Cut-Elimination for a Logic with Definitions and Induction

Raymond McDowell
Department of Computer Science
Widener University
Chester, PA 19013 USA

Dale Miller
Department of Computer Science and Engineering
200 Pond Laboratory
The Pennsylvania State University
University Park, PA 16802 USA

May 15, 1998

Abstract

In order to reason about specifications of computations that are given via the proof search or logic programming paradigm one needs to have at least some forms of induction and some principle for reasoning about the ways in which terms are built and the ways in which computations can progress. The literature contains many approaches to formally adding these reasoning principles with logic specifications. We choose an approach based on the sequent calculus and design an intuitionistic logic $FO\lambda^{\Delta\mathbb{N}}$ that includes natural number induction and a notion of definition. We have detailed elsewhere that this logic has a number of applications. In this paper we prove the cut-elimination theorem for $FO\lambda^{\Delta\mathbb{N}}$, adapting a technique due to Tait and Martin-Löf. This cut-elimination proof is technically interesting and significantly extends previous results of this kind.

1 Introduction

As one attempts to prove a given sequent by placing above it an inference rule, zero or more unproven sequents will arise for the premise of the inference rule and these sequents will, in general, involve some different formulas than the conclusion sequent. Such changes in sequents during the search for a proof have been used to provide a rich and flexible framework for the specification of a wide range of computations. Of course, to make proof search resemble a computational process, the cut rule needs to be avoided; that is, when attempting to model a computation by constructing a proof, it seems sensible not to oblige the search to also search for lemmas to establish. The search for lemmas is part of the creative activity of mathematicians when they look for proofs and does not seem part of the notion of mechanical computation. The cut-elimination theorem, when available, could be used to argue that the search for cut-free proofs is a complete proof procedure. The logic programming paradigm can be defined, at least abstractly, using this notion of proof search, although a further restriction on the search of proofs is often made. In particular, the notion of “goal-directed search” that seems to be a natural aspect of the logic programming paradigm has been formulated using the technical notion of *uniform proof* [18, 17]. To retain completeness of uniform proofs, restrictions on logical formulas need to maintained. For example, completeness of uniform proofs can be achieved in classical logic by restricting to Horn clauses [19]; in intuitionistic logic by restricting to hereditary Harrop formulas [18]; and in linear logic by choosing the proper logical connectives [1, 17].

There are numerous examples of specifying computations within these logics and with using meta-theoretic properties of those logics to infer properties of computations. We only mention a few of these examples here. Intuitionistic logic has been used to specify both the dynamic and static semantics of functional programming languages [10], and theorems that relate these two semantics (such as subject reduction or type preservation) are rendered as simple consequences of the proof theory of intuitionistic logic [14]. In [17], various linear logic encodings of simple objects with state are given and proved equivalent within linear logic. Also in that paper, a small programming language with references is specified, and techniques for proving the equivalence of two programs are given based on resolution within linear logic. In [2], Chirimar provides two specifications of the operational semantics of the DLX RISC processor [20], one capturing its sequential, machine code semantics and the other capturing its concurrent, pipe-lined semantics. Using simple properties of proofs in linear logic, he is able to formally show the equivalence of those two specifications.

Moving from the classical theory of Horn clauses (the logical foundations of Prolog) to all of linear logic (as in the Forum specification language [17]) greatly increases the expressive power of the logic programming paradigm. While Horn clauses are, of course, powerful enough to specify all computations, such specifications need to represent most of the dynamics of a computation within atomic formulas, that is, within the non-logical layer of the language. As a result, deep properties of the ambient logic, such as cut-elimination, are of only limited use when reasoning about Horn clause specifications since such properties only supply meaning for the logical constants. As more expressive logics are used, more dynamics of a computation can be captured by various aspects of the logic, and this increases the likelihood that properties of the logic can be used to derive properties of the specified computations.

There is a difference, however, between specifying a computation and reasoning about a computation, and, in particular, reasoning about computation often requires induction and some way to considering all possible paths that a given computation could proceed or a given object could have been constructed. In the literature, there have been various approaches to providing for these missing features. Within type theory, for example, induction over data structures and over proofs can be used for reasoning about computations [21]. Within logic programming, there are various ways to turn the closed world assumption into a proof principle, such as SLDNF [4]. In this paper, we consider another approach that introduces new inference rules into the sequent calculus of intuitionistic logic. In particular, we add to the sequent calculus a rule for induction on natural numbers and inference rules for treating logic specifications as *definitions* instead of as *theories*. Our approach to definitions follows lines developed by Schroeder-Heister [25], Eriksson [5], Girard [9], and Stärk [28].

Our needs for reasoning about specifications, however, forced us to develop a single extension to intuitionistic logic, called $FO\lambda^{\Delta\mathbb{N}}$ (pronounced “fold-n”), that goes beyond the logics studied in previous works. In particular, we needed one logic that allowed for not only induction and definitions but also for higher-order quantification (but not predicate quantification) since we wished to treat *higher-order abstract syntax* [23]. When designing a new sequent calculus to be used for reasoning, it is important to establish a cut-elimination theorem since this one result can be used to show the consistency of the logic as well as that the consequence relation is closed under modus ponens. The key features of $FO\lambda^{\Delta\mathbb{N}}$ (induction, definitions, and higher-order quantification) interact in complicated ways, so previous cut-elimination proofs for logics with these features in isolation do not carry over to this new system. The bulk of this paper is a presentation of a proof of cut-elimination for $FO\lambda^{\Delta\mathbb{N}}$.

The paper is organized as follows. In the next section, we briefly describe some uses that have been made of $FO\lambda^{\Delta\mathbb{N}}$. In section 3 we introduce the logic and some of its basic properties. We proceed in Section 4 to give an overview of the cut-elimination proof. Section 5 specifies the reduction rules that will be used to eliminate applications of the cut rule. This is followed by a section which provides some auxiliary definitions and their properties. Section 7 contains the proof of cut-elimination. We conclude with a brief comparison of our work with related work of Martin-Löf and Schroeder-Heister.

2 Applications of $FO\lambda^{\Delta\mathbb{N}}$

One use of $FO\lambda^{\Delta\mathbb{N}}$ has been to reason about Horn clause programs. For example, Horn clauses can be used to specify a predicate that relates a list to its length and another predicate that relates two lists if they are permutations of each other. It is an easy matter to give a proof in $FO\lambda^{\Delta\mathbb{N}}$ that if two lists are permutations of each other, then those two lists have the same length [13, Chapter 2]. Many similar theorems can be found throughout McDowell's dissertation [13].

As we shall see, the integration of definitions into sequent calculus makes it possible to perform a case analysis on possible ways that a specified computation can progress. If exploited properly, it is possible to capture notions such as simulation and bisimulation between two processes. The paper [16] shows how this can be accomplished in abstract transition systems and CCS.

A final area where $FO\lambda^{\Delta\mathbb{N}}$ has been used to reason about specifications is in the area of logical frameworks and higher-order abstract syntax. Logical frameworks have been successfully used to give high-level, modular, and formal specifications of many important judgments in the area of programming languages and inference systems. These judgments, such as "the term M denotes a program", "the program M evaluates to the value V", and "the program M has type T", are represented by predicates in the specification logic or by types in a dependent type calculus. One of the advantages of such formal specifications is that they allow logical and mathematical analyses to be used to prove properties about the specified systems. Given the specification of evaluation for a functional programming language, for example, we may wish to prove that the language is deterministic or that evaluation preserves types.

One challenge in reasoning about such specifications centers on the use of higher-order abstract syntax, an elegant and declarative encoding of abstraction and substitution [23]. With most approaches to syntactic representation, the details of variable binding and substitution must be carefully addressed throughout a specification, and theorems about substitution and bound variables can dominate the system analyses. With higher-order abstract syntax, on the other hand, these features are specified concisely and their basic properties follow immediately from the specification logic. However, reasoning within a logical framework about systems represented in higher-order abstract syntax has been difficult since the logics that support this notion of syntax do not provide facilities for the fundamental operations of case analysis and induction. Moreover, higher-order abstract syntax leads to types and recursive definitions that do not give rise to monotone inductive operators, making inductive principles difficult to find. In [14] the authors have shown that these difficulties can be overcome within $FO\lambda^{\Delta\mathbb{N}}$. See [13, 14] for more on how $FO\lambda^{\Delta\mathbb{N}}$ can be used as a meta-logic for an intuitionistic and linear logical framework.

The Pi derivation editor of Eriksson [6] was designed for the finitary calculus of partial inductive definitions [5]. Because of $FO\lambda^{\Delta\mathbb{N}}$'s close relationship with the finitary calculus of partial inductive definitions, the Pi editor can be used to construct $FO\lambda^{\Delta\mathbb{N}}$ derivations. The many examples of specifications and proofs in $FO\lambda^{\Delta\mathbb{N}}$ reported in [13, 14] were constructed using this editor.

Dependent typed λ-calculi have been used to specify computations in ways analogous to the logic programming setting presented here [22]. Schürmann and Pfenning [26] presented a meta-logic for such a dependent typed λ-terms that can be used to reason about higher-order deductions in ways similar to uses of $FO\lambda^{\Delta\mathbb{N}}$ [14].

3 The Logic $FO\lambda^{\Delta\mathbb{N}}$

The basic logic is an intuitionistic version of a subset of Church's Simple Theory of Types [3] in which meta-level formulas will be given the type o. The logical connectives are $\bot$, $\top$, $\wedge$, $\vee$, $\supset$, $\forall_\tau$, and $\exists_\tau$. The quantification types τ (and thus the types of variables) are restricted to not contain o. Thus $FO\lambda^{\Delta\mathbb{N}}$ supports quantification over higher-order (non-predicate) types, a crucial feature for higher-order abstract syntax, but has a first-order proof theory since there is no quantification over predicate types. We will use sequents of the form $\Gamma \longrightarrow B$, where Γ is a finite multiset of formulas and B is a single formula. The basic inference rules for the logic are shown in Table 1. In the $\forall\mathcal{R}$ and $\exists\mathcal{L}$ rules, y is an eigenvariable that is not free in the lower sequent of the rule. The multicut (mc) rule is a generalization of cut due to Slaney [27], and is used to simplify the presentation of the cut-elimination proof.

We introduce the natural numbers via the constants $z : nt$ for zero and $s : nt \to nt$ for successor and the predicate $nat : nt \to o$. The right and left rules for this new predicate are

$$\frac{}{\Gamma \longrightarrow nat\ z}\ nat\mathcal{R} \qquad\qquad \frac{\Gamma \longrightarrow nat\ I}{\Gamma \longrightarrow nat\ (s\ I)}\ nat\mathcal{R}$$

$$\frac{\longrightarrow B\,z \quad B\,j \longrightarrow B\,(s\ j) \quad B\,I, \Gamma \longrightarrow C}{nat\ I, \Gamma \longrightarrow C}\ nat\mathcal{L} \quad .$$

In the left rule, the predicate $B : nt \to o$ represents the property that is proved by induction, and j is an eigenvariable that is not free in B. The third premise of that inference rule witnesses the fact that, in general, B will express a property stronger than $(\bigwedge \Gamma) \supset C$.

Because the induction predicate B in the $nat\mathcal{L}$ rule is not necessarily a subformula of the formula C or any formula in Γ, the subformula property does not hold for $FO\lambda^{\Delta\mathbb{N}}$. In fact, we can derive the following inference rule from the induction rule:

$$\frac{\longrightarrow B \quad B, \Gamma \longrightarrow C}{nat\ I, \Gamma \longrightarrow C} \quad .$$

This derived rule resembles the cut rule but requires a nat assumption. Although $FO\lambda^{\Delta\mathbb{N}}$ does not have the subformula property, the cut-elimination theorem still provides a strong basis for reasoning about proofs in $FO\lambda^{\Delta\mathbb{N}}$ [13, 14, 16]. In fact, the formulation of the $nat\mathcal{L}$ rule and the failure of the subformula property reflect the fact that in actual mathematical practice, finding the

Table 1: Inference rules for the core of $FO\lambda^{\Delta\mathbb{N}}$

$$\frac{}{\bot, \Gamma \longrightarrow B}\ \bot\mathcal{L} \qquad \frac{}{\Gamma \longrightarrow \top}\ \top\mathcal{R}$$

$$\frac{B, \Gamma \longrightarrow D}{B \wedge C, \Gamma \longrightarrow D}\ \wedge\mathcal{L} \qquad \frac{C, \Gamma \longrightarrow D}{B \wedge C, \Gamma \longrightarrow D}\ \wedge\mathcal{L} \qquad \frac{B[t/x], \Gamma \longrightarrow C}{\forall x.B, \Gamma \longrightarrow C}\ \forall\mathcal{L}$$

$$\frac{\Gamma \longrightarrow B \quad \Gamma \longrightarrow C}{\Gamma \longrightarrow B \wedge C}\ \wedge\mathcal{R} \qquad \frac{\Gamma \longrightarrow B[y/x]}{\Gamma \longrightarrow \forall x.B}\ \forall\mathcal{R}$$

$$\frac{B, \Gamma \longrightarrow D \quad C, \Gamma \longrightarrow D}{B \vee C, \Gamma \longrightarrow D}\ \vee\mathcal{L} \qquad \frac{B[y/x], \Gamma \longrightarrow C}{\exists x.B, \Gamma \longrightarrow C}\ \exists\mathcal{L}$$

$$\frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow B \vee C}\ \vee\mathcal{R} \qquad \frac{\Gamma \longrightarrow C}{\Gamma \longrightarrow B \vee C}\ \vee\mathcal{R} \qquad \frac{\Gamma \longrightarrow B[t/x]}{\Gamma \longrightarrow \exists x.B}\ \exists\mathcal{R}$$

$$\frac{\Gamma \longrightarrow B \quad C, \Gamma \longrightarrow D}{B \supset C, \Gamma \longrightarrow D}\ \supset\mathcal{L} \qquad \frac{B, \Gamma \longrightarrow C}{\Gamma \longrightarrow B \supset C}\ \supset\mathcal{R}$$

$$\frac{}{A, \Gamma \longrightarrow A}\ \mathit{init}, \text{ where } A \text{ is atomic} \qquad \frac{B, B, \Gamma \longrightarrow C}{B, \Gamma \longrightarrow C}\ c\mathcal{L}$$

$$\frac{\Delta_1 \longrightarrow B_1 \quad \cdots \quad \Delta_n \longrightarrow B_n \quad B_1, \ldots, B_n, \Gamma \longrightarrow C}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc, \text{ where } n \geq 0$$

proper induction hypothesis requires insight and creativity; they are not simply rearrangements of the subformulas of the conclusion. As a result, any implementation of $FO\lambda^{\Delta\mathbb{N}}$ will necessarily be interactive, at least for the invention of many induction hypotheses.

A *definitional clause* is written $\forall\bar{x}[p\,\bar{t} \stackrel{\triangle}{=} B]$, where p is a predicate constant, every free variable of the formula B is also free in at least one term in the list $\bar{t}$ of terms, and all variables free in $\bar{t}$ are contained in the list $\bar{x}$ of variables. Since all free variables in $p\,\bar{t}$ and B are universally quantified, we often leave these quantifiers implicit when displaying definitional clauses. The atomic formula $p\,\bar{t}$ is called the *head* of the clause, and the formula B is called the *body*. The symbol $\stackrel{\triangle}{=}$ is used simply to indicate a definitional clause: it is not a logical connective. A *definition* is a (perhaps infinite) set of definitional clauses. The same predicate may occur in the head of multiple clauses of a definition: it is best to think of a definition as a mutually recursive definition of the predicates in the heads of the clauses.

Definitions are employed in $FO\lambda^{\Delta\mathbb{N}}$ via left and right introduction rules for atomic formulas. If we impose no restrictions on definitions, the cut-elimination theorem does not hold [24]. Two different approaches have been taken to retain the admissibility of cut. First, if the structural rule of contraction is removed or restricted (as it is in linear logic, for example), cut-elimination can be established [9, 25]. Another approach, more appropriate for use here since we wish to work within an intuitionistic setting, is to restrict the occurrences of implications within the body of definitions. In [25], Schroeder-Heister proved the cut-elimination theorem for an intuitionistic logic in which no implications are allowed within definitions. Here we shall allow implications in the body of definitions if they are suitably stratified. Toward that end we assume that each predicate symbol p in the language has associated with it a natural number $\mathrm{lvl}(p)$, the *level* of the predicate. The following definition extends the notion of level to formulas and derivations.

Definition 1 Given a formula B, its *level* $\mathrm{lvl}(B)$ is defined as follows:

1. $\mathrm{lvl}(p\,\bar{t}) = \mathrm{lvl}(p)$
2. $\mathrm{lvl}(\bot) = \mathrm{lvl}(\top) = 0$
3. $\mathrm{lvl}(B \wedge C) = \mathrm{lvl}(B \vee C) = \max(\mathrm{lvl}(B), \mathrm{lvl}(C))$
4. $\mathrm{lvl}(B \supset C) = \max(\mathrm{lvl}(B) + 1, \mathrm{lvl}(C))$
5. $\mathrm{lvl}(\forall x.B) = \mathrm{lvl}(\exists x.B) = \mathrm{lvl}(B)$.

Given a derivation Π of $\Gamma \longrightarrow B$, $\mathrm{lvl}(\Pi) = \mathrm{lvl}(B)$.

We shall now require that for every definitional clause $\forall\bar{x}[p\,\bar{t} \stackrel{\triangle}{=} B]$, $\mathrm{lvl}(B) \leq \mathrm{lvl}(p)$.

The logic $FO\lambda^{\Delta\mathbb{N}}$ has uses definitions in left and right-introduction rules for atoms; the following relation will be useful for describing those inference rules.

Definition 2 Let the four-place relation $\mathrm{dfn}(\rho, A, \sigma, B)$ be defined to hold for the formulas A and B and the substitutions ρ and σ if there is a clause $\forall\bar{x}[A' \stackrel{\triangle}{=} B]$ in the given definition such that $A\rho = A'\sigma$.

The right and left rules for atoms are

$$\frac{\Gamma \longrightarrow B\theta}{\Gamma \longrightarrow A}\ \mathit{def}\mathcal{R}, \text{ where } \mathrm{dfn}(\epsilon, A, \theta, B)$$

$$\frac{\{B\sigma, \Gamma\rho \longrightarrow C\rho \mid \mathrm{dfn}(\rho, A, \sigma, B)\}}{A, \Gamma \longrightarrow C}\ \mathit{def}\mathcal{L}\quad ,$$

where ϵ is the empty substitution and the bound variables $\bar{x}$ in the definitional clauses are chosen to be distinct from the variables free in the lower sequent of the rule. Specifying a set of sequents as the premise should be understood to mean that each sequent in the set is a premise of the rule. The right rule corresponds to the logic programming notion of *backchaining* if we think of $\stackrel{\triangle}{=}$ in definitional clauses as reverse implication. The left rule is similar to *definitional reflection* [25] (not to be confused with another notion of reflection often considered between a meta-logic and object-logic) and to an inference rule used by Girard in his note on fixed points [9]. Notice that in the $\mathit{def}\mathcal{L}$ rule, the free variables of the conclusion can be instantiated in the premises.

The number of premises of the $\mathit{def}\mathcal{L}$ rule may be zero or may be infinite. If the formula A does not unify with the head of any definitional clause, then the number of premises will be zero. In this case, A is an unprovable formula logically equivalent to $\bot$, and $\mathit{def}\mathcal{L}$ corresponds to the $\bot\mathcal{L}$ rule. If the formula A does unify with the head of a definitional clause, the number of premises could be infinite, since the domains of the substitutions ρ and σ may include variables that are not free in A and B. In general we wish to work with inference rules with a finite number of premises. This can be achieved by restricting definitions to have only a finite number of clauses and to restrict the use of $\mathit{def}\mathcal{L}$ rule to those formulas A such that for every definitional clause there is a finite, complete set of unifiers (CSU) [11] of A and the head of the clause. Consider the following inference rule due to Eriksson [5]

$$\frac{\{B\theta, \Gamma\theta \longrightarrow C\theta \mid \theta \in CSU(A, A') \text{ for some clause } \forall\bar{x}[A' \stackrel{\triangle}{=} B]\}}{A, \Gamma \longrightarrow C}\ \mathit{def}\mathcal{L}_{CSU}\quad ,$$

where the variables $\bar{x}$ are chosen to be distinct from the variables free in the lower sequent of the rule. When the CSUs and definition are finite, this rule will have a finite number of premises. Notice that in first-order logics, a CSU will have at most one member, namely the most general unifier (MGU).

Proposition 3 *The rules* $\mathit{def}\mathcal{L}$ *and* $\mathit{def}\mathcal{L}_{CSU}$ *are interadmissible, that is, if* $FO\lambda^{\Delta\mathbb{N}}$ *is formulated with either* $\mathit{def}\mathcal{L}$ *or* $\mathit{def}\mathcal{L}_{CSU}$*, the other rule is admissible in that formulation.*

Proof Given the set of derivations

$$\left\{ \begin{array}{c} \Pi^{\theta,B} \\ B\theta, \Gamma\theta \longrightarrow C\theta \end{array} \right\}_{\theta \in CSU(A,A') \text{ for some clause } \forall\bar{x}[A' \stackrel{\triangle}{=} B]} \quad ,$$

we can construct a derivation of $A, \Gamma \longrightarrow C$ using $\mathit{def}\mathcal{L}$ as follows. For any definitional clause $\forall\bar{x}[A' \stackrel{\triangle}{=} B]$ and substitutions ρ and σ such that $A\rho = A'\sigma$, the substitution

$$\rho_\sigma(y) = \begin{cases} \sigma(y) & \text{if } y \in FV(A') \\ \rho(y) & \text{otherwise} \end{cases}$$

will be a unifier of A and A'. Thus for some $\theta \in CSU(A, A')$ there is a substitution θ' such that ρ_σ is $\theta \circ \theta'$. (Notice that composition of substitution is defined so that $A(\theta \circ \theta') = (A\theta)\theta'$.) We can thus use $\Pi^{\theta,B}\theta'$ as the premise derivation of $B\sigma, \Gamma\rho \longrightarrow C\rho$ for $\mathit{def}\mathcal{L}$. (We will formally define what it means to apply a substitution to a derivation in Definition 5. For now it is enough to know that it yields a derivation whose endsequent is obtained by applying the substitution to the endsequent of the original derivation.)

Given the set of derivations

$$\left\{ \begin{array}{c} \Pi^{\rho,\sigma,B} \\ B\sigma, \Gamma\rho \longrightarrow C\rho \end{array} \right\}_{\mathrm{dfn}(\rho,A,\sigma,B)} ,$$

we can construct a derivation of $A, \Gamma \longrightarrow C$ using $def\mathcal{L}_{CSU}$ as follows. For any definitional clause $\forall\bar{x}[A' \triangleq B]$ and substitution $\theta \in CSU(A, A')$, $\mathrm{dfn}(\theta, A, \theta, B)$ holds. We can thus use $\Pi^{\theta,\theta,B}$ as the premise derivation of $B\theta, \Gamma\theta \longrightarrow C\theta$ for $def\mathcal{L}$. ∎

Observe that several of the rules of $FO\lambda^{\Delta\mathbb{N}}$ may have variables that are free in the premise but not in the conclusion: this results from the eigenvariable y of $\forall\mathcal{R}$ and $\exists\mathcal{L}$, the term t of $\forall\mathcal{L}$ and $\exists\mathcal{R}$, the cut formulas $B_1, \ldots, B_n$ of mc, the induction predicate B of $nat\mathcal{L}$, and the substitutions ρ and σ of $def\mathcal{L}$. We view the choice of such variables as arbitrary and identify all derivations that differ only in the choice of variables that are not free in end-sequent.

We define an ordinal measure which corresponds to the height of a derivation:

Definition 4 Given a derivation Π with premise derivations $\{\Pi_i\}_i$, the measure $\mathrm{ht}(\Pi)$ is the least upper bound of $\{\mathrm{ht}(\Pi_i) + 1\}_i$.

Substitutions are finite maps from variables to terms. It is common to view substitutions as maps from terms to terms by applying the substitution to all free variables of a term. We can then extend the mapping in turn to formulas and multisets by applying it to every term in a formula and every formula in a multiset. The following definition extends substitutions yet again to apply to derivations. Since we identify derivations that differ only in the choice of variables that are not free in the end-sequent, we will assume that such variables are chosen to be distinct from the variables in the domain of the substitution and from the free variables of the range of the substitution. Thus applying a substitution to a derivation will only affect the variables free in the end-sequent.

Definition 5 If Π is a derivation of $\Gamma \longrightarrow C$ and θ is a substitution, then we define the derivation $\Pi\theta$ of $\Gamma\theta \longrightarrow C\theta$ as follows:

1. Suppose Π ends with the $def\mathcal{L}$ rule

$$\frac{\left\{ \begin{array}{c} \Pi^{\rho,\sigma,B} \\ B\sigma, \Gamma'\rho \longrightarrow C\rho \end{array} \right\}_{\mathrm{dfn}(\rho,A,\sigma,B)}}{A, \Gamma' \longrightarrow C}\ def\mathcal{L} .$$

   Observe that if $\mathrm{dfn}(\rho', A\theta, \sigma', B)$ then $\mathrm{dfn}(\theta \circ \rho', A, \sigma', B)$. Thus $\Pi\theta$ is

$$\frac{\left\{ \begin{array}{c} \Pi^{\theta\circ\rho',\sigma',B} \\ B\sigma', \Gamma'\theta\rho' \longrightarrow C\theta\rho' \end{array} \right\}_{\mathrm{dfn}(\rho',A\theta,\sigma',B)}}{A\theta, \Gamma'\theta \longrightarrow C\theta}\ def\mathcal{L} .$$

2. If Π ends with any other rule and has premise derivations $\Pi_1, \ldots, \Pi_n$, then $\Pi\theta$ also ends with the same rule and has premise derivations $\Pi_1\theta, \ldots, \Pi_n\theta$.

Lemma 6 *For any substitution θ and derivation Π of $\Gamma \longrightarrow C$, $\Pi\theta$ is a derivation of $\Gamma\theta \longrightarrow C\theta$.*

Proof This lemma states that Definition 5 is well-constructed, and follows by induction on $\mu(\Pi)$. Observe that if Π ends with the $def\mathcal{R}$ rule

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma \longrightarrow B\sigma\end{array}}{\Gamma \longrightarrow A}\ def\mathcal{R} \quad ,$$

then $\mathrm{dfn}(\epsilon, A, \sigma, B)$, and so it is also true that $\mathrm{dfn}(\epsilon, A\theta, \sigma \circ \theta, B)$. Therefore

$$\dfrac{\begin{array}{c}\Pi'\theta \\ \Gamma\theta \longrightarrow B\sigma\theta\end{array}}{\Gamma\theta \longrightarrow A\theta}\ def\mathcal{R}$$

is a valid derivation. ∎

Lemma 7 *For any derivation* Π *and substitution* θ*,* $\mathrm{ht}(\Pi) \geq \mathrm{ht}(\Pi\theta)$*.*

Proof The proof of this lemma is a simple induction on $\mathrm{ht}(\Pi)$. The measures may not be equal because when the derivations end with the $def\mathcal{L}$ rule, some of the premise derivations of Π may not be needed to construct the premise derivations of $\Pi\theta$. ∎

Our logic does not contain a weakening rule; instead we allow extra assumptions in the axioms. The following definition provides meta-level weakening on derivations. Since we identify derivations that differ only in the choice of variables that are not free in the end-sequent, we will assume that such variables are chosen to be distinct from the free variables of the weakening formulas.

Definition 8 If Π is a derivation of $\Gamma \longrightarrow C$ and Δ is a multiset of formulas, then we define the derivation $w(\Delta, \Pi)$ of $\Gamma, \Delta \longrightarrow C$ as follows:

1. If Π ends with the $def\mathcal{L}$ rule

$$\dfrac{\left\{\begin{array}{c}\Pi^{\rho,\sigma,B} \\ B\sigma, \Gamma'\rho \longrightarrow C\rho\end{array}\right\}}{A, \Gamma' \longrightarrow C}\ def\mathcal{L} \quad ,$$

then $w(\Delta, \Pi)$ is

$$\dfrac{\left\{\begin{array}{c}w(\Delta\rho, \Pi^{\rho,\sigma,B}) \\ B\sigma, \Gamma'\rho, \Delta\rho \longrightarrow C\rho\end{array}\right\}}{A, \Gamma', \Delta \longrightarrow C}\ def\mathcal{L} \quad .$$

2. If Π ends with the $nat\mathcal{L}$ rule

$$\dfrac{\begin{array}{c}\Pi_1 \\ \longrightarrow B\,z\end{array} \quad \begin{array}{c}\Pi_2 \\ B\,j \longrightarrow B\,(s\,j)\end{array} \quad \begin{array}{c}\Pi_3 \\ B\,I, \Gamma \longrightarrow C\end{array}}{nat\ I, \Gamma \longrightarrow C}\ nat\mathcal{L} \quad ,$$

then $w(\Delta, \Pi)$ is

$$\dfrac{\begin{array}{c}\Pi_1 \\ \longrightarrow B\,z\end{array} \quad \begin{array}{c}\Pi_2 \\ B\,j \longrightarrow B\,(s\,j)\end{array} \quad \begin{array}{c}w(\Delta, \Pi_3) \\ B\,I, \Gamma, \Delta \longrightarrow C\end{array}}{nat\ I, \Gamma, \Delta \longrightarrow C}\ nat\mathcal{L} \quad .$$

3. If Π ends with the mc rule

$$\frac{\begin{array}{c}\Pi_1\\ \Delta_1 \longrightarrow B_1\end{array} \quad \cdots \quad \begin{array}{c}\Pi_n\\ \Delta_n \longrightarrow B_n\end{array} \quad \begin{array}{c}\Pi'\\ B_1, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc \quad ,$$

then $w(\Delta, \Pi)$ is

$$\frac{\begin{array}{c}\Pi_1\\ \Delta_1 \longrightarrow B_1\end{array} \quad \cdots \quad \begin{array}{c}\Pi_n\\ \Delta_n \longrightarrow B_n\end{array} \quad \begin{array}{c}w(\Delta, \Pi')\\ B_1, \ldots, B_n, \Gamma, \Delta \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma, \Delta \longrightarrow C}\ mc \quad .$$

4. If Π ends with any other rule and has premise derivations $\Pi_1, \ldots, \Pi_n$, then $w(\Delta, \Pi)$ also ends with the same rule and has premise derivations $w(\Delta, \Pi_1), \ldots, w(\Delta, \Pi_n)$.

The following lemmas can be proved by induction on the measure of the given derivation.

Lemma 9 *For any multiset Δ of formulas and derivation Π of $\Gamma \longrightarrow C$, $w(\Delta, \Pi)$ is a derivation of $\Gamma, \Delta \longrightarrow C$.*

Lemma 10 *For any derivation Π and multiset Δ of formulas,* $\mathrm{ht}(\Pi) = \mathrm{ht}(w(\Delta, \Pi))$.

Lemma 11 *For any derivation Π, multiset Δ of formulas, and substitution θ,*

$$w(\Delta, \Pi)\theta \quad \textit{and} \quad w(\Delta\theta, \Pi\theta)$$

are the same derivation.

Lemma 12 *For any derivation Π and multisets Δ and Δ' of formulas,*

$$w(\Delta, w(\Delta', \Pi)) \quad \textit{and} \quad w(\Delta \cup \Delta', \Pi)$$

are the same derivation.

4 Overview of the Cut-Elimination Proof

Gentzen's classic proof of cut-elimination for first-order logic [7] uses an induction involving the number of logical connectives in the cut formula. A cut on a compound formula is replaced by cuts on its subformulas, which necessarily contain a lower number of connectives. For example, the derivation

$$\frac{\dfrac{\begin{array}{c}\Pi_1\\ \Delta \longrightarrow B_1\end{array} \quad \begin{array}{c}\Pi_2\\ \Delta \longrightarrow B_2\end{array}}{\Delta \longrightarrow B_1 \wedge B_2}\ \wedge\mathcal{R} \qquad \dfrac{\begin{array}{c}\Pi_3\\ B_1, \Gamma \longrightarrow C\end{array}}{B_1 \wedge B_2, \Gamma \longrightarrow C}\ \wedge\mathcal{L}}{\Delta, \Gamma \longrightarrow C}\ mc$$

is reduced to

$$\frac{\begin{array}{c}\Pi_1\\ \Delta \longrightarrow B_1\end{array} \quad \begin{array}{c}\Pi_3\\ B_1, \Gamma \longrightarrow C\end{array}}{\Delta, \Gamma \longrightarrow C}\ mc \quad .$$

By the induction hypothesis, this cut on B_1 is eliminable, hence the original cut on $B_1 \wedge B_2$ is also eliminable. In first-order logic, when the cut formula is atomic, the cut can easily be removed by permuting the cut up toward the leaves of the proof; eventually an initial rule is reached, at which point the removal of the cut is trivial.

In $FO\lambda^{\Delta\mathbb{N}}$, however, the rules for natural numbers and definitions act on atoms, so the atomic case is not simple. Consider, for example, the derivation

$$\dfrac{\dfrac{\dfrac{\Pi_1}{\Delta \longrightarrow B\theta}}{\Delta \longrightarrow A}\, def\mathcal{R} \qquad \dfrac{\left\{\dfrac{\Pi^{\rho,\sigma,D}}{D\sigma, \Gamma\rho \longrightarrow C\rho}\right\}}{A, \Gamma \longrightarrow C}\, def\mathcal{L}}{\Delta, \Gamma \longrightarrow C}\, mc \quad .$$

The obvious reduction for this is a cut between Π_1 and the appropriate premise of the $def\mathcal{L}$ rule; however, $B\theta$ is a formula of arbitrary complexity, and so will in general have a greater number of connectives than the atom A, which has zero. Thus a different induction measure is needed.

Schroeder-Heister proves cut-elimination for several logics with definitions [24, 25] by including the number of uses of the $def\mathcal{L}$ rule in the derivation as part of the induction measure. However, the logics he considers do not contain induction; the inclusion of the $nat\mathcal{R}$ and $nat\mathcal{L}$ rules in $FO\lambda^{\Delta\mathbb{N}}$ complicates things further. The derivation

$$\dfrac{\dfrac{\dfrac{\Pi_1}{\Delta \longrightarrow nat\ I}}{\Delta \longrightarrow nat\ (s\ I)}\, nat\mathcal{R} \qquad \dfrac{\dfrac{\Pi_2}{\longrightarrow B\, z} \quad \dfrac{\Pi_3}{B\, j \longrightarrow B\, (s\, j)} \quad \dfrac{\Pi_4}{B\, (s\ I), \Gamma \longrightarrow C}}{nat\ (s\ I), \Gamma \longrightarrow C}\, nat\mathcal{L}}{\Delta, \Gamma \longrightarrow C}\, mc$$

can be reduced in a number of ways, but the reductions are all variations of the derivation

$$\dfrac{\dfrac{\dfrac{\Pi_1}{\Delta \longrightarrow nat\ I} \qquad \dfrac{\dfrac{\Pi_2}{\longrightarrow B\, z} \quad \dfrac{\Pi_3}{B\, j \longrightarrow B\, (s\, j)} \quad \dfrac{\Pi_3[I/j]}{B\, I \longrightarrow B\, (s\ I)}}{nat\ I \longrightarrow B\, (s\ I)}\, nat\mathcal{L}}{\Delta \longrightarrow B\, (s\ I)}\, mc \qquad \dfrac{\Pi_4}{B\, (s\ I), \Gamma \longrightarrow C}}{\Delta, \Gamma \longrightarrow C}\, mc \quad .$$

Here, the cut on the atomic formula $nat\ (s\ I)$ is replaced by two cuts, one on the atom $nat\ I$ and the other on the formula $B\, (s\ I)$. It is not clear what induction measure can be used here. For the first cut, the atom $nat\ I$ contains no logical connectives, but this is true of the original cut formula $nat\ (s\ I)$ as well. The number of $nat\mathcal{R}$ rules in the right subderivation of the cut has gone down by one, but the duplication of Π_3 might offset this. For the second cut, the cut formula $B\, (s\ I)$ is not related at all to the original cut formula; it certainly can have no fewer connectives than the atom $nat\ (s\ I)$. And though its left premise is shorter than the left premise of the original cut, it is unclear how the heights of the right premises compare.

It should be noted, however, that the complicating factor here is not the presence of an induction rule, but our use of the nat predicate in the induction rule. If we remove the $nat\mathcal{R}$ rules from the logic and reformulate the induction rule to be

$$\dfrac{\longrightarrow B\, z \quad B\, j \longrightarrow B\, (s\, j) \quad B\, I, \Gamma \longrightarrow C}{\Gamma \longrightarrow C}\, ind \quad ,$$

then Schroeder-Heister's proofs can be extended to that logic. Despite this, we prefer to include the *nat* predicate in our formulation of the logic. At an aesthetic level, our formulation maintains the symmetry between right and left rules of the logic. Including the *nat* predicate also keeps the form of the induction rule for natural numbers consistent with the form of the induction rules we can derive from it for defined predicates [13]. Finally, the *nat* predicate plays a key role in the adequacy proofs for encodings of intuitionistic and linear logic frameworks in $FO\lambda^{\Delta\mathbb{N}}$ [14].

Our proof of the cut-elimination theorem for $FO\lambda^{\Delta\mathbb{N}}$ uses a technique introduced by Tait [29] to prove normal form theorems. Martin-Löf extended the method to apply beyond terms to natural deduction proofs [12], and we use it here in a sequent calculus setting. Rather than associate an induction measure with derivations, we use the derivations themselves as a measure by defining well-founded orderings on derivations, and performing the induction relative to those orderings. The basis for the orderings is a set of reduction rules, such as those suggested above, that will be used to eliminate applications of the cut rule. We give these reduction rules in Section 5. This is followed by a section which discusses two orderings on derivations, a *normalizability* ordering and a *reducibility* ordering. The well-foundedness of the normalizability ordering for a derivation implies that the reduction rules can be used to reduce the derivation to a cut-free derivation of the same end-sequent. The reducibility ordering is a superset of the normalizability ordering; thus its well-foundedness implies the well-foundedness of the normalizability ordering. (This notion of reducibility was called convertibility by Tait and computability by Martin-Löf. We prefer to avoid these terms since they carry other meanings in theoretical computer science and, instead, use reducibility after Girard [8].) In Section 7 we prove the key lemma: for every derivation, the tree of its successive predecessors in the reducibility relation is well-founded. From this we conclude that the corresponding tree in the normalizability relation is also well-founded, and hence the cut rule can be eliminated from that derivation. Since this holds for every derivation, the consistency of $FO\lambda^{\Delta\mathbb{N}}$ follows.

5 Reduction Rules for Derivations

Here we define a reduction relation between derivations, which is an adaptation of the reduction rules used in Gentzen's original Hauptsatz [7].

Definition 13 We define a *reduction* relation between derivations. The redex is always a derivation Ξ ending with the multicut rule

$$\frac{\begin{array}{c}\Pi_1\\ \Delta_1 \longrightarrow B_1\end{array} \quad \cdots \quad \begin{array}{c}\Pi_n\\ \Delta_n \longrightarrow B_n\end{array} \quad \begin{array}{c}\Pi\\ B_1,\dots,B_n,\Gamma \longrightarrow C\end{array}}{\Delta_1,\dots,\Delta_n,\Gamma \longrightarrow C}\ mc \quad .$$

If $n = 0$, Ξ reduces to the premise derivation Π.

For $n > 0$ we specify the reduction relation based on the last rule of the premise derivations. If the rightmost premise derivation Π ends with a left rule acting on a cut formula B_i, then the last rule of Π_i and the last rule of Π together determine the reduction rules that apply. We classify these rules according to the following criteria: we call the rule an *essential* case when Π_i ends with a right rule; if it ends with a left rule, it is a *right-commutative* case; if Π_i ends with the *init* rule, then we have an *axiom* case; a *multicut* case arises when it ends with the *mc* rule. When Π does not end with a left rule acting on a cut formula, then its last rule is alone sufficient to determine the reduction rules that apply. If Π ends in a rule acting on a formula other than a cut formula, then we call this a *left-commutative* case. A *structural* case results when Π ends with a contraction on a cut formula. If Π ends with the *init* rule, this is also an axiom case; similarly a multicut case arises if Π ends in the *mc* rule. For simplicity of presentation, we always show $i = 1$.

Essential cases:

$\wedge\mathcal{R}/\wedge\mathcal{L}$: If Π_1 and Π are

$$\dfrac{\dfrac{\Pi_1'}{\Delta_1 \longrightarrow B_1'} \quad \dfrac{\Pi_1''}{\Delta_1 \longrightarrow B_1''}}{\Delta_1 \longrightarrow B_1' \wedge B_1''}\ \wedge\mathcal{R} \qquad\qquad \dfrac{\dfrac{\Pi'}{B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C}}{B_1' \wedge B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \wedge\mathcal{L}\ ,$$

then Ξ reduces to

$$\dfrac{\dfrac{\Pi_1'}{\Delta_1 \longrightarrow B_1'} \quad \dfrac{\Pi_2}{\Delta_2 \longrightarrow B_2} \quad \cdots \quad \dfrac{\Pi_n}{\Delta_n \longrightarrow B_n} \quad \dfrac{\Pi'}{B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc\ .$$

The case for the other $\wedge\mathcal{L}$ rule is symmetric.

$\vee\mathcal{R}/\vee\mathcal{L}$: If Π_1 and Π are

$$\dfrac{\dfrac{\Pi_1'}{\Delta_1 \longrightarrow B_1'}}{\Delta_1 \longrightarrow B_1' \vee B_1''}\ \vee\mathcal{R} \qquad\qquad \dfrac{\dfrac{\Pi'}{B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C} \quad \dfrac{\Pi''}{B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}}{B_1' \vee B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \vee\mathcal{L}\ ,$$

then Ξ reduces to

$$\dfrac{\dfrac{\Pi_1'}{\Delta_1 \longrightarrow B_1'} \quad \dfrac{\Pi_2}{\Delta_2 \longrightarrow B_2} \quad \cdots \quad \dfrac{\Pi_n}{\Delta_n \longrightarrow B_n} \quad \dfrac{\Pi'}{B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc\ .$$

The case for the other $\vee\mathcal{R}$ rule is symmetric.

$\supset\mathcal{R}/\supset\mathcal{L}$: Suppose Π_1 and Π are

$$\dfrac{\dfrac{\Pi_1'}{B_1', \Delta_1 \longrightarrow B_1''}}{\Delta_1 \longrightarrow B_1' \supset B_1''}\ \supset\mathcal{R} \qquad\qquad \dfrac{\dfrac{\Pi'}{B_2, \ldots, B_n, \Gamma \longrightarrow B_1'} \quad \dfrac{\Pi''}{B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}}{B_1' \supset B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \supset\mathcal{L}\ .$$

Let Ξ_1 be

$$\dfrac{\dfrac{\left\{ \dfrac{\Pi_i}{\Delta_i \longrightarrow B_i} \right\}_{i \in \{2..n\}} \quad \dfrac{\Pi'}{B_2, \ldots, B_n, \Gamma \longrightarrow B_1'}}{\Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow B_1'}\ mc \quad \dfrac{\Pi_1'}{B_1', \Delta_1 \longrightarrow B_1''}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow B_1''}\ mc\ .$$

Then Ξ reduces to

$$\dfrac{\dfrac{\dfrac{\Xi_1}{\ldots \longrightarrow B_1''} \quad \left\{ \dfrac{\Pi_i}{\Delta_i \longrightarrow B_i} \right\}_{i \in \{2..n\}} \quad \dfrac{\Pi''}{B_1'', \{B_i\}_{i\in\{2..n\}}, \Gamma \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\, mc}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\, c\mathcal{L} \quad .$$

We use the double horizontal lines to indicate that the relevant inference rule (in this case, $c\mathcal{L}$) may need to be applied zero or more times.

$\forall\mathcal{R}/\forall\mathcal{L}$: If Π_1 and Π are

$$\dfrac{\dfrac{\Pi_1'}{\Delta_1 \longrightarrow B_1'[y/x]}}{\Delta_1 \longrightarrow \forall x.B_1'}\, \forall\mathcal{R} \qquad\qquad \dfrac{\dfrac{\Pi'}{B_1'[t/x], B_2, \ldots, B_n, \Gamma \longrightarrow C}}{\forall x.B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C}\, \forall\mathcal{L} \quad ,$$

then Ξ reduces to

$$\dfrac{\dfrac{\Pi_1'[t/y]}{\Delta_1 \longrightarrow B_1'[t/x]} \quad \left\{ \dfrac{\Pi_i}{\Delta_i \longrightarrow B_i} \right\}_{i \in \{2..n\}} \quad \dfrac{\Pi'}{\ldots \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\, mc \quad .$$

$\exists\mathcal{R}/\exists\mathcal{L}$: If Π_1 and Π are

$$\dfrac{\dfrac{\Pi_1'}{\Delta_1 \longrightarrow B_1'[t/x]}}{\Delta_1 \longrightarrow \exists x.B_1'}\, \exists\mathcal{R} \qquad\qquad \dfrac{\dfrac{\Pi'}{B_1'[y/x], B_2, \ldots, B_n, \Gamma \longrightarrow C}}{\exists x.B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C}\, \exists\mathcal{L} \quad ,$$

then Ξ reduces to

$$\dfrac{\dfrac{\Pi_1'}{\Delta_1 \longrightarrow B_1'[t/x]} \quad \left\{ \dfrac{\Pi_i}{\Delta_i \longrightarrow B_i} \right\}_{i \in \{2..n\}} \quad \dfrac{\Pi'[t/y]}{\ldots \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\, mc \quad .$$

$nat\mathcal{R}/nat\mathcal{L}$: Suppose Π_1 is $\dfrac{}{\Delta_1 \longrightarrow nat\ z}\, nat\mathcal{R}$ and Π is

$$\dfrac{\dfrac{\Pi'}{\longrightarrow D\, z} \quad \dfrac{\Pi''}{D\, j \longrightarrow D\,(s\ j)} \quad \dfrac{\Pi'''}{D\, z, B_2, \ldots, B_n, \Gamma \longrightarrow C}}{nat\ z, B_2, \ldots, B_n, \Gamma \longrightarrow C}\, nat\mathcal{L} \quad .$$

Then Ξ reduces to

$$\dfrac{\dfrac{w(\Delta_1, \Pi')}{\Delta_1 \longrightarrow D\, z} \quad \left\{ \dfrac{\Pi_i}{\Delta_i \longrightarrow B_i} \right\}_{i \in \{2..n\}} \quad \dfrac{\Pi'''}{D\, z, B_2, \ldots, B_n, \Gamma \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\, mc \quad .$$

$nat\mathcal{R}/nat\mathcal{L}$**:** Suppose Π_1 is

$$\dfrac{\begin{array}{c}\Pi_1' \\ \Delta_1 \longrightarrow nat\ I\end{array}}{\Delta_1 \longrightarrow nat\ (s\ I)}\ nat\mathcal{R}$$

and Π is

$$\dfrac{\begin{array}{c}\Pi' \\ \longrightarrow D\,z\end{array} \quad \begin{array}{c}\Pi'' \\ D\,j \longrightarrow D\,(s\ j)\end{array} \quad \begin{array}{c}\Pi''' \\ D\,(s\ I), B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{nat\ (s\ I), B_2, \ldots, B_n, \Gamma \longrightarrow C}\ nat\mathcal{L} \quad .$$

Let Ξ_1 be

$$\dfrac{\begin{array}{c}\Pi_1' \\ \Delta_1 \longrightarrow nat\ I\end{array} \quad \dfrac{\begin{array}{c}\Pi' \\ \longrightarrow D\,z\end{array} \quad \begin{array}{c}\Pi'' \\ D\,j \longrightarrow D\,(s\ j)\end{array} \quad \dfrac{}{D\,I \longrightarrow D\,I}\ init}{nat\ I \longrightarrow D\,I}\ nat\mathcal{L}}{\Delta_1 \longrightarrow D\,I}\ mc \quad ,$$

and Ξ_2 be

$$\dfrac{\begin{array}{c}\Xi_1 \\ \Delta_1 \longrightarrow D\,I\end{array} \quad \begin{array}{c}\Pi''[I/j] \\ D\,I \longrightarrow D\,(s\ I)\end{array}}{\Delta_1 \longrightarrow D\,(s\ I)}\ mc \quad .$$

Then Ξ reduces to

$$\dfrac{\begin{array}{c}\Xi_2 \\ \Delta_1 \longrightarrow D\,(s\ I)\end{array} \quad \left\{\begin{array}{c}\Pi_i \\ \Delta_i \longrightarrow B_i\end{array}\right\}_{i\in\{2..n\}} \quad \begin{array}{c}\Pi''' \\ D\,(s\ I), B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc \quad .$$

$def\mathcal{R}/def\mathcal{L}$**:** Suppose Π_1 and Π are

$$\dfrac{\begin{array}{c}\Pi_1' \\ \Delta_1 \longrightarrow B_1'\theta\end{array}}{\Delta_1 \longrightarrow B_1}\ def\mathcal{R} \qquad\qquad \dfrac{\left\{\begin{array}{c}\Pi^{\rho,\sigma,D} \\ D\sigma, B_2\rho, \ldots, B_n\rho, \Gamma\rho \longrightarrow C\rho\end{array}\right\}}{B_1, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ def\mathcal{L} \quad .$$

Then by the $def\mathcal{R}$ rule in Π_1 $\mathrm{dfn}(\epsilon, B_1, \theta, B_1')$ holds. Let θ' be the restriction of θ to the variables $\bar{x}$ of the relevant definitional clause. Since B_1' is the body of this clause, its free variables are included in $\bar{x}$, and so $B_1'\theta' = B_1'\theta$. Then Ξ reduces to

$$\dfrac{\begin{array}{c}\Pi_1' \\ \Delta_1 \longrightarrow B_1'\theta\end{array} \quad \left\{\begin{array}{c}\Pi_i \\ \Delta_i \longrightarrow B_i\end{array}\right\}_{i\in\{2..n\}} \quad \begin{array}{c}\Pi^{\epsilon,\theta',B_1'} \\ B_1'\theta, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc \quad .$$

<u>*Left-commutative cases:*</u>

$\bullet\mathcal{L}/\circ\mathcal{L}$**:** Suppose Π ends with a left rule other than $c\mathcal{L}$ acting on B_1, and Π_1 is

$$\dfrac{\left\{\begin{array}{c}\Pi_1^i \\ \Delta_1^i \longrightarrow B_1\end{array}\right\}}{\Delta_1 \longrightarrow B_1}\ \bullet\mathcal{L} \quad ,$$

where $\bullet\mathcal{L}$ is any left rule except $\supset\mathcal{L}$, $def\mathcal{L}$, or $nat\mathcal{L}$ (but including $c\mathcal{L}$). Then Ξ reduces to

$$\cfrac{\left\{\cfrac{\begin{array}{c}\Pi_1^i\\ \Delta_1^i \longrightarrow B_1\end{array} \quad \left\{\begin{array}{c}\Pi_j\\ \Delta_j \longrightarrow B_j\end{array}\right\}_{j\in\{2..n\}} \quad \begin{array}{c}\Pi\\ B_1,\ldots,B_n,\Gamma \longrightarrow C\end{array}}{\Delta_1^i,\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow C}\ mc\right\}}{\Delta_1,\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow C}\ \bullet\mathcal{L} \quad .$$

$\supset\mathcal{L}/\circ\mathcal{L}$**:** Suppose Π ends with a left rule other than $c\mathcal{L}$ acting on B_1 and Π_1 is

$$\cfrac{\begin{array}{c}\Pi_1'\\ \Delta_1' \longrightarrow D_1'\end{array} \quad \begin{array}{c}\Pi_1''\\ D_1'',\Delta_1' \longrightarrow B_1\end{array}}{D_1' \supset D_1'',\Delta_1' \longrightarrow B_1}\ \supset\mathcal{L} \quad .$$

Let Ξ_1 be

$$\cfrac{\begin{array}{c}\Pi_1''\\ D_1'',\Delta_1' \longrightarrow B_1\end{array} \quad \begin{array}{c}\Pi_2\\ \Delta_2 \longrightarrow B_2\end{array} \quad \cdots \quad \begin{array}{c}\Pi_n\\ \Delta_n \longrightarrow B_n\end{array} \quad \begin{array}{c}\Pi\\ B_1,\ldots,B_n,\Gamma \longrightarrow C\end{array}}{D_1'',\Delta_1',\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow C}\ mc \quad .$$

Then Ξ reduces to

$$\cfrac{\begin{array}{c}w(\Delta_2 \cup \ldots \cup \Delta_n \cup \Gamma, \Pi_1')\\ \Delta_1',\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow D_1'\end{array} \quad \begin{array}{c}\Xi_1\\ D_1'',\Delta_1',\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow C\end{array}}{D_1' \supset D_1'',\Delta_1',\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow C}\ \supset\mathcal{L} \quad .$$

$nat\mathcal{L}/\circ\mathcal{L}$**:** Suppose Π ends with a left rule other than $c\mathcal{L}$ acting on B_1, and Π_1 is

$$\cfrac{\begin{array}{c}\Pi_1^1\\ \longrightarrow D_1\,z\end{array} \quad \begin{array}{c}\Pi_1^2\\ D_1\,j \longrightarrow D_1\,(s\ j)\end{array} \quad \begin{array}{c}\Pi_1^3\\ D_1\,I,\Delta_1' \longrightarrow B_1\end{array}}{nat\ I,\Delta_1' \longrightarrow B_1}\ nat\mathcal{L} \quad .$$

Let Ξ_1 be

$$\cfrac{\begin{array}{c}\Pi_1^3\\ D_1\,I,\Delta_1' \longrightarrow B_1\end{array} \quad \left\{\begin{array}{c}\Pi_i\\ \Delta_i \longrightarrow B_i\end{array}\right\}_{i\in\{2..n\}} \quad \begin{array}{c}\Pi\\ B_1,\ldots,B_n,\Gamma \longrightarrow C\end{array}}{D_1\,I,\Delta_1',\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow C}\ mc \quad .$$

Then Ξ reduces to

$$\cfrac{\begin{array}{c}\Pi_1^1\\ \longrightarrow D_1\,z\end{array} \quad \begin{array}{c}\Pi_1^2\\ D_1\,j \longrightarrow D_1\,(s\ j)\end{array} \quad \begin{array}{c}\Xi_1\\ D_1\,I,\Delta_1',\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow C\end{array}}{nat\ I,\Delta_1',\Delta_2,\ldots,\Delta_n,\Gamma \longrightarrow C}\ nat\mathcal{L} \quad .$$

$def\mathcal{L}/\circ\mathcal{L}$**:** If Π ends with a left rule other than $c\mathcal{L}$ acting on B_1 and Π_1 is

$$\cfrac{\left\{\begin{array}{c}\Pi_1^{\rho,\sigma,D}\\ D\sigma,\Delta_1'\rho \longrightarrow B_1\rho\end{array}\right\}}{A,\Delta_1' \longrightarrow B_1}\ def\mathcal{L} \quad ,$$

then Ξ reduces to

$$\dfrac{\left\{\dfrac{\begin{array}{c}\Pi_1^{\rho,\sigma,D}\\ D\sigma, \Delta_1'\rho \longrightarrow B_1\rho\end{array} \quad \left\{\begin{array}{c}\Pi_i\rho\\ \Delta_i\rho \longrightarrow B_i\rho\end{array}\right\}_{i\in\{2..n\}} \quad \begin{array}{c}\Pi\rho\\ \ldots \longrightarrow C\rho\end{array}}{D\sigma, \Delta_1'\rho, \Delta_2\rho, \ldots, \Delta_n\rho, \Gamma\rho \longrightarrow C\rho}\, mc\right\}}{A, \Delta_1', \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\, def\mathcal{L} \quad .$$

<u>*Right-commutative cases:*</u>

$-/\circ\mathcal{L}$**:** Suppose Π is

$$\dfrac{\left\{\begin{array}{c}\Pi^i\\ B_1, \ldots, B_n, \Gamma^i \longrightarrow C\end{array}\right\}}{B_1, \ldots, B_n, \Gamma \longrightarrow C}\, \circ\mathcal{L} \quad ,$$

where $\circ\mathcal{L}$ is any left rule other than $\supset\mathcal{L}$, $def\mathcal{L}$, or $nat\mathcal{L}$ (but including $c\mathcal{L}$) acting on a formula other than $B_1, \ldots, B_n$. Then Ξ reduces to

$$\dfrac{\left\{\dfrac{\begin{array}{c}\Pi_1\\ \Delta_1 \longrightarrow B_1\end{array} \quad \cdots \quad \begin{array}{c}\Pi_n\\ \Delta_n \longrightarrow B_n\end{array} \quad \begin{array}{c}\Pi^i\\ B_1, \ldots, B_n, \Gamma^i \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma^i \longrightarrow C}\, mc\right\}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\, \circ\mathcal{L} \quad .$$

$-/\supset\mathcal{L}$**:** Suppose Π is

$$\dfrac{\begin{array}{c}\Pi'\\ B_1, \ldots, B_n, \Gamma' \longrightarrow D'\end{array} \quad \begin{array}{c}\Pi''\\ B_1, \ldots, B_n, D'', \Gamma' \longrightarrow C\end{array}}{B_1, \ldots, B_n, D' \supset D'', \Gamma' \longrightarrow C}\, \supset\mathcal{L} \quad .$$

Let Ξ_1 be

$$\dfrac{\begin{array}{c}\Pi_1\\ \Delta_1 \longrightarrow B_1\end{array} \quad \cdots \quad \begin{array}{c}\Pi_n\\ \Delta_n \longrightarrow B_n\end{array} \quad \begin{array}{c}\Pi'\\ B_1, \ldots, B_n, \Gamma' \longrightarrow D'\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma' \longrightarrow D'}\, mc$$

and Ξ_2 be

$$\dfrac{\begin{array}{c}\Pi_1\\ \Delta_1 \longrightarrow B_1\end{array} \quad \cdots \quad \begin{array}{c}\Pi_n\\ \Delta_n \longrightarrow B_n\end{array} \quad \begin{array}{c}\Pi''\\ B_1, \ldots, B_n, D'', \Gamma' \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, D'', \Gamma' \longrightarrow C}\, mc \quad .$$

Then Ξ reduces to

$$\dfrac{\begin{array}{c}\Xi_1\\ \Delta_1, \ldots, \Delta_n, \Gamma' \longrightarrow D'\end{array} \quad \begin{array}{c}\Xi_2\\ \Delta_1, \ldots, \Delta_n, D'', \Gamma' \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, D' \supset D'', \Gamma' \longrightarrow C}\, \supset\mathcal{L} \quad .$$

$-/nat\mathcal{L}$**:** Suppose Π is

$$\dfrac{\begin{array}{c}\Pi'\\ \longrightarrow D\,z\end{array} \quad \begin{array}{c}\Pi''\\ D\,j \longrightarrow D\,(s\;j)\end{array} \quad \begin{array}{c}\Pi'''\\ B_1, \ldots, B_n, D\,I, \Gamma' \longrightarrow C\end{array}}{B_1, \ldots, B_n, \mathrm{nat}\;I, \Gamma' \longrightarrow C}\, nat\mathcal{L} \quad .$$

Let Ξ_1 be

$$\dfrac{\overset{\Pi_1}{\Delta_1 \longrightarrow B_1} \quad \cdots \quad \overset{\Pi_n}{\Delta_n \longrightarrow B_n} \quad \overset{\Pi'''}{B_1, \ldots, B_n, D\,I, \Gamma' \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, D\,I, \Gamma' \longrightarrow C}\ mc \quad .$$

then Ξ reduces to

$$\dfrac{\overset{\Pi'}{\longrightarrow D\,z} \quad \overset{\Pi''}{D\,j \longrightarrow D\,(s\ j)} \quad \overset{\Xi_1}{\Delta_1, \ldots, \Delta_n, D\,I, \Gamma' \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, nat\ I, \Gamma' \longrightarrow C}\ nat\mathcal{L} \quad .$$

$-/def\mathcal{L}$**:** If Π is

$$\dfrac{\left\{ \overset{\Pi^{\rho,\sigma,D}}{B_1\rho, \ldots, B_n\rho, D\sigma, \Gamma'\rho \longrightarrow C\rho} \right\}}{B_1, \ldots, B_n, A, \Gamma' \longrightarrow C}\ def\mathcal{L} \quad ,$$

then Ξ reduces to

$$\dfrac{\left\{ \dfrac{\left\{ \overset{\Pi_i\rho}{\Delta_i\rho \longrightarrow B_i\rho} \right\}_{i\in\{1..n\}} \quad \overset{\Pi^{\rho,\sigma,D}}{\{B_i\rho\}_{i\in\{1..n\}}, D\sigma, \Gamma'\rho \longrightarrow C\rho}}{\Delta_1\rho, \ldots, \Delta_n\rho, D\sigma, \Gamma'\rho \longrightarrow C\rho}\ mc \right\}}{\Delta_1, \ldots, \Delta_n, A, \Gamma' \longrightarrow C}\ def\mathcal{L} \quad .$$

$-/\circ\mathcal{R}$**:** If Π is

$$\dfrac{\left\{ \overset{\Pi^i}{B_1, \ldots, B_n, \Gamma^i \longrightarrow C^i} \right\}}{B_1, \ldots, B_n, \Gamma \longrightarrow C}\ \circ\mathcal{R} \quad ,$$

where $\circ\mathcal{R}$ is any right rule, then Ξ reduces to

$$\dfrac{\left\{ \dfrac{\overset{\Pi_1}{\Delta_1 \longrightarrow B_1} \quad \cdots \quad \overset{\Pi_n}{\Delta_n \longrightarrow B_n} \quad \overset{\Pi^i}{B_1, \ldots, B_n, \Gamma^i \longrightarrow C^i}}{\Delta_1, \ldots, \Delta_n, \Gamma^i \longrightarrow C^i}\ mc \right\}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ \circ\mathcal{R} \quad .$$

<u>*Multicut cases:*</u>

$mc/\circ\mathcal{L}$**:** If Π ends with a left rule other than $c\mathcal{L}$ acting on B_1 and Π_1 ends with a multicut and reduces to Π_1', then Ξ reduces to

$$\dfrac{\overset{\Pi_1'}{\Delta_1 \longrightarrow B_1} \quad \overset{\Pi_2}{\Delta_2 \longrightarrow B_2} \quad \cdots \quad \overset{\Pi_n}{\Delta_n \longrightarrow B_n} \quad \overset{\Pi}{B_1, \ldots, B_n, \Gamma \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc \quad .$$

$-/mc$**:** Suppose Π is

$$\dfrac{\left\{ \overset{\Pi^j}{\{B_i\}_{i\in I^j}, \Gamma^j \longrightarrow D^j} \right\}_{j\in\{1..m\}} \quad \overset{\Pi'}{\{D^j\}_{j\in\{1..m\}}, \{B_i\}_{i\in I'}, \Gamma' \longrightarrow C}}{B_1, \ldots, B_n, \Gamma^1, \ldots, \Gamma^m, \Gamma' \longrightarrow C}\ mc \quad ,$$

where $I^1, \ldots, I^m, I'$ partition the formulas $\{B_i\}_{i\in\{1..n\}}$ among the premise derivations Π_1, $\ldots$, Π_m,Π'. For $1 \le j \le m$ let Ξ^j be

$$\dfrac{\left\{ \Delta_i \overset{\Pi_i}{\longrightarrow} B_i \right\}_{i\in I^j} \quad \overset{\Pi^j}{\{B_i\}_{i\in I^j}, \Gamma^j \longrightarrow D^j}}{\{\Delta_i\}_{i\in I^j}, \Gamma^j \longrightarrow D^j}\ mc \quad .$$

Then Ξ reduces to

$$\dfrac{\left\{ \ldots \overset{\Xi^j}{\longrightarrow} D^j \right\}_{j\in\{1..m\}} \quad \left\{ \Delta_i \overset{\Pi_i}{\longrightarrow} B_i \right\}_{i\in I'} \quad \ldots \overset{\Pi'}{\longrightarrow} C}{\Delta_1, \ldots, \Delta_n, \Gamma^1, \ldots \Gamma^m, \Gamma' \longrightarrow C}\ mc \quad .$$

<u>*Structural case:*</u>

$-/c\mathcal{L}$: If Π is

$$\dfrac{\overset{\Pi'}{B_1, B_1, B_2, \ldots, B_n, \Gamma \longrightarrow C}}{B_1, \ldots, B_n, \Gamma \longrightarrow C}\ c\mathcal{L} \quad ,$$

then Ξ reduces to

$$\dfrac{\dfrac{\Delta_1 \overset{\Pi_1}{\longrightarrow} B_1 \quad \left\{ \Delta_i \overset{\Pi_i}{\longrightarrow} B_i \right\}_{i\in\{1..n\}} \quad \overset{\Pi'}{B_1, B_1, B_2, \ldots, B_n, \Gamma \longrightarrow C}}{\Delta_1, \Delta_1, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ c\mathcal{L} \quad .$$

<u>*Axiom cases:*</u>

***init*$/\circ\mathcal{L}$:** If Π ends with either *nat*$\mathcal{L}$ or *def*$\mathcal{L}$ acting on B_1 and Π_1 ends with the *init* rule, then Ξ reduces to

$$\dfrac{\Delta_2 \overset{\Pi_2}{\longrightarrow} B_2 \quad \cdots \quad \Delta_n \overset{\Pi_n}{\longrightarrow} B_n \quad \overset{w(\Delta_1 \setminus B_1, \Pi)}{\Delta_1, B_2, \ldots, B_n, \Gamma \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc \quad .$$

$-/$*init*: If Π ends with the *init* rule and C is a formula in Γ, then Ξ reduces to

$$\dfrac{}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ init \quad .$$

If Π ends with the *init* rule, but C is not a formula in Γ, then C must be one of the cut formulas, say B_1. In this case Ξ reduces to $w(\Delta_2 \cup \ldots \cup \Delta_n \cup \Gamma, \Pi_1)$.

An inspection of the rules of the logic and this definition will reveal that every derivation ending with a multicut has a reduct. Because we use a multiset as the left side of the sequent, there may be ambiguity as to whether a formula occurring on the left side of the rightmost premise to a multicut rule is in fact a cut formula, and if so, which of the left premises corresponds to it. As a result, several of the reduction rules may apply, and so a derivation may have multiple reducts.

The following lemma states that the reduction relation is preserved by weakening.

Lemma 14 *If Ξ reduces to Ξ', then, for any multiset Δ of formulas, $w(\Delta, \Xi)$ reduces to $w(\Delta, \Xi')$.*

The proof of this lemma is a simple case analysis on the relevant clauses of Def. 13 and makes use of Def. 8 and Lemmas 11 and 12.

6 Normalizability and Reducibility

We now define two properties of derivations: normalizability and reducibility. Each of these properties implies that the derivation can be reduced to a cut-free derivation of the same end-sequent.

Definition 15 We define the set of *normalizable* derivations to be the smallest set that satisfies the following conditions:

1. If a derivation Π ends with a multicut, then it is normalizable if for every substitution θ there is a normalizable reduct of $\Pi\theta$.

2. If a derivation ends with any rule other than a multicut, then it is normalizable if the premise derivations are normalizable.

These clauses assert that a given derivation is normalizable provided certain (perhaps infinitely many) other derivations are normalizable. If we call these other derivations the predecessors of the given derivation, then a derivation is normalizable if and only if the tree of the derivation and its successive predecessors is well-founded. In this case, the well-founded tree is call the *normalization* of the derivation.

Since a normalization is well-founded, it has an associated induction principle: for any property P of derivations, if for every derivation Π in the normalization, P holds for every predecessor of Π implies that P holds for Π, then P holds for every derivation in the normalization.

Lemma 16 *If there is a normalizable derivation of a sequent, then there is a cut-free derivation of the sequent.*

Proof Let Π be a normalizable derivation of the sequent $\Gamma \longrightarrow B$. We show by induction on the normalization of Π that there is a cut-free derivation of $\Gamma \longrightarrow B$.

1. If Π ends with a multicut, then one of its reducts is one of its predecessors (by way of the empty substitution) and so is normalizable. But the reduct is also a derivation of $\Gamma \longrightarrow B$, so by the induction hypothesis this sequent has a cut-free derivation.

2. Suppose Π ends with a rule other than multicut. Since we are given that Π is normalizable, by definition the premise derivations are normalizable. These premise derivations are the predecessors of Π, so by the induction hypothesis there are cut-free derivations of the premises. Thus there is a cut-free derivation of $\Gamma \longrightarrow B$. ∎

The next two lemmas are also proved by induction on the normalization of the given derivation. The proof of the second lemma uses Lemmas 11 and 14 for the case when the derivation ends with a multicut.

Lemma 17 *If Π is a normalizable derivation, then for any substitution θ, $\Pi\theta$ is normalizable.*

Lemma 18 *If Π is a normalizable derivation, then for any multiset Δ of formulas, $w(\Delta, \Pi)$ is normalizable.*

We now define the property of reducibility for derivations. We do this by induction on the level of the derivation: in the definition of reducibility for derivations of level i we assume that reducibility is already defined for all levels $j < i$. (Recall from Definition 1 that the level of a derivation is defined to be the level of the consequent of its end-sequent.)

Definition 19 For any i, we define the set of *reducible* i-level derivations to be the smallest set of i-level derivations that satisfies the following conditions:

1. If a derivation Π ends with a multicut, then it is reducible if for every substitution θ there is a reducible reduct of $\Pi\theta$.

2. Suppose the derivation ends with the implication right rule

$$\dfrac{\overset{\Pi}{B, \Gamma \longrightarrow C}}{\Gamma \longrightarrow B \supset C} \supset \mathcal{R} \quad .$$

   Then the derivation is reducible if the premise derivation Π is reducible and, for every substitution θ, multiset Δ of formulas, and reducible derivation Π' of $\Delta \longrightarrow B\theta$, the derivation

$$\dfrac{\overset{\Pi'}{\Delta \longrightarrow B\theta} \quad \overset{\Pi\theta}{B\theta, \Gamma\theta \longrightarrow C\theta}}{\Delta, \Gamma\theta \longrightarrow C\theta} \; mc$$

   is reducible.

3. If the derivation ends with the implication left rule or the *nat* left rule, then it is reducible if the right premise derivation is reducible and the other premise derivations are normalizable.

4. If the derivation ends with any other rule, then it is reducible if the premise derivations are reducible.

These clauses assert that a given derivation is reducible provided certain (perhaps infinitely many) other derivations are reducible. If we call these other derivations the predecessors of the given derivation, then a derivation is reducible only if the tree of the derivation and its successive predecessors is well-founded. In this case, the well-founded tree is call the *reduction* of the derivation.

In defining reducibility for a derivation of $\Gamma \longrightarrow B \supset C$ ending with $\supset \mathcal{R}$ we quantify over reducible derivations of $\Delta \longrightarrow B\theta$. This is legitimate since we are defining reducibility for a derivation having level $\max(\mathrm{lvl}(B) + 1, \mathrm{lvl}(C))$, so the set of reducible derivations having level $\mathrm{lvl}(B\theta) = \mathrm{lvl}(B)$ is already defined. For a derivation ending with $\supset \mathcal{L}$ or $nat\mathcal{L}$, some premise derivations may have consequents with a higher level than that of the consequent of the conclusion. As a result, we cannot use the reducibility of those premise derivations to define the reducibility of the derivation as a whole, since the reducibility of the premise derivations may not yet be defined. Thus we use the weaker notion of normalizability for those premise derivations. Also observe that the consequent of the premise to the rule $def\mathcal{R}$ cannot have a higher level than the consequent of the conclusion because of the level restriction on definitional clauses. Finally, as with normalizations, reductions have associated induction principles.

The following lemmas are proved by induction on the reduction of the given derivation. The proof of Lemma 20 is straightforward. The proofs of Lemmas 21 and 22 use Lemmas 17 and 18, respectively, for the case when the derivation ends with $\supset \mathcal{L}$ or $nat\mathcal{L}$. The proof of Lemma 22 also requires Lemmas 11 and 14.

Lemma 20 *If a derivation is reducible, then it is normalizable.*

Lemma 21 *If* Π *is a reducible derivation, then for any substitution* θ*,* $\Pi\theta$ *is reducible.*

Lemma 22 *If* Π *is a reducible derivation, then for any multiset* Δ *of formulas,* $w(\Delta, \Pi)$ *is reducible.*

7 Cut-Elimination

In the previous section we proved that every reducible derivation is normalizable and that every normalizable derivation can be reduced to a cut-free derivation of the same end-sequent. In this section we show that every $FO\lambda^{\Delta\mathbb{N}}$ derivation is reducible, and thus every derivable sequent can be derived without the cut rule. The consistency of $FO\lambda^{\Delta\mathbb{N}}$ is then a simple corollary of the cut-elimination theorem.

Lemma 23 *For any derivation* Π *of* $B_1, \ldots, B_n, \Gamma \longrightarrow C$ *and reducible derivations* $\Pi_1, \ldots, \Pi_n$ *of* $\Delta_1 \longrightarrow B_1, \ldots, \Delta_n \longrightarrow B_n$ $(n \geq 0)$*, the derivation* Ξ

$$\cfrac{\overset{\Pi_1}{\Delta_1 \longrightarrow B_1} \quad \cdots \quad \overset{\Pi_n}{\Delta_n \longrightarrow B_n} \quad \overset{\Pi}{B_1, \ldots, B_n, \Gamma \longrightarrow C}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc$$

is reducible.

Proof The proof is by induction on $\mathrm{ht}(\Pi)$, with subordinate inductions on n and on the reductions of $\Pi_1, \ldots, \Pi_n$. The proof does not rely on the order of the inductions on reductions. Thus when we need to distinguish one of the Π_i, we shall refer to it as Π_1 without loss of generality.

The derivation Ξ is reducible if for every substitution θ some reduct of $\Xi\theta$ is reducible. If $n = 0$, then $\Xi\theta$ reduces to $\Pi\theta$. By Lemma 21 it suffices to show that Π is reducible. This is proved by a case analysis of the last rule in Π. For each case, the result follows easily from the outer induction hypothesis and Definition 19. The $\supset\mathcal{R}$ case requires that substitution for variables doesn't increase the measure of a derivation (Lemma 7). In the cases for $\supset\mathcal{L}$ and $nat\mathcal{L}$ we need the additional information that reducibility implies normalizability (Lemma 20).

For $n > 0$ we proceed with a case analysis of the reduction rules that apply to Ξ (and thus to $\Xi\theta$) to show that in fact every reduct of $\Xi\theta$ is reducible. Most cases follow easily from the induction hypothesis, Definition 19, and Lemmas 7, 10, 17, 18, 20, 21, and 22. We show the interesting cases below.

$\supset\mathcal{R}/\supset\mathcal{L}$**:** Π_1 and Π are

$$\cfrac{\overset{\Pi_1'}{B_1', \Delta_1 \longrightarrow B_1''}}{\Delta_1 \longrightarrow B_1' \supset B_1''}\supset\mathcal{R} \qquad\qquad \cfrac{\overset{\Pi'}{B_2, \ldots, B_n, \Gamma \longrightarrow B_1'} \quad \overset{\Pi''}{B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}}{B_1' \supset B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}\supset\mathcal{L}\ .$$

Recall that substitution for variables preserves reducibility (Lemma 21) and does not increase the measure of a derivation (Lemma 7). Thus the derivation Ξ_1

$$\cfrac{\overset{\Pi_2\theta}{\Delta_2\theta \longrightarrow B_2\theta} \quad \cdots \quad \overset{\Pi_n\theta}{\Delta_n\theta \longrightarrow B_n\theta} \quad \overset{\Pi'\theta}{B_2\theta, \ldots, B_n\theta, \Gamma\theta \longrightarrow B_1'\theta}}{\Delta_2\theta, \ldots, \Delta_n\theta, \Gamma\theta \longrightarrow B_1'\theta}\ mc$$

is reducible by the outer induction hypothesis. Since we are given that Π_1 is reducible, by Definition 19 the derivation Ξ_2

$$\dfrac{\begin{matrix}\Xi_1 \\ \Delta_2\theta, \ldots, \Delta_n\theta, \Gamma\theta \longrightarrow B_1'\theta\end{matrix} \quad \begin{matrix}\Pi_1'\theta \\ B_1'\theta, \Delta_1\theta \longrightarrow B_1''\theta\end{matrix}}{\Delta_1\theta, \ldots, \Delta_n\theta, \Gamma\theta \longrightarrow B_1''\theta}\ mc$$

is reducible. Therefore the derivation

$$\dfrac{\dfrac{\begin{matrix}\Xi_2 \\ \ldots \longrightarrow B_1''\theta\end{matrix} \quad \left\{\begin{matrix}\Pi_i\theta \\ \Delta_i\theta \longrightarrow B_i\theta\end{matrix}\right\}_{i\in\{2..n\}} \quad \begin{matrix}\Pi''\theta \\ B_1''\theta, \{B_i\theta\}_{i\in\{2..n\}}, \Gamma\theta \longrightarrow C\theta\end{matrix}}{\Delta_1\theta, \ldots, \Delta_n\theta, \Gamma\theta, \Delta_2\theta, \ldots, \Delta_n\theta, \Gamma\theta \longrightarrow C\theta}\ mc}{\Delta_1\theta, \ldots, \Delta_n\theta, \Gamma\theta \longrightarrow C\theta}\ c\mathcal{L},$$

which is the reduct of $\Xi\theta$, is reducible by the outer induction hypothesis and Definition 19.

$nat\mathcal{R}/nat\mathcal{L}$**:** Π_1 is

$$\dfrac{\begin{matrix}\Pi_1' \\ \Delta_1 \longrightarrow nat\ I\end{matrix}}{\Delta_1 \longrightarrow nat\ (s\ I)}\ nat\mathcal{R}$$

and Π is

$$\dfrac{\begin{matrix}\Pi' \\ \longrightarrow D\,z\end{matrix} \quad \begin{matrix}\Pi'' \\ D\,j \longrightarrow D\,(s\ j)\end{matrix} \quad \begin{matrix}\Pi''' \\ D\,(s\ I), B_2, \ldots, B_n, \Gamma \longrightarrow C\end{matrix}}{nat\ (s\ I), B_2, \ldots, B_n, \Gamma \longrightarrow C}\ nat\mathcal{L}.$$

Consider the derivation Ξ_1

$$\dfrac{\begin{matrix}\Pi_1' \\ \Delta_1 \longrightarrow nat\ I\end{matrix} \quad \dfrac{\begin{matrix}\Pi' \\ \longrightarrow D\,z\end{matrix} \quad \begin{matrix}\Pi'' \\ D\,j \longrightarrow D\,(s\ j)\end{matrix} \quad \dfrac{}{D\,I \longrightarrow D\,I}\ init}{nat\ I \longrightarrow D\,I}\ nat\mathcal{L}}{\Delta_1 \longrightarrow D\,I}\ mc.$$

Since the measure of the right premise derivation is no larger than ht(Π), Ξ_1 is reducible by induction on the reduction of Π_1 (Π_1' is a predecessor of Π_1). Again recall that substitution for variables preserves reducibility (Lemma 21) and does not increase the measure of a derivation (Lemma 7). The derivation Ξ_2

$$\dfrac{\begin{matrix}\Xi_1\theta \\ \Delta_1\theta \longrightarrow D\theta\, I\theta\end{matrix} \quad \begin{matrix}\Pi''\theta[I\theta/j] \\ D\theta\, I\theta \longrightarrow D\theta\,(s\ I\theta)\end{matrix}}{\Delta_1\theta \longrightarrow D\theta\,(s\ I\theta)}\ mc$$

is then reducible by the outer induction hypothesis. Therefore the derivation

$$\dfrac{\begin{matrix}\Xi_2 \\ \Delta_1\theta \longrightarrow D\theta\,(s\ I\theta)\end{matrix} \quad \left\{\begin{matrix}\Pi_i\theta \\ \Delta_i\theta \longrightarrow B_i\theta\end{matrix}\right\}_{i\in\{2..n\}} \quad \begin{matrix}\Pi'''\theta \\ \ldots \longrightarrow C\theta\end{matrix}}{\Delta_1\theta, \ldots, \Delta_n\theta, \Gamma\theta \longrightarrow C\theta}\ mc,$$

which is the reduct of $\Xi\theta$, is reducible by the outer induction hypothesis.

$def\mathcal{L}/\circ\mathcal{L}$**:** Π_1 and $\Pi_1\theta$ are

$$\cfrac{\left\{\begin{matrix}\Pi_1^{\rho,\sigma,D}\\ D\sigma,\Delta_1'\rho\longrightarrow B_1\rho\end{matrix}\right\}}{A,\Delta_1'\longrightarrow B_1}\;def\mathcal{L}\qquad\qquad \cfrac{\left\{\begin{matrix}\Pi_1^{\theta\circ\rho',\sigma',D}\\ D\sigma',\Delta_1'\theta\rho'\longrightarrow B_1\theta\rho'\end{matrix}\right\}}{A\theta,\Delta_1'\theta\longrightarrow B_1\theta}\;def\mathcal{L}\quad.$$

The derivation $\Xi^{\rho',\sigma',D}$

$$\cfrac{\begin{matrix}\Pi_1^{\theta\circ\rho',\sigma',D}\\ D\sigma',\Delta_1'\theta\rho'\longrightarrow B_1\theta\rho'\end{matrix}\quad\left\{\begin{matrix}\Pi_i\theta\rho'\\ \Delta_i\theta\rho'\longrightarrow B_i\theta\rho'\end{matrix}\right\}_{i\in\{2..n\}}\quad\begin{matrix}\Pi\theta\rho'\\ \ldots\longrightarrow C\theta\rho'\end{matrix}}{D\sigma',\Delta_1'\theta\rho',\Delta_2\theta\rho',\ldots,\Delta_n\theta\rho',\Gamma\theta\rho'\longrightarrow C\theta\rho'}\;mc\quad.$$

is reducible by Lemmas 7 and 21 and induction on the reduction of Π_1 ($\Pi_1^{\theta\circ\rho',\sigma',D}$ is a predecessor of Π_1). Therefore the derivation

$$\cfrac{\left\{\begin{matrix}\Xi^{\rho',\sigma',D}\\ D\sigma',\Delta_1'\theta\rho',\Delta_2\theta\rho',\ldots,\Delta_n\theta\rho',\Gamma\theta\rho'\longrightarrow C\theta\rho'\end{matrix}\right\}}{A\theta,\Delta_1'\theta,\Delta_2\theta,\ldots,\Delta_n\theta,\Gamma\theta\longrightarrow C\theta}\;def\mathcal{L}\quad,$$

which is the reduct of $\Xi\theta$, is reducible by Definition 19.

$-/\supset\mathcal{R}$**:** Ξ has the form

$$\cfrac{\begin{matrix}\Pi_1\\ \Delta_1\longrightarrow B_1\end{matrix}\quad\cdots\quad\begin{matrix}\Pi_n\\ \Delta_n\longrightarrow B_n\end{matrix}\quad\cfrac{\begin{matrix}\Pi'\\ C',B_1,\ldots,B_n,\Gamma\longrightarrow C''\end{matrix}}{B_1,\ldots,B_n,\Gamma\longrightarrow C'\supset C''}\supset\mathcal{R}}{\Delta_1,\ldots,\Delta_n,\Gamma\longrightarrow C'\supset C''}\;mc\quad.$$

Once again recall that substitution for variables preserves reducibility (Lemma 21) and does not increase the measure of a derivation (Lemma 7). The derivation Ξ_1

$$\cfrac{\begin{matrix}\Pi_1\theta\\ \Delta_1\theta\longrightarrow B_1\theta\end{matrix}\quad\cdots\quad\begin{matrix}\Pi_n\theta\\ \Delta_n\theta\longrightarrow B_n\theta\end{matrix}\quad\begin{matrix}\Pi'\theta\\ C'\theta,B_1\theta,\ldots,B_n\theta,\Gamma\theta\longrightarrow C''\theta\end{matrix}}{C'\theta,\Delta_1\theta,\ldots,\Delta_n\theta,\Gamma\theta\longrightarrow C''\theta}\;mc$$

is reducible by the outer induction hypothesis. For any substitutions θ' and θ'' and reducible derivation Ξ', the derivation

$$\cfrac{\begin{matrix}\Xi'\theta''\\ (\Delta'\longrightarrow C'\theta\theta')\theta''\end{matrix}\quad\left\{\begin{matrix}\Pi_i\theta\theta'\theta''\\ (\Delta_i\longrightarrow B_i)\theta\theta'\theta''\end{matrix}\right\}_{i\in\{1..n\}}\quad\begin{matrix}\Pi'\theta\theta'\theta''\\ (\ldots\longrightarrow C'')\theta\theta'\theta''\end{matrix}}{\Delta'\theta'',\Delta_1\theta\theta'\theta'',\ldots,\Delta_n\theta\theta'\theta'',\Gamma\theta\theta'\theta''\longrightarrow C''\theta\theta'\theta''}\;mc$$

is reducible by the outer induction hypothesis. This is a reduct of $\Xi_2\theta''$, where Ξ_2 is

$$\cfrac{\begin{matrix}\Xi'\\ \Delta'\longrightarrow C'\theta\theta'\end{matrix}\quad\begin{matrix}\Xi_1\theta'\\ C'\theta\theta',\Delta_1\theta\theta',\ldots,\Delta_n\theta\theta',\Gamma\theta\theta'\longrightarrow C''\theta\theta'\end{matrix}}{\Delta',\Delta_1\theta\theta',\ldots,\Delta_n\theta\theta',\Gamma\theta\theta'\longrightarrow C''\theta\theta'}\;mc\quad.$$

Since a reduct of $\Xi_2\theta''$ is reducible for every θ'', by Definition 19 Ξ_2 is reducible. Since Ξ_1 is reducible and Ξ_2 is reducible for every substitution θ' and reducible derivation Ξ', by Definition 19

$$\frac{\begin{array}{c}\Xi_1\\ C'\theta, \Delta_1\theta, \ldots, \Delta_n\theta, \Gamma\theta \longrightarrow C''\theta\end{array}}{\Delta_1\theta, \ldots, \Delta_n\theta, \Gamma\theta \longrightarrow C'\theta \supset C''\theta} \supset \mathcal{R}$$

is reducible. This last derivation is the reduct of $\Xi\theta$ by the current reduction rule. ∎

Corollary 24 *Every derivation is reducible.*

Proof This result follows immediately from Lemma 23 with $n = 0$. ∎

Theorem 25 *If a sequent is derivable, then there is a cut-free derivation of the sequent.*

Proof This result follows immediately from Corollary 24, Lemma 20, and Lemma 16. ∎

Since there is no right rule for $\bot$, there is no cut-free derivation of $\longrightarrow \bot$. Thus consistency is a simple corollary of cut-elimination.

Corollary 26 *There is no* $FO\lambda^{\Delta\mathbb{N}}$ *derivation of the sequent* $\longrightarrow \bot$.

8 Related Work

The logic $FO\lambda^{\Delta\mathbb{N}}$ is related to Schroeder-Heister's "logics with definitional reflection" [24]. He proved cut-elimination for two logics: the first without contraction but allowing arbitrary implications in definitions, the second with contraction but only implication-free definitions. He also showed a counter-example to cut-elimination for the logic with both contraction and definitions with arbitrary implications, but conjectured that cut-elimination should hold if the definitions were stratified (as we accomplish in $FO\lambda^{\Delta\mathbb{N}}$ through the level restriction). The proof presented in this paper clearly establishes that Schroeder-Heister's conjecture is true.

However, there are significant differences between Schroeder-Heister's logics and ours. The first is that $FO\lambda^{\Delta\mathbb{N}}$ uses a stronger version of the left rule for definitions; Schroeder-Heister has extended his cut-elimination results to logics with this stronger rule [25]. More significantly, Schroeder-Heister has no induction rules in his logics. Because of the presence of the $nat\mathcal{L}$ rule in $FO\lambda^{\Delta\mathbb{N}}$, Schroeder-Heister's cut-elimination proofs do not extend to our setting.

The proof of cut-elimination presented in this paper is patterned after Martin-Löf's normalization proof for a natural deduction system with iterated inductive definitions [12]. Our work can be viewed as an adaptation of his to the sequent calculus setting: our rules for definitions and natural numbers roughly correspond to his introduction and elimination rules for inductively defined predicates.

Acknowledgments

We gratefully acknowledge interaction with Catuscia Palamidessi which led to the introduction of levels to allow restricted use of implication in definitions. The authors have been funded in part by the grants ONR N00014-93-1-1324, NSF CCR-92-09224, NSF CCR-94-00907, ARO DAAH04-95-1-0092, and AFOSR F49620-96-1-0204.

References

[1] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[2] Jawahar Chirimar. *Proof Theoretic Approach to Specification Languages.* PhD thesis, University of Pennsylvania, February 1995.

[3] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[4] K. L. Clark. Negation as failure. In J. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[5] Lars-Henrik Eriksson. A finitary version of the calculus of partial inductive definitions. In L.-H. Eriksson, L. Hallnäs, and P. Schroeder-Heister, editors, *Proceedings of the Second International Workshop on Extensions to Logic Programming*, volume 596 of *Lecture Notes in Artificial Intelligence*, pages 89–134. Springer-Verlag, 1991.

[6] Lars-Henrik Eriksson. Pi: an interactive derivation editor for the calculus of partial inductive definitions. In A. Bundy, editor, *Proceedings of the Twelfth International Conference on Automated Deduction*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 821–825. Springer-Verlag, June 1994.

[7] Gerhard Gentzen. Investigations into logical deductions, 1935. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland Publishing Co., 1969.

[8] Jean-Yves Girard. Une extension de l'interprétation de Gödel à l'analyse, et son application à l'émination des coupures dans l'analyse et la théorie des types. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*, pages 63–92, Amsterdam, 1971. North-Holland.

[9] Jean-Yves Girard. A fixpoint theorem in linear logic. A message posted on the `linear@cs.stanford.edu` mailing list, `http://www.csl.sri.com/linear/mailing-list-traffic/www/07/mail_3.html`, February 1992.

[10] John Hannan. Extended natural semantics. *Journal of Functional Programming*, 3(2):123–152, April 1993.

[11] Gérard Huet. A unification algorithm for typed λ-calculus. *Theoretical Computer Science*, 1:27–57, 1975.

[12] Per Martin-Löf. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*, pages 179–216. North-Holland, 1971.

[13] Raymond McDowell. *Reasoning in a Logic with Definitions and Induction.* PhD thesis, University of Pennsylvania, December 1997.

[14] Raymond McDowell and Dale Miller. A logic for reasoning with higher-order abstract syntax. In G. Winskel, editor, *Twelfth Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, June 1997.

[15] Raymond McDowell, Dale Miller, and Catuscia Palamidessi. Encoding transition systems in sequent calculus: Preliminary report. In *Proceedings of the 1996 Workshop on Linear Logic*, volume 3 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1996.

[16] Raymond McDowell, Dale Miller, and Catuscia Palamidessi. Encoding transition systems in sequent calculus. *Theoretical Computer Science*, to appear. Preliminary version appeared as [15].

[17] Dale Miller. Forum: A multiple-conclusion specification language. *Theoretical Computer Science*, 165:201–232, 1996.

[18] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

[19] Gopalan Nadathur and Dale Miller. Higher-order Horn clauses. *Journal of the ACM*, 37(4):777–814, October 1990.

[20] David Patterson and John Hennessy. *Computer Architecture: A Quantitative Approach*. Morgan Kaufman Publishers, Inc., 1990.

[21] Christine Paulin-Mohring. Inductive definitions in the system Coq: Rules and properties. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, pages 328–345, 1993.

[22] Frank Pfenning. The practice of logical frameworks. In Hélène Kirchner, editor, *Proceedings of the Colloquium on Trees in Algebra and Programming*, volume 1059 of *Springer-Verlag LNCS*, pages 119–134, April 1996.

[23] Frank Pfenning and Conal Elliot. Higher-order abstract syntax. In *Proceedings of the ACM-SIGPLAN Conference on Programming Language Design and Implementation*, pages 199–208. ACM Press, June 1988.

[24] Peter Schroeder-Heister. Cut-elimination in logics with definitional reflection. In D. Pearce and H. Wansing, editors, *Nonclassical Logics and Information Processing*, volume 619 of *Lecture Notes in Computer Science*, pages 146–171. Springer-Verlag, 1992.

[25] Peter Schroeder-Heister. Rules of definitional reflection. In M. Vardi, editor, *Proceedings, Eighth Annual IEEE Symposium on Logic in Computer Science*, pages 222–232. IEEE Computer Society Press, June 1993.

[26] Carsten Schürmann and Frank Pfenning. Automated theorem proving in a simple meta-logic for lf. To appear at CADE-15, Lindau, Germany, July 1998.

[27] John Slaney. Solution to a problem of Ono and Komori. *Journal of Philosophic Logic*, 18:103–111, 1989.

[28] R. F. Stärk. Cut-property and negation as failure. *International Journal of Foundations of Computer Science*, 5(2):129–164, 1994.

[29] W.W. Tait. Intensional interpretations of functionals of finite type I. *Journal of Symbolic Logic*, 32(2):198–212, 1967.

Papers by McDowell are available via anonymous ftp from `ftp.cis.upenn.edu` in `pub/papers/mcdowell` or via the WWW at `http://www.cis.upenn.edu/~mcdowell`. Papers by Miller are available via anonymous ftp from `ftp.cse.psu.edu` in `pub/dale` or via the WWW at `http://www.cse.psu.edu/~dale`.