, critical pairs coming from overlapping left and right commutative conversions cannot be joined.

To complete the system by orienting critical pairs is not directly feasible, as there are rules like `ri1` (see Figure 13.9) that have non-pattern right-hand-sides, and hence generate non-pattern pairs. It is possible, however, to come up with a slightly different proof for cut-elimination where all rules have pattern RHSs. Pursuing this technique led us to a system that could be completed, although not necessarily by a finite number of rules: orienting critical pairs generated rules that admitted new critical overlaps, that had to be oriented in rules, and so on. The completion process seemed to diverge, which discouraged any further attempt in this direction.

For this reason we investigated another solution, consisting in eliminating non-trivial critical pairs in the original system by removing some of the non-deterministic choices in the cut-elimination process. Specifically, we imposed that right commutative conversions could be applied in case all other cases fail, i.e. only to those cases where the cut formula is the principal formula in $\mathcal{D}$.

Figure 13.8 shows how all four possible cases of cut-elimination of non-initial derivations are handled.

	Principal in $\mathcal{G}$	Non-Principal in $\mathcal{G}$
Principal in $\mathcal{D}$	Essential	Right-Commutative
Non-Principal in $\mathcal{D}$	Left-Commutative	Left-Commutative

Figure 13.8: Cut Elimination for Non-Initial Derivations

Translating these ideas into our HTRS, what we need to do is to specialize each rule of Figure 13.12 to those cases where the derivation `D` ends with a right rule. For example,

```
rr3  : cut A D ([h : hyp A] orr1 (G1 h)) => orr1 (cut A D G1).
```

is converted into

```
rr3_1 : cut (A1 and A2) (andr D1 D2) ([h : hyp (A1 and A2)] orr1 (G1 h))
        => orr1 (cut (A1 and A2) (andr D1 D2) G1).
rr3_2 : cut (A1 imp A2) (impr D1) ([h : hyp (A1 imp A2)] orr1 (G1 h))
        => orr1 (cut (A1 imp A2) (impr D1) G1).
rr3_3 : cut (A1 or A2) (orr1 D1) ([h : hyp (A1 or A2)] orr1 (G1 h))
        => orr1 (cut (A1 or A2) (orr1 D1) G1).
rr3_4 : cut (A1 or A2) (orr2 D2) ([h : hyp (A1 or A2)] orr1 (G1 h))
        => orr1 (cut (A1 or A2) (orr2 D2) G1).
```

```
rr3_5 : cut (not A1) (notr D1) ([h : hyp (not A1)] orr1 (G1 h))
        => orr1 (cut (not A1) (notr D1) G1).
rr3_6 : cut (true) (truer) ([h : hyp (true)] orr1 (G1 h))
        => orr1 (cut (true) (truer) G1).
```

The resulting HTRS admits has a fairly large number of critical pairs, 7459 to be precise, but they all can be shown to be trivial.

```
ri1 : cut A (axiom H) G => (G H).

ri2 : cut A D ([h : hyp A] axiom h) => D.
```

Figure 13.9: Cut Elimination - Initial Rules

```
re1 : cut (A1 and A2) (andr D1 D2)
          ([h : hyp (A1 and A2)] andl1 ([h1 : hyp A1] (G1 h1 h)) h)
      => cut A1 D1 ([h1 : hyp A1] cut (A1 and A2) (andr D1 D2) (G1 h1)).

re2 : cut (A1 and A2) (andr D1 D2)
          ([h : hyp (A1 and A2)] andl2 ([h2 : hyp A2] (G2 h2 h)) h)
      => cut A2 D2 ([h2 : hyp A2] cut (A1 and A2) (andr D1 D2) (G2 h2)).

re3 : cut (A1 imp A2) (impr D1)
          ([h : hyp (A1 imp A2)] impl (G1 h) ([h2 : hyp A2] G2 h2 h) h)
      => cut A2 (cut A1 (cut (A1 imp A2) (impr D1) G1) D1)
                  ([h2 : hyp A2] cut (A1 imp A2) (impr D1) (G2 h2)).

re4 : cut (A1 or A2) (orr1 D1)
          ([h : hyp (A1 or A2)] orl ([h1 : hyp A1] G1 h1 h)
                                    ([h2 : hyp A2] G2 h2 h) h)
      => cut A1 D1 ([h1 : hyp A1] cut (A1 or A2) (orr1 D1) (G1 h1)).

re5 : cut (A1 or A2) (orr2 D2)
          ([h : hyp (A1 or A2)] orl ([h1 : hyp A1] G1 h1 h)
                                    ([h2 : hyp A2] G2 h2 h) h)
      => cut A2 D2 ([h2 : hyp A2] cut (A1 or A2) (orr2 D2) (G2 h2)).

re6 : cut (not A1) (notr D1)
          ([h : hyp (not A1)] notl (G1 h) h)
      => cut A1 (cut (not A1) (notr D1) G1) (D1 C).
```

Figure 13.10: Cut Elimination - Essential Rules

```
rl1 : cut A (andl1 D1 H) G => andl1 ([h1 : hyp A1] cut A (D1 h1) G) H.

rl2 : cut A (andl2 D2 H) G => andl2 ([h2 : hyp A2] cut A (D2 h2) G) H.

rl3 : cut A (impl D1 D2 H) G => impl D1 ([h2 : hyp A2] cut A (D2 h2) G) H.

rl4 : cut A (orl D1 D2 H) G
      => orl ([h1 : hyp A1] cut A (D1 h1) G) ([h2 : hyp A2] cut A (D2 h2) G) H.

rl5 : cut A (notl D H) G => notl D H.

rl6 : cut A (falsel H) G => falsel H.
```

Figure 13.11: Cut Elimination - Left-commutative Rules

```
rr0  : cut A D ([h : hyp A] axiom H1) => axiom H1.

rr1  : cut A D ([h : hyp A] andr (G1 h) (G2 h)) => andr (cut A D G1) (cut A D G2).

rr2  : cut A D ([h : hyp A] impr ([h1 : hyp C1] G1 h1 h))
       => impr ([h1 : hyp C1] cut A D (G1 h1)).

rr3  : cut A D ([h : hyp A] orr1 (G1 h)) => orr1 (cut A D G1).

rr4  : cut A D ([h : hyp A] orr2 (G2 h)) => orr2 (cut A D G2).

rr5  : cut A D ([h : hyp A] notr (G h))
       => notr ([p : o][h1 : hyp C1] cut A D ([h : hyp A] G h p h1)).

rr6  : cut A D ([h : hyp (A1 and A2)] truer) => truer.

rr7  : cut A D ([h : hyp A] andl1 ([h1 : hyp B1] (G1 h1 h)) H)
       => andl1 ([h1 : hyp B1] cut A D (G1 h1)) H.

rr8  : cut A D ([h : hyp A] andl2 ([h2 : hyp B2] (G2 h2 h)) H)
       => andl2 ([h2 : hyp B2] cut A D (G2 h2)) H.

rr9  : cut A D ([h : hyp A] impl (G1 h) ([h2 : hyp B2] G2 h2 h) H)
       => impl (cut A D G1) ([h2 : hyp B2] cut A D (G2 h2)) H.

rr10 : cut A D ([h : hyp A] orl ([h1 : hyp B1] G1 h1 h) ([h2 : hyp B2] G2 h2 h) H)
       => orl ([h1 : hyp B1] cut A D (G1 h1)) ([h2 : hyp B2] cut A D (G2 h2)) H.

rr11 : cut A D ([h : hyp A] notl (G1 h) H) => notl (cut A D G1) H.

rr12 : cut A D ([h : hyp A] falsel H) => falsel H.
```

Figure 13.12: Cut Elimination - Right-commutative Rules

Index

Bibliography

[1] Andrea Asperti and Giuseppe Longo. Categories, types, and structures: An introduction to category theory for the working computer scientist (MIT press, 1991). *SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory)*, 22, 1991.

[2] Jürgen Avenhaus, Carlos Loría Sáenz, and Joachim Steinbach. A reduction ordering for higher-order terms. Technical Report SR-95-03, Fachbereich Informatik, Universität Kaiserlautern, 1995.

[3] Franz Baader and Tobias Nipkow. *Rewriting and All That*. Cambridge University Press, Shaftesbury Road, Cambridge, England, 1998.

[4] Leo Bachmair and Nachum Dershowitz. Completion for rewriting modulo a congruence. *TCS: Theoretical Computer Science*, 67, 1989.

[5] Franco Barbanera and Maribel Fernández. Modularity of termination and confluence in combinations of rewrite systems with λ_ω. In *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 1993.

[6] Franco Barbanera, Maribel Fernández, and Herman Geuvers. Modularity of strong normalization and confluence in the algebraic lambda-cube. In *LICS: IEEE Symposium on Logic in Computer Science*, 1994.

[7] Hendrik Pieter Barendregt. *The Lambda-Calculus: Its Syntax and Semantics*. North-Holland, Amsterdam, 1980.

[8] Hendrik Pieter Barendregt. Lambda calculi with types. In *Handbook of Logic in Computer Science, Volumes 1 (Background: Mathematical Structures) and 2 (Background: Computational Structures), Abramsky & Gabbay & Maibaum (Eds.), Clarendon*, volume 2. Oxford University Press, 1992.

[9] Val Breazu-Tannen. Combining algebra and higher-order types. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 82–90, 1988.

[10] Manuel Clavel, Francisco Duràn, Steven Eker, and Jose Meseguer. Principles of Maude. In *Proceedings of the 1st International Workshop on Rewriting Logic and its Applications*, Pacific Grove, California, September 1996.

[11] Robert Constable and Douglas Howe. NuPrl as a general logic. In P. Odifreddi, editor, *Logic and Computation*. Academic Press, 1990.

[12] Roberto Di Cosmo. On the power of simple diagrams. In Harald Ganzinger, editor, *Proceedings of the 7th International Conference on Rewriting Techniques and Applications*, pages 200–214, New Brunswick, New Jersey, July 1996. Springer-Verlag LNCS 1103.

[13] Pierre-Louis Curien. *Categorical Combinators, Sequential Algorithms and Functional Programming*. Progress in Theoretical Computer Science. Birkhauser, 1993.

[14] Nachum Dershowitz. Orderings for term-rewriting systems. *TCS: Theoretical Computer Science*, 17, 1982.

[15] M. Dincbas, P. Van Hentenryck, H. Simonis, A. Aggoun, T. Graf, and F. Berthier. The constraint logic programming language CHIP. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, Tokyo, Japan, December 1988.

[16] Gilles Dowek, Amy Felty, Hugo Herbelin, Gérard Huet, Chet Murthy, Catherine Parent, Christine Paulin-Mohring, and Benjamin Werner. The Coq proof assistant user's guide. Rapport Techniques 154, INRIA, Rocquencourt, France, 1993. Version 5.8.

[17] Gilles Dowek, Gérard Huet, and Benjamin Werner. On the definition of the eta-long normal form in the type systems of the cube Calculus of Constructions. In Herman Geuvers, editor, *Informal Proceedings of the Workshop on Types for Proofs and Programs*, Nijmegen, The Netherlands, May 1993.

[18] Thom Früwirth. Theory and practice of Constraint Handling Rules. *Journal of Logic Programming*, 37(1-3):95–138, October 1998.

[19] Wolfgang Gehrke. *Decidability Results for Categorical Notions Related to Monads by Rewriting Techniques*. PhD thesis, Research Institute for Symbolic Computation, Linz, Austria, May 1995. Available as RISC Report Number 95-30.

[20] Wolfgang Gehrke and Frank Pfenning. ElfRW distribution, 1996.

[21] Gerhard Gentzen. Investigations into logical deduction. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*. North-Holland, 1969.

[22] Herman Geuvers. The Church-Rosser property for $\beta\eta$-reduction in typed λ-calculi. In A. Scedrov, editor, *Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 453–460, Santa Cruz, California, June 1992.

[23] Neil Ghani. Eta-expansions in dependent type theory—the Calculus of Constructions. In R. Hindley, editor, *Proceedings fo the Third International Conference on Typed Lambda Calculus and Applications (TLCA'97)*, Nancy, France, April 1997. Springer-Verlag LNCS.

[24] Joseph Goguen, Claude Kirchner, Helene Kirchner, Aristide Mégrelis, and José Meseguer. An introduction to OBJ3. In Jean-Pierre Jouannaud and Stephane Kaplan, editors, *Proceedings of the Workshop on Conditional Term Rewriting Systems (Orsay, France)*, pages 258–263. Springer-Verlag LNCS 308, Berlin, 1988.

[25] Thérèse Hardin and Alain Laville. Proof of termination of the rewriting system SUBST on CCL. *Theoretical Computer Science*, 46:305–312, 1986.

[26] Robert Harper. An equational formulation of LF. Technical Report ECS-LFCS-88-67, University of Edinburgh, October 1988.

[27] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. In *Symposium on Logic in Computer Science*, pages 194–204. IEEE Computer Society Press, June 1987.

[28] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinery*, 40(1):143–184, January 1993.

[29] Gérard Huet. Confluent reductions: abstract properties and applications to term rewrite systems. *Journal of the Association for Computing Machinery*, 27(4):787–821, October 1980.

[30] Gérard Huet. A complete proof of correctness of the Knuth-Bendix completeness algorithm. *JCSS: Journal of Computer and System Sciences*, 23, 1981.

[31] Gérard Huet and D. C. Oppen. *Equations and Rewrite Rules: A Survey*, pages 349–405. Academic Press, New York, 1980.

[32] Joxan Jaffar, Spiro Michayov, Peter Stuckey, and Roland Yap. The CLP($\mathcal{R}$) language and system. *ACM Transactions on Programming Languages and Systems*, 14(3):339–395, July 1992.

[33] Jean-Pierre Jouannaud and Albert Rubio. Rewrite orderings for higher-order terms in η-long β-normal form and the recursive path ordering. *Theoretical Computer Science*, 208:33–58, 1998.

[34] Stefan Kahrs. Towards a domain theory for termination proofs. In Jieh Hsiang, editor, *Proceedings of the Sixth International Conference on Rewriting Techniques and Applications*, pages 241–255, Kaiserslautern, Germany, April 1995. Springer-Verlag LNCS 914.

[35] Stephen C. Kleene. *Introduction to Metamathematics*. D. van Nostrand, Princeton, New Jersey, 1952.

[36] Jan Willem Klop. Term rewriting systems: a tutorial. *Bulletin of the European Association for Theoretical Computer Science*, 32:143–182, June 1987.

[37] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebra. In John Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[38] Zhaohui Luo and Robert Pollack. The LEGO proof development system: A user's manual. Technical Report ECS-LFCS-92-211, University of Edinburgh, May 1992.

[39] Olav Lysne and Javier Piris. A termination ordering for higher order rewrite systems. In Jieh Hsiang, editor, *Proceedings of the Sixth International Conference on Rewriting Techniques and Applications*, pages 26–40, Kaiserslautern, Germany, April 1995. Springer-Verlag LNCS 914.

[40] Narciso Martì-Oliet and Jose Meseguer. Rewriting logic as a logical and semantical framework. Technical Report SRI-CSL-93-05, SRI International, August 1993.

[41] Per Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology and Philosophy of Science VI*, pages 153–175. North-Holland, 1980.

[42] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. Technical Report 2, Scuola di Specializzazione in Logica Matematica, Dipartimento di Matematica, Università di Siena, 1985.

[43] Per Martin-Löf. Truth of a propositions, evidence of a judgement, validity of a proof. Notes to a talk given at the workshop *Theory of Meaning*, Centro Fiorentino di Storia e Filosofia della Scienza, June 1985.

[44] Spiro Michaylov and Frank Pfenning. An empirical study of the runtime behavior of higher-order logic programs. In D. Miller, editor, *Proceedings of the Workshop on the λProlog Programming Language*, pages 257–271, Philadelphia, Pennsylvania, July 1992. University of Pennsylvania. Available as Technical Report MS-CIS-92-86.

[45] Spiro Michaylov and Frank Pfenning. Higher-order logic programming as constraint logic programming. In *Position Papers for the First Workshop on Principles and Practice of Constraint Programming*, pages 221–229, Newport, Rhode Island, April 1993. Brown University.

[46] Dale Miller. A theory of modules for logic programming. In Robert M. Keller, editor, *Third Annual IEEE Symposium on Logic Programming*, pages 106–114, Salt Lake City, Utah, September 1986.

[47] Dale Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. In Peter Schroeder-Heister, editor, *Proceedings of the International Workshop on Extensions of Logic Programming*, pages 253–281, Tübingen, Germany, 1989. Springer-Verlag LNAI 475.

[48] Robin Milner. Calculi for synchrony and asynchrony. *TCS: Theoretical Computer Science*, 25, 1983.

[49] Robin Milner. *Communication and Concurrency*. Prentice Hall, New York, 1989.

[50] Robin Milner. Action structures and the π-calculus. In *Proceedings of the NATO Summer School on Logic and Computation*. Springer-Verlag, 1993.

[51] George C. Necula. *Compiling with Proofs.* PhD thesis, School of Computer Science, Carnegie Mellon University, September 1998. Available as Technical Report CMU-CS-98-154.

[52] R.P. Nederpelt. *Strong Normalization in a Typed Lambda Calculus with Lambda Structured Types.* PhD thesis, Eindhoven University of Technology, 1973.

[53] M. H. A. Newman. On theories with a combinatorial definition of equivalence. *Annals of Mathematics*, 43(2):223–243, 1942.

[54] Tobias Nipkow. Higher-order critical pairs. In G. Kahn, editor, *Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 342–349, Amsterdam, The Netherlands, July 1991.

[55] Tobias Nipkow. Orthogonal higher-order rewrite systems are confluent. In M. Bezem and J.F. Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, pages 306–317, Utrecht, The Netherlands, May 1993.

[56] Tobias Nipkow and Zhenyu Qian. Modular higher-order E-unification. In R. V. Book, editor, *Proc. 4th Int. Conf. Rewriting Techniques and Applications*, pages 200–214. Springer-Verlag LNCS 488, 1991.

[57] Lawrence C. Paulson. Isabelle: The next 700 theorem provers. In P. Odifreddi, editor, *Logic and Computer Science*, pages 361–386. Academic Press, 1990.

[58] Gerald E. Peterson and Mark E. Stickel. Complete sets of reductions for some equational theories. *Journal of the ACM*, 28(2):233–264, April 1981.

[59] Frank Pfenning. Logic programming in the LF logical framework. In Gérard Huet and Gordon Plotkin, editors, *Logical Frameworks*, pages 149–181. Cambridge University Press, 1991.

[60] Frank Pfenning. Computation and deduction. Unpublished lecture notes, 277 pp. Revised May 1994, April 1996, May 1992.

[61] Frank Pfenning. A structural proof of cut elimination and its representation in a logical framework. Technical Report CMU-CS-94-218, Department of Computer Science, Carnegie Mellon University, November 1994.

[62] Frank Pfenning and Conal Elliott. Higher-order abstract syntax. In *Proceedings of the ACM SIGPLAN '88 Symposium on Language Design and Implementation*, pages 199–208, Atlanta, Georgia, June 1988.

[63] Frank Pfenning and Carsten Schürmann. Twelf user's guide. Technical report, Carnegie Mellon University, School of Computer Science, 1998. Available as Technical Report CMU-CS-98-173, Carnegie Mellon University.

[64] Dag Prawitz. *Natural Deduction: a Proof-Theoretical Study.* Almquist and Wiskell, 1965.

[65] Christian Prehofer. Higher-order narrowing. In *Proceedings of the Ninth Annual IEEE Symposium on Logic in Computer Science*, pages 507–516, Paris, France, July 1994. IEEE Computer Society Press.

[66] Christian Prehofer. *Solving Higher-Order Equations: From Logic to Programming.* PhD thesis, Technische Universität München, March 1995.

[67] Anne Salvesen. The Church-Rosser theorem for LF with $\beta\eta$-reduction. Unpublished notes to a talk given at the First Workshop on Logical Frameworks in Antibes, France, May 1990.

[68] J. van de Pol and H. Schwichtenberg. Strict functionals for termination proofs. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, pages 350–364, Edinburgh, Scotland, April 1995. Springer-Verlag LNCS 902.

[69] Maartin H. van Emden and Robert Kowalski. The semantics of predicate logic as a programming language. *Journal of the Association for Computing Machinery*, 23(4):733–742, 1976.

[70] Vincent van Oostrom and Femke van Raamsdonk. Comparing combinatory reduction systems and higher-order rewrite systems. Technical Report IR-333, Vrije Universiteit Amsterdam, Computer Science, August 1993.

[71] Roberto Virga. Higher-order superposition for dependent types. In Harald Ganzinger, editor, *Proceedings of the 7th International Conference on Rewriting Techniques and Applications*, pages 123–137, New Brunswick, New Jersey, July 1996. Springer-Verlag LNCS 1103. Extended version available as Technical Report CMU-CS-95-150, May 1995.

[72] Roberto Virga. Twelf(X): Extending Twelf to rationals and beyond. Forthcoming, 1999.

[73] David H. D. Warren. An abstract PROLOG instruction set. Technical Report 309, Artificial Intelligence Center, Computer Science and Technology Division, SRI International, Menlo Park, CA, October 1983.

[74] David A. Wolfram. Rewriting, and equational unification: The higher-order cases. In Ronald V. Book, editor, *Proceedings of the Fourth International Conference on Rewriting Techniques and Applications*, pages 25–36, Como, Italy, April 1991. Springer-Verlag LNCS 488.