Cut-Simulation in Impredicative Logics

Christoph E. Benzmüller[1], Chad E. Brown[1], and Michael Kohlhase[2]

[1] Saarland University, Saarbrücken, Germany (chris|cebrown@ags.uni-sb.de)
[2] International University Bremen, Bremen, Germany (m.kohlhase@iu-bremen.de)

Abstract. We investigate cut-elimination and cut-simulation in impredicative (higher-order) logics. We illustrate that adding simple axioms such as Leibniz equations to a calculus for an impredicative logic — in our case a sequent calculus for classical type theory — is like adding cut. The phenomenon equally applies to prominent axioms like Boolean- and functional extensionality, induction, choice, and description. This calls for the development of calculi where these principles are built-in instead of being treated axiomatically.

1 Introduction

One of the key questions of automated reasoning is the following: "When does a set Φ of sentences have a model?" In fact, given reasonable assumptions about calculi, most inference problems can be reduced to determining (un)-satisfiability of a set Φ of sentences. Since building models for Φ is hard in practice, much research in computational logic has concentrated on finding sufficient conditions for satisfiability, e.g. whether there is a Hintikka set $\mathcal{H}$ extending Φ.

Of course in general the answer to the satisfiability question depends on the class of models at hand. In classical first-order logic, model classes are well-understood. In impredicative higher-order logic, there is a whole landscape of plausible model classes differing in their treatment of functional and Boolean extensionality. Satisfiability then strongly depends on these classes, for instance, the set $\Phi := \{a, b, qa, \neg qb\}$ is unsatisfiable in a model class where the universes of Booleans are required to have at most two members (see property $\mathfrak{b}$ below), but satisfiable in the class without this restriction.

In [5] we have shown that certain (i.e. *saturated*) Hintikka sets always have models and have derived syntactical conditions (so-called *saturated* abstract consistency properties) for satisfiability from this fact. The importance of abstract consistency properties is that one can check completeness for a calculus $\mathcal{C}$ by verifying proof-theoretic conditions (checking that $\mathcal{C}$-irrefutable sets of formulae have the saturated abstract consistency property) instead of performing model-theoretic analysis (for historical background of the method in first-order logic, cf. [10, 13, 14]). Unfortunately, the saturation condition (if Φ is abstractly consistent, then one of $\Phi \cup \{\mathbf{A}\}$ or $\Phi \cup \{\neg\mathbf{A}\}$ is as well for all sentences $\mathbf{A}$) is very difficult to prove for machine-oriented calculi (indeed as hard as cut elimination).

In this paper we investigate further the relation between the lack of the subformula property in the saturation condition (we need to "guess" whether to extend Φ by $\mathbf{A}$ or $\neg\mathbf{A}$ on our way to a Hintikka set for all sentences $\mathbf{A}$) and the cut rule (where we have to "guess, i.e. search for in an automated reasoning setting" the cut formula $\mathbf{A}$). A side result is the insight that there exist "cut-strong" formulae which support the effective simulation of cut in calculi for impredicative logics.

In Section 2, we will fix notation and review the relevant results from [5]. We define in Section 3 a basic sequent calculus and study the correspondence between saturation in abstract consistency classes and cut-elimination. In Section 4 we introduce the notion of "cut-strong" formulae and sequents and show that they support the effective simulation of cut. In Section 5 we demonstrate that the pertinent extensionality axioms are cut-strong. We develop alternative extensionality rules which do not suffer from this problem. Further rules are needed to ensure Henkin completeness for this calculus with extensionality. These new rules correspond to the acceptability conditions we propose in Section 6 to ensure the existence of models and the existence of saturated extensions of abstract consistence classes.

2 Higher-Order Logic

In [5] we have re-examined the semantics of classical higher-order logic with the purpose of clarifying the role of extensionality. For this we have defined eight classes of higher-order models with respect to various combinations of Boolean extensionality and three forms of functional extensionality. We have also developed a methodology of abstract consistency (by providing the necessary model existence theorems) needed for instance, to analyze completeness of higher-order calculi with respect to these model classes. We now briefly summarize the main notions and results of [5] as required for this paper. Our impredicative logic of choice is Church's classical type theory.

Syntax: Church's Simply Typed λ-Calculus. As in [9], we formulate higher-order logic ($\mathcal{HOL}$) based on the simply typed λ-calculus. The set of simple types $\mathcal{T}$ is freely generated from basic types o and ι using the function type constructor $\rightarrow$.

For formulae we start with a set $\mathcal{V}$ of (typed) variables (denoted by $X_\alpha, Y, Z, X^1_\beta, X^2_\gamma \ldots$) and a signature Σ of (typed) constants (denoted by $c_\alpha, f_{\alpha\rightarrow\beta}, \ldots$). We let $\mathcal{V}_\alpha$ (Σ_α) denote the set of variables (constants) of type α. The signature Σ of constants includes the logical constants $\neg_{o\rightarrow o}$, $\vee_{o\rightarrow o\rightarrow o}$ and $\Pi^\alpha_{(\alpha\rightarrow o)\rightarrow o}$ for each type α; all other constants in Σ are called parameters. As in [5], we assume there is an infinite cardinal $\aleph_s$ such that the cardinality of Σ_α is $\aleph_s$ for each type α (cf. [5](3.16)). The set of $\mathcal{HOL}$-formulae (or terms) are constructed from typed variables and constants using application and λ-abstraction. We let $\mathit{wff}_\alpha(\Sigma)$ be the set of all terms of type α and $\mathit{wff}(\Sigma)$ be the set of all terms.

We use vector notation to abbreviate k-fold applications and abstractions as $\mathbf{A}\overline{\mathbf{U}^k}$ and $\lambda\overline{X^k}.\mathbf{A}$, respectively. We also use Church's dot notation so that **.** stands for a (missing) left bracket whose mate is as far to the right as possible (consistent with given brackets). We use infix notation $\mathbf{A} \vee \mathbf{B}$ for $((\vee\mathbf{A})\mathbf{B})$ and binder notation $\forall X_\alpha\text{.}\mathbf{A}$ for $(\Pi^\alpha(\lambda X_\alpha\text{.}\mathbf{A}_o))$. We further use $\mathbf{A} \wedge \mathbf{B}$, $\mathbf{A} \Rightarrow \mathbf{B}$, $\mathbf{A} \Leftrightarrow \mathbf{B}$ and $\exists X_\alpha\text{.}\mathbf{A}$ as shorthand for formulae defined in terms of $\neg$, $\vee$ and Π^α (cf. [5]). Finally, we let $(\mathbf{A}_\alpha \doteq^\alpha \mathbf{B}_\alpha)$ denote the Leibniz equation $\forall P_{\alpha\to o}\text{.}(P\mathbf{A}) \Rightarrow \text{.}P\mathbf{B}$.

Each occurrence of a variable in a term is either bound by a λ or free. We use $free(\mathbf{A})$ to denote the set of free variables of $\mathbf{A}$ (i.e., variables with a free occurrence in $\mathbf{A}$). We consider two terms to be equal if the terms are the same up to the names of bound variables (i.e., we consider α-conversion implicitly). A term $\mathbf{A}$ is closed if $free(\mathbf{A})$ is empty. We let $cwff_\alpha(\Sigma)$ denote the set of closed terms of type α and $cwff(\Sigma)$ denote the set of all closed terms. Each term $\mathbf{A} \in wff_o(\Sigma)$ is called a proposition and each term $\mathbf{A} \in cwff_o(\Sigma)$ is called a sentence.

We denote substitution of a term $\mathbf{A}_\alpha$ for a variable X_α in a term $\mathbf{B}_\beta$ by $[\mathbf{A}/X]\mathbf{B}$. Since we consider α-conversion implicitly, we assume the bound variables of $\mathbf{B}$ avoid variable capture.

Two common relations on terms are given by β-reduction and η-reduction. A β-redex $(\lambda X\text{.}\mathbf{A})\mathbf{B}$ β-reduces to $[\mathbf{B}/X]\mathbf{A}$. An η-redex $(\lambda X\text{.}\mathbf{C}X)$ (where $X \notin free(\mathbf{C})$) η-reduces to $\mathbf{C}$. For $\mathbf{A}, \mathbf{B} \in wff_\alpha(\Sigma)$, we write $\mathbf{A} \equiv_\beta \mathbf{B}$ to mean $\mathbf{A}$ can be converted to $\mathbf{B}$ by a series of β-reductions and expansions. Similarly, $\mathbf{A} \equiv_{\beta\eta} \mathbf{B}$ means $\mathbf{A}$ can be converted to $\mathbf{B}$ using both β and η. For each $\mathbf{A} \in wff(\Sigma)$ there is a unique β-normal form (denoted $\mathbf{A}{\downarrow_\beta}$) and a unique $\beta\eta$-normal form (denoted $\mathbf{A}{\downarrow_{\beta\eta}}$). From this fact we know $\mathbf{A} \equiv_\beta \mathbf{B}$ ($\mathbf{A} \equiv_{\beta\eta} \mathbf{B}$) iff $\mathbf{A}{\downarrow_\beta} \equiv \mathbf{B}{\downarrow_\beta}$ ($\mathbf{A}{\downarrow_{\beta\eta}} \equiv \mathbf{B}{\downarrow_{\beta\eta}}$).

A non-atomic formula in $wff_o(\Sigma)$ is any formula whose β-normal form is of the form $[c\overline{\mathbf{A}^n}]$ where c is a logical constant. An atomic formula is any other formula in $wff_o(\Sigma)$.

Semantics: Eight Model Classes. For each $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$ (the latter set will be abbreviated by ⧉ in the remainder) we define $\mathfrak{M}_*$ to be the class of all Σ-models $\mathcal{M}$ such that $\mathcal{M}$ satisfies property $\mathfrak{q}$ and each of the additional properties $\{\eta, \xi, \mathfrak{f}, \mathfrak{b}\}$ indicated in the subscript $*$ (cf. [5](3.49)). Special cases of Σ-models are Henkin models and standard models (cf. [5](3.50 and 3.51)). Every model in $\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$ is isomorphic to a Henkin model (see the discussion following [5](3.68)).

Saturated Abstract Consistency Classes and Model Existence. Finally, we review the model existence theorems proved in [5]. There are three stages to obtaining a model in our framework. First, we obtain an abstract consistency class Γ_Σ (usually defined as the class of irrefutable sets of sentences with respect to some calculus). Second, given a (sufficiently pure) set of sentences Φ in the abstract consistency class Γ_Σ we construct a Hintikka set $\mathcal{H}$ extending Φ. Third, we construct a model of this Hintikka set (and hence a model of Φ).

A Σ-abstract consistency class Γ_Σ is a class of sets of Σ-sentences. An abstract consistency class is always required to be closed under subsets (cf. [5](6.1)). Sometimes we require the stronger property that Γ_Σ is compact, i.e., a set Φ is in Γ_Σ iff every finite subset of Φ is in Γ_Σ (cf. [5](6.1,6.2)).

To describe further properties of abstract consistency classes, we use the notation $S * a$ for $S \cup \{a\}$ as in [5]. The following is a list of properties a class Γ_Σ of sets of sentences can satisfy with respect to arbitrary $\Phi \in \Gamma_\Sigma$ (cf. [5](6.5)):

∇_c If $\mathbf{A}$ is atomic, then $\mathbf{A} \notin \Phi$ or $\neg\mathbf{A} \notin \Phi$.
$\nabla_\neg$ If $\neg\neg\mathbf{A} \in \Phi$, then $\Phi * \mathbf{A} \in \Gamma_\Sigma$.
∇_β If $\mathbf{A} \equiv_\beta \mathbf{B}$ and $\mathbf{A} \in \Phi$, then $\Phi * \mathbf{B} \in \Gamma_\Sigma$.
∇_η If $\mathbf{A} \equiv_{\beta\eta} \mathbf{B}$ and $\mathbf{A} \in \Phi$, then $\Phi * \mathbf{B} \in \Gamma_\Sigma$.
$\nabla_\vee$ If $\mathbf{A} \vee \mathbf{B} \in \Phi$, then $\Phi * \mathbf{A} \in \Gamma_\Sigma$ or $\Phi * \mathbf{B} \in \Gamma_\Sigma$.
$\nabla_\wedge$ If $\neg(\mathbf{A} \vee \mathbf{B}) \in \Phi$, then $\Phi * \neg\mathbf{A} * \neg\mathbf{B} \in \Gamma_\Sigma$.
$\nabla_\forall$ If $\Pi^\alpha \mathbf{F} \in \Phi$, then $\Phi * \mathbf{FW} \in \Gamma_\Sigma$ for each $\mathbf{W} \in cwff_\alpha(\Sigma)$.
$\nabla_\exists$ If $\neg\Pi^\alpha \mathbf{F} \in \Phi$, then $\Phi * \neg(\mathbf{F}w) \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in any sentence of Φ.
$\nabla_\mathfrak{b}$ If $\neg(\mathbf{A} \doteq^o \mathbf{B}) \in \Phi$, then $\Phi * \mathbf{A} * \neg\mathbf{B} \in \Gamma_\Sigma$ or $\Phi * \neg\mathbf{A} * \mathbf{B} \in \Gamma_\Sigma$.
∇_ξ If $\neg(\lambda X_\alpha.\mathbf{M} \doteq^{\alpha\to\beta} \lambda X_\alpha.\mathbf{N}) \in \Phi$, then $\Phi * \neg([w/X]\mathbf{M} \doteq^\beta [w/X]\mathbf{N}) \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in any sentence of Φ.
$\nabla_\mathfrak{f}$ If $\neg(\mathbf{G} \doteq^{\alpha\to\beta} \mathbf{H}) \in \Phi$, then $\Phi * \neg(\mathbf{G}w \doteq^\beta \mathbf{H}w) \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in any sentence of Φ.
∇_{sat} Either $\Phi * \mathbf{A} \in \Gamma_\Sigma$ or $\Phi * \neg\mathbf{A} \in \Gamma_\Sigma$.

We say Γ_Σ is an abstract consistency class if it is closed under subsets and satisfies $\nabla_c, \nabla_\neg, \nabla_\beta, \nabla_\vee, \nabla_\wedge, \nabla_\forall$ and $\nabla_\exists$. We let $\mathfrak{Acc}_\beta$ denote the collection of all abstract consistency classes. For each $* \in \boxplus$ we refine $\mathfrak{Acc}_\beta$ to a collection $\mathfrak{Acc}_*$ where the additional properties $\{\nabla_\eta, \nabla_\xi, \nabla_\mathfrak{f}, \nabla_\mathfrak{b}\}$ indicated by $*$ are required (cf. [5](6.7)). We say an abstract consistency class Γ_Σ is saturated if ∇_{sat} holds.

Using ∇_c (atomic consistency) and the fact that there are infinitely many parameters at each type, we can show every abstract consistency class satisfies non-atomic consistency. That is, for every abstract consistency class Γ_Σ, $\mathbf{A} \in cwff_o(\Sigma)$ and $\Phi \in \Gamma_\Sigma$, we have either $\mathbf{A} \notin \Phi$ or $\neg\mathbf{A} \notin \Phi$ (cf. [5](6.10)).

In [5](6.32) we show that sufficiently Σ-pure sets in saturated abstract consistency classes extend to saturated Hintikka sets. (A set of sentences Φ is sufficiently Σ-pure if for each type α there is a set $\mathcal{P}_\alpha$ of parameters of type α with cardinality $\aleph_s$ and such that no parameter in $\mathcal{P}$ occurs in a sentence in Φ.)

In the Model Existence Theorem for Saturated Sets [5](6.33) we show that these saturated Hintikka sets can be used to construct models $\mathcal{M}$ which are members of the corresponding model classes $\mathfrak{M}_*$. Then we conclude (cf. [5](6.34)):

Model Existence Theorem for Saturated Abstract Consistency Classes: *For all $* \in \boxplus$, if Γ_Σ is a saturated abstract consistency class in $\mathfrak{Acc}_*$ and $\Phi \in \Gamma_\Sigma$ is a sufficiently Σ-pure set of sentences, then there exists a model $\mathcal{M} \in \mathfrak{M}_*$ that satisfies Φ. Furthermore, each domain of $\mathcal{M}$ has cardinality at most $\aleph_s$.*

In [5] we apply the abstract consistency method to analyze completeness for different natural deduction calculi. Unfortunately, the saturation condition is very difficult to prove for machine-oriented calculi (indeed as we will see in Section 3 it is equivalent to cut elimination), so Theorem [5](6.34) cannot be easily used for this purpose directly.

In Section 6 we therefore motivate and present a set of extra conditions for $\mathfrak{Acc}_{\beta\mathfrak{fb}}$ we call **acceptability** conditions. The new conditions are sufficient to prove model existence.

Basic Rules

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta * \mathbf{A} * \neg\mathbf{A}}\,\mathcal{G}(init) \qquad \frac{\Delta * \mathbf{A}}{\Delta * \neg\neg\mathbf{A}}\,\mathcal{G}(\neg)$$

$$\frac{\Delta * \neg\mathbf{A} \quad \Delta * \neg\mathbf{B}}{\Delta * \neg(\mathbf{A} \lor \mathbf{B})}\,\mathcal{G}(\lor_-) \qquad \frac{\Delta * \mathbf{A} * \mathbf{B}}{\Delta * (\mathbf{A} \lor \mathbf{B})}\,\mathcal{G}(\lor_+)$$

$$\frac{\Delta * \neg\,(\mathbf{A}\mathbf{C}){\downarrow_\beta} \quad \mathbf{C} \in cwff_\alpha(\Sigma)}{\Delta * \neg\Pi^\alpha\mathbf{A}}\,\mathcal{G}(\Pi_-^{\mathbf{C}}) \qquad \frac{\Delta * \,(\mathbf{A}c){\downarrow_\beta} \quad c_\alpha \in \Sigma \text{ } new}{\Delta * \Pi^\alpha\mathbf{A}}\,\mathcal{G}(\Pi_+^c)$$

Inversion Rule

$$\frac{\Delta * \neg\neg\mathbf{A}}{\Delta * \mathbf{A}}\,\mathcal{G}(Inv^\neg)$$

Weakening and Cut Rules

$$\frac{\Delta}{\Delta \cup \Delta'}\,\mathcal{G}(weak) \qquad \frac{\Delta * \mathbf{C} \quad \Delta * \neg\mathbf{C}}{\Delta}\,\mathcal{G}(cut)$$

Fig. 1. Sequent Calculus Rules

3 Sequent Calculi, Cut and Saturation

We will now study cut-elimination and cut-simulation with respect to (one-sided) sequent calculi.

Sequent Calculi $\mathcal{G}$. We consider a sequent to be a finite set Δ of β-normal sentences from $cwff_o(\Sigma)$. A sequent calculus $\mathcal{G}$ provides an inductive definition for when $\Vdash_\mathcal{G} \Delta$ holds. We say a sequent calculus rule

$$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Delta}\,r$$

is **admissible** in $\mathcal{G}$ if $\Vdash_\mathcal{G} \Delta$ holds whenever $\Vdash_\mathcal{G} \Delta_i$ for all $1 \leq i \leq n$. For any natural number $k \geq 0$, we call an admissible rule r k**-admissible** if any instance of r can be replaced by a derivation with at most k additional proof steps. Given a sequent Δ, a model $\mathcal{M}$, and a class $\mathfrak{M}$ of models, we say Δ is *valid for* $\mathcal{M}$ (or *valid for* $\mathfrak{M}$), if $\mathcal{M} \models \mathbf{D}$ for some $\mathbf{D} \in \Delta$ (or Δ is valid for every $\mathcal{M} \in \mathfrak{M}$). As for sets in abstract consistency classes, we use the notation $\Delta * \mathbf{A}$ to denote the set $\Delta \cup \{\mathbf{A}\}$ (which is simply Δ if $\mathbf{A} \in \Delta$). Figure 1 introduces several sequent calculus rules. Some of these rules will be used to define sequent calculi, while others will be shown admissible (or even k-admissible).

Abstract Consistency Classes for Sequent Calculi. For any sequent calculus $\mathcal{G}$ we can define a class $\Gamma_\Sigma^\mathcal{G}$ of sets of sentences. Under certain assumptions, $\Gamma_\Sigma^\mathcal{G}$ is an abstract consistency class. First we adopt the notation $\neg\Phi$ and $\Phi{\downarrow_\beta}$ for the sets $\{\neg\mathbf{A} | \mathbf{A} \in \Phi\}$ and $\{\,\mathbf{A}{\downarrow_\beta} | \mathbf{A} \in \Phi\}$, resp., where $\Phi \subseteq cwff_o(\Sigma)$. Furthermore, we assume this use of $\neg$ binds more strongly than $\cup$ or $*$, so that $\neg\Phi \cup \Delta$ means $(\neg\Phi) \cup \Delta$ and $\neg\Phi * \mathbf{A}$ means $(\neg\Phi) * \mathbf{A}$.

Definition 1 *Let $\mathcal{G}$ be a sequent calculus. We define $\Gamma_{\Sigma}^{\mathcal{G}}$ to be the class of all finite $\Phi \subset cwff_o(\Sigma)$ such that $\Vdash_{\mathcal{G}} \neg\, \Phi{\downarrow_\beta}$ does not hold.*

In a straightforward manner, one can prove the following results (see [7]).

Lemma 2 *Let $\mathcal{G}$ be a sequent calculus such that $\mathcal{G}(Inv^{\neg})$ is admissible. For any finite sets Φ and Δ of sentences, if $\Phi \cup \neg\Delta \notin \Gamma_{\Sigma}^{\mathcal{G}}$, then $\Vdash_{\mathcal{G}} \neg\, \Phi{\downarrow_\beta} \cup \Delta{\downarrow_\beta}$ holds.*

Theorem 3 *Let $\mathcal{G}$ be a sequent calculus. If the rules $\mathcal{G}(Inv^{\neg})$, $\mathcal{G}(\neg)$, $\mathcal{G}(weak)$, $\mathcal{G}(init)$, $\mathcal{G}(\vee_-)$, $\mathcal{G}(\vee_+)$, $\mathcal{G}(\Pi_-^{\mathbf{C}})$ and $\mathcal{G}(\Pi_+^c)$ are admissible in $\mathcal{G}$, then $\Gamma_{\Sigma}^{\mathcal{G}} \in \mathfrak{Acc}_\beta$.*

We can furthermore show the following relationship between saturation and cut (see [7]).

Theorem 4 *Let $\mathcal{G}$ be a sequent calculus.*

1. *If $\mathcal{G}(cut)$ is admissible in $\mathcal{G}$, then $\Gamma_{\Sigma}^{\mathcal{G}}$ is saturated.*
2. *If $\mathcal{G}(\neg)$ and $\mathcal{G}(Inv^{\neg})$ are admissible in $\mathcal{G}$ and $\Gamma_{\Sigma}^{\mathcal{G}}$ is saturated, then $\mathcal{G}(cut)$ is admissible in $\mathcal{G}$.*

Since saturation is equivalent to admissibility of cut, we need weaker conditions than saturation. A natural condition to consider is the existence of saturated extensions.

Definition 5 (Saturated Extension) *Let $* \in$ ⧉ and $\Gamma_\Sigma, \Gamma'_\Sigma \in \mathfrak{Acc}_*$ be abstract consistency classes. We say Γ'_Σ is an **extension** of Γ_Σ if $\Phi \in \Gamma'_\Sigma$ for every sufficiently Σ-pure $\Phi \in \Gamma_\Sigma$. We say Γ'_Σ is a **saturated extension** of Γ_Σ if Γ'_Σ is saturated and an extension of Γ_Σ.*

There exist abstract consistency classes Γ in $\mathfrak{Acc}_{\beta\mathfrak{fb}}$ which have no saturated extension.

Example 6 *Let $a_o, b_o, q_{o\to o} \in \Sigma$ and $\Phi := \{a, b, (qa), \neg(qb)\}$. We construct an abstract consistency class Γ_Σ from Φ by first building the closure Φ' of Φ under relation $\equiv_\beta$ and then taking the power set of Φ'. It is easy to check that this Γ_Σ is in $\mathfrak{Acc}_{\beta\mathfrak{fb}}$. Suppose we have a saturated extension Γ'_Σ of Γ_Σ in $\mathfrak{Acc}_{\beta\mathfrak{fb}}$. Then $\Phi \in \Gamma'_\Sigma$ since Φ is finite (hence sufficiently pure). By saturation, $\Phi * (a \doteq^o b) \in \Gamma'_\Sigma$ or $\Phi * \neg(a \doteq^o b) \in \Gamma'_\Sigma$. In the first case, applying $\nabla_\forall$ with the constant q, $\nabla_\vee$ and ∇_c contradicts $(qa), \neg(qb) \in \Phi$. In the second case, $\nabla_\mathfrak{b}$ and ∇_c contradict $a, b \in \Phi$.*

Existence of any saturated extension of a sound sequent calculus $\mathcal{G}$ implies admissibility of cut. The proof uses the model existence theorem for saturated abstract consistency classes (cf. [5](6.34)). The full proof is in [7].

Theorem 7 *Let $\mathcal{G}$ be a sequent calculus which is sound for $\mathfrak{M}_*$. If $\Gamma_{\Sigma}^{\mathcal{G}}$ has a saturated extension $\Gamma'_\Sigma \in \mathfrak{Acc}_*$, then $\mathcal{G}(cut)$ is admissible in $\mathcal{G}$.*

Sequent Calculus $\mathcal{G}_\beta$. We now study a particular sequent calculus $\mathcal{G}_\beta$ defined by the rules $\mathcal{G}(init)$, $\mathcal{G}(\neg)$, $\mathcal{G}(\vee_-)$, $\mathcal{G}(\vee_+)$, $\mathcal{G}(\Pi^{\mathbf{C}}_-)$ and $\mathcal{G}(\Pi^c_+)$ (cf. Figure 1). It is easy to show that $\mathcal{G}_\beta$ is sound for the eight model classes and in particular for class $\mathfrak{M}_\beta$.

The reader may easily prove the following Lemma.

Lemma 8 *Let $\mathbf{A} \in cwff_o(\Sigma)$ be an atom, $\mathbf{B} \in cwff_\alpha(\Sigma)$, and Δ be a sequent. In $\mathcal{G}_\beta$*

1. *$\Delta * \mathbf{A} \Leftrightarrow \mathbf{A} := \Delta * \neg(\neg(\neg\mathbf{A} \vee \mathbf{A}) \vee \neg(\neg\mathbf{A} \vee \mathbf{A}))$ is derivable in 7 steps and*
2. *$\Delta * \mathbf{B} \doteq^\alpha \mathbf{B} := \Delta * \Pi^\alpha(\lambda P_{\alpha\to o}\!\centerdot \neg(P\mathbf{B}) \vee (P\mathbf{B})$ is derivable in 3 steps.*

The proof of the next Lemma is by induction on derivations and is given in [7].

Lemma 9 *The rules $\mathcal{G}(Inv^\neg)$ and $\mathcal{G}(weak)$ are 0-admissible in $\mathcal{G}_\beta$.*

Theorem 10 *The sequent calculus $\mathcal{G}_\beta$ is complete for the model class $\mathfrak{M}_\beta$ and the rule $\mathcal{G}(cut)$ is admissible.*

Proof: By Theorem 3 and Lemma 9, $\Gamma_\Sigma^{\mathcal{G}_\beta} \in \mathfrak{Acc}_\beta$. Suppose $\Vdash_{\mathcal{G}_\beta} \Delta$ does not hold. Then $\neg\Delta \in \mathfrak{Acc}_\beta$ by Lemma 2. By the model existence theorem for $\mathfrak{Acc}_\beta$ (cf. [6](8.1)) there exists a model for $\neg\Delta$ in $\mathfrak{M}_\beta$. This gives completeness of $\mathcal{G}_\beta$. We can use completeness to conclude cut is admissible in $\mathcal{G}_\beta$. $\square$

Andrews proves admissibility of cut for a sequent calculus similar to $\mathcal{G}_\beta$ in [1]. The proof in [1] contains the essential ingredients for showing completeness.

We will now show that $\mathcal{G}(cut)$ actually becomes k-admissible in $\mathcal{G}_\beta$ if certain formulae are available in the sequent Δ we wish to prove.

4 Cut-Simulation

Cut-Strong Formulae and Sequents. k-cut-strong formulae can be used to effectively simulate cut. Effectively means that the elimination of each application of a cut-rule introduces maximally k additional proof steps, where k is constant.

Definition 11 *Given a formula $\mathbf{A} \in cwff_o(\Sigma)$, and an arbitrary but fixed number $k > 0$. We call formula $\mathbf{A}$ **k-cut-strong** for $\mathcal{G}$ (or simply **cut-strong**) if the cut rule variant*

$$\frac{\Delta * \mathbf{C} \quad \Delta * \neg\mathbf{C}}{\Delta * \neg\mathbf{A}}\,\mathcal{G}(cut^{\mathbf{A}})$$

is k-admissible in $\mathcal{G}$.

Our examples below illustrate that cut-strength of a formula usually only weakly depends on calculus $\mathcal{G}$: it only presumes standard ingredients such as β-normalization, weakening, and rules for the logical connectives.

We present some simple examples of cut-strong formulae for our sequent calculus $\mathcal{G}_\beta$. A corresponding phenomenon is observable in other higher-order calculi, for instance, for the calculi presented in [1, 4, 8, 11].

Example 12 *Formula* $\forall P_o.P := \Pi^o(\lambda P_o.P)$ *is* 3*-cut-strong in* $\mathcal{G}_\beta$*. This is justified by the following derivation which actually shows that rule* $\mathcal{G}(cut^{\mathbf{A}})$ *for this specific choice of* $\mathbf{A}$ *is derivable in* $\mathcal{G}_\beta$ *by maximally* 3 *additional proof steps. The only interesting proof step is the instantiation of* P *with formula* $\mathbf{D} := \neg\mathbf{C} \lor \mathbf{C}$ *in rule* $\mathcal{G}(\Pi^{\mathbf{D}}_-)$*. (Note that* $\mathbf{C}$ *must be* β*-normal; sequents such as* $\Delta * \mathbf{C}$ *by definition contain only* β*-normal formulae.)*

$$\dfrac{\dfrac{\dfrac{\Delta * \mathbf{C}}{\Delta * \neg\neg\mathbf{C}}\;\mathcal{G}(\neg) \qquad \Delta * \neg\mathbf{C}}{\Delta * \neg(\neg\mathbf{C} \lor \mathbf{C})}\;\mathcal{G}(\lor_-)}{\Delta * \neg\Pi^o(\lambda P_o.P)}\;\mathcal{G}(\Pi^{\mathbf{D}}_-)$$

Clearly, $\forall P_o.P$ *is not a very interesting cut-strong formula since it implies falsehood, i.e. inconsistency.*

Example 13 *The formula* $\forall P_o.P \Rightarrow P := \Pi^o(\lambda P_o.\neg P \lor P)$ *is* 3*-cut-strong in* $\mathcal{G}_\beta$*. This is an example of a tautologous cut-strong formula. Now* P *is simply instantiated with* $\mathbf{D} := \mathbf{C}$ *in rule* $\mathcal{G}(\Pi^{\mathbf{D}}_-)$*. Except for this first step the derivation is identical to the one for Example 12.*

Example 14 *Leibniz equations* $\mathbf{M} \doteq^\alpha \mathbf{N} := \Pi^\alpha(\lambda P.\neg P\mathbf{M} \lor P\mathbf{N})$ *(for arbitrary formulae* $\mathbf{M}, \mathbf{N} \in cwff_\alpha(\Sigma)$ *and types* $\alpha \in \mathcal{T}$*) are* 3*-cut-strong in* $\mathcal{G}_\beta$*. This includes the special cases* $\mathbf{M} \doteq^\alpha \mathbf{M}$*. Now* P *is instantiated with* $\mathbf{D} := \lambda X_\alpha.\mathbf{C}$ *in rule* $\mathcal{G}(\Pi^{\mathbf{D}}_-)$*. Except for this first step the derivation is identical to the one for Example 12.*

Example 15 *The original formulation of higher-order logic (cf. [12]) contained comprehension axioms of the form* $\mathcal{C} := \exists P_{\alpha^1 \to \dots \to \alpha^n \to o}\forall \overline{X^n}.P\overline{X^n} \Leftrightarrow \mathbf{B}_o.$ *where* $\mathbf{B}_o \in wff_o(\Sigma)$ *is arbitrary with* $P \notin free(\mathbf{B})$*. Church eliminated the need for such axioms by formulating higher-order logic using typed* λ*-calculus. We will now show that the instance* $\mathcal{C}^I := \exists P_{\iota \to o}.\forall X_\iota.PX \Leftrightarrow X \doteq^\iota X$ *is* 16*-cut-strong in* $\mathcal{G}_\beta$ *(note that* $\mathcal{G}(weak)$ *is* 0*-admissible). This motivates building-in comprehension principles instead of treating comprehension axiomatically.*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\textit{3 steps; see Lemma 8}\\ \vdots \\ \Delta * \neg(pa \Rightarrow a \doteq^\iota a) * a \doteq^\iota a\end{array}}{\Delta * \neg(pa \Rightarrow a \doteq^\iota a) * \neg\neg(a \doteq^\iota a)}\;\mathcal{G}(\neg) \qquad \mathcal{D}}{\Delta * \neg(pa \Rightarrow a \doteq^\iota a) * \neg(\neg(a \doteq^\iota a) \lor pa)}\;\mathcal{G}(\lor_-)}{\Delta * \neg(pa \Rightarrow a \doteq^\iota a) \lor \neg(a \doteq^\iota a \Rightarrow pa)}\;\mathcal{G}(\lor_+)}{\Delta * \neg\neg(\neg(pa \Rightarrow a \doteq^\iota a) \lor \neg(a \doteq^\iota a \Rightarrow pa))}\;\mathcal{G}(\neg)}{\Delta * \neg\Pi^\iota(\lambda X_\iota.pX \Leftrightarrow X \doteq^\iota X)}\;\mathcal{G}(\Pi^{a_\iota}_-)}{\Delta * \Pi^{\iota \to o}(\lambda P^{\iota \to o}.\neg\Pi^\iota(\lambda X_\iota.pX \Leftrightarrow X \doteq^\iota X))}\;\mathcal{G}(\Pi^{p_{\iota \to o}}_+)}{\Delta * \mathcal{C}^I}\;\mathcal{G}(\neg)$$

Derivation $\mathcal{D}$ is:

$$\dfrac{\dfrac{\overline{\Delta * pa * \neg pa}\ \mathcal{G}(init)}{\Delta * \neg\neg pa * \neg pa}\ \mathcal{G}(\neg) \qquad \dfrac{\dfrac{\Delta * \mathbf{C} \quad \Delta * \neg\mathbf{C}}{\Delta * \neg(a \doteq^{o} a)}\ \vdots\ \text{3 steps; see Example 14}}{\Delta * \neg(a \doteq^{\iota} a) * \neg pa}\ \mathcal{G}(weak)}{\Delta * \neg(\neg pa \vee a \doteq^{\iota} a) * \neg pa}\ \mathcal{G}(\vee_{-})$$

As we will show later, many prominent axioms for higher-order logic also belong to the class of cut-strong formulae.

Next we define cut-strong sequents.

Definition 16 *A sequent Δ is called k-cut-strong (or simply* ***cut-strong****) if there exists a a k-cut-strong formula $\mathbf{A} \in cwff_o(\Sigma)$ such that $\neg\mathbf{A} \in \Delta$.*

Cut-Simulation. The cut-simulation theorem is a main result of this paper. It says that cut-strong sequents support an effective simulation (and thus elimination) of cut in $\mathcal{G}_\beta$. Effective means that the size of cut-free derivation grows only linearly for the number of cut rule applications to be eliminated.

We first fix the following calculi: Calculus $\mathcal{G}_\beta^{cut}$ extends $\mathcal{G}_\beta$ by the rule $\mathcal{G}(cut)$ and calculus $\mathcal{G}_\beta^{cut^{\mathbf{A}}}$ extends $\mathcal{G}_\beta$ by the rule $\mathcal{G}(cut^{\mathbf{A}})$ for some arbitrary but fixed cut-strong formula $\mathbf{A}$.

Theorem 17 *Let Δ be a k-cut-strong sequent such that $\neg\mathbf{A} \in \Delta$ for some k-cut-strong formula $\mathbf{A}$. For each derivation $\mathcal{D}$: $\Vdash_{\mathcal{G}_\beta^{cut}} \Delta$ with d proof steps there exists an alternative derivation $\mathcal{D}'$: $\Vdash_{\mathcal{G}_\beta^{cut^{\mathbf{A}}}} \Delta$ with d proof steps.*

Proof: Note that the rules $\mathcal{G}(cut)$ and $\mathcal{G}(cut^{\mathbf{A}})$ coincide whenever $\neg\mathbf{A} \in \Delta$. Intuitively, we can replace each occurrence of $\mathcal{G}(cut)$ in $\mathcal{D}$ by $\mathcal{G}(cut^{\mathbf{A}})$ in order to obtain a $\mathcal{D}'$ of same size. Technically, in the induction proof one must weaken to ensure $\neg\mathbf{A}$ stays in the sequent and carry out a parameter renaming to make sure the eigenvariable condition is satisfied. $\square$

Theorem 18 *Let Δ be a k-cut-strong sequent such that $\neg\mathbf{A} \in \Delta$ for some k-cut-strong formula $\mathbf{A}$. For each derivation $\mathcal{D}$: $\Vdash_{\mathcal{G}_\beta^{cut^{\mathbf{A}}}} \Delta$ with d proof steps and with n applications of rule $\mathcal{G}(cut)$ there exists an alternative derivation $\mathcal{D}'$: $\Vdash_{\mathcal{G}_\beta} \Delta$ with maximally $d + nk$ proof steps.*

Proof: $\mathbf{A}$ is k-cut-strong so by definition $\mathcal{G}(cut^{\mathbf{A}})$ is k-admissible in $\mathcal{G}_\beta$. This means that $\mathcal{G}(cut^{\mathbf{A}})$ can be eliminated in $\mathcal{D}$ and each single elimination of $\mathcal{G}(cut^{\mathbf{A}})$ introduces maximally k new proof steps. Now the assertion can be easily obtained by a simple induction over n. $\square$

Corollary 19 *Let Δ be a k-cut-strong sequent. For each derivation $\mathcal{D}$: $\Vdash_{\mathcal{G}_\beta^{cut}} \Delta$ with d proof steps and n applications of rule $\mathcal{G}(cut)$ there exists an alternative cut-free derivation $\mathcal{D}'$: $\Vdash_{\mathcal{G}_\beta} \Delta$ with maximally $d + nk$ proof steps.*

5 The Extensionality Axioms are Cut-Strong

We have shown comprehension axioms can be cut-strong (cf. Example 15). Further prominent examples of cut-strong formulae are the Boolean and functional extensionality axioms. The Boolean extensionality axiom (abbreviated $\mathcal{B}_o$ in the remainder) is

$$\forall A_o\boldsymbol{.}\forall B_o\boldsymbol{.}(A \Leftrightarrow B) \Rightarrow A \doteq^o B$$

The infinitely many functional extensionality axioms (abbreviated $\mathcal{F}_{\alpha\beta}$) are parameterized over $\alpha, \beta \in \mathcal{T}$.

$$\forall F_{\alpha\to\beta}\boldsymbol{.}\forall G_{\alpha\to\beta}\boldsymbol{.}(\forall X_\alpha\boldsymbol{.}FX \doteq^\beta GX) \Rightarrow F \doteq^{\alpha\to\beta} G$$

These axioms usually have to be added to higher-order calculi to reach Henkin completeness, i.e. completeness with respect to model class $\mathfrak{M}_{\beta\mathfrak{fb}}$. For example, Huet's constrained resolution approach as presented in [11] is not Henkin complete without adding extensionality axioms. For instance, the need for adding Boolean extensionality is actually illustrated by the set of unit literals $\Phi := \{a, b, (qa), \neg(qb)\}$ from Example 6. As the reader may easily check, this clause set Φ, which is inconsistent for Henkin semantics, cannot be proven by Huet's system without, e.g, adding the Boolean extensionality axiom. By relying on results in [1], Huet essentially shows completeness with respect to model class $\mathfrak{M}_\beta$ as opposed to Henkin semantics.

We will now investigate whether adding the extensionality axioms to a machine-oriented calculus in order to obtain Henkin completeness is a suitable option.

Theorem 20 *The Boolean extensionality axiom $\mathcal{B}_o$ is a* 14*-cut-strong formula in $\mathcal{G}_\beta$.*

Proof: The following derivation justifies this theorem (a_o is a parameter).

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\text{7 steps; see Lemma 8}\\ \vdots\\ \Delta * a \Leftrightarrow a\end{array}}{\Delta * \neg\neg(a \Leftrightarrow a)}\,\mathcal{G}(\neg) \qquad \begin{array}{c}\Delta * \mathbf{C} \quad \Delta * \neg\mathbf{C}\\ \vdots\ \text{3 steps; see Example 14}\\ \Delta * \neg(a \doteq^o a)\end{array}}{\Delta * \neg(\neg(a \Leftrightarrow a) \vee a \doteq^o a)}\,\mathcal{G}(\vee_-)}{\Delta * \neg\mathcal{B}_o}\,2 \times \mathcal{G}(\Pi_-^a)$$

$\square$

Theorem 21 *The functional extensionality axioms $\mathcal{F}_{\alpha\beta}$ are* 11*-cut-strong formulae in $\mathcal{G}_\beta$.*

Proof: The following derivation justifies this theorem ($f_{\alpha\to\beta}$ is a parameter).

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\text{3 steps; see Lemma 8}\\ \vdots\\ \Delta * fa \doteq^\beta fa\end{array}}{\Delta * (\forall X_\alpha\boldsymbol{.}fX \doteq^\beta fX)}\,\mathcal{G}(\Pi_+^{a_\alpha})}{\Delta * \neg\neg\forall X_\alpha\boldsymbol{.}fX \doteq^\beta fX}\,\mathcal{G}(\neg) \qquad \begin{array}{c}\Delta * \mathbf{C} \quad \Delta * \neg\mathbf{C}\\ \vdots\ \text{3 steps; see Example 14}\\ \Delta * \neg(f \doteq^{\alpha\to\beta} f)\end{array}}{\Delta * \neg(\neg(\forall X_\alpha\boldsymbol{.}fX \doteq^\beta fX) \vee f \doteq^{\alpha\to\beta} f)}\,\mathcal{G}(\vee_-)}{\Delta * \neg\mathcal{F}_{\alpha\beta}}\,2 \times \mathcal{G}(\Pi_-^f)$$

$\square$

$$\frac{\Delta * \neg\mathcal{F}_{\alpha\beta} \quad \alpha \to \beta \in \mathcal{T}}{\Delta}\,\mathcal{G}(\mathcal{F}_{\alpha\beta}) \qquad \frac{\Delta * \neg\mathcal{B}_o}{\Delta}\,\mathcal{G}(\mathcal{B})$$

Fig. 2. Axiomatic Extensionality Rules

In [4] and [8] we have already argued that the extensionality principles should not be treated axiomatically in machine-oriented higher-order calculi and there we have developed resolution and sequent calculi in which these principles are built-in. Here we have now developed a strong theoretical justification for this work: Theorems 20, 21 and 19 tell us that adding the extensionality principles $\mathcal{B}_o$ and $\mathcal{F}_{\alpha\beta}$ as axioms to a calculus is like adding a cut rule.

In Figure 2 we show rules that add Boolean and functional extensionality in an axiomatic manner to $\mathcal{G}_\beta$. More precisely we add rules $\mathcal{G}(\mathcal{F}_{\alpha\beta})$ and $\mathcal{G}(\mathcal{B})$ allowing to introduce the axioms for any sequent Δ; this way we address the problem of the infinitely many possible instantiations of the type-schematic functional extensional axiom $\mathcal{F}_{\alpha\beta}$. Calculus $\mathcal{G}_\beta$ enriched by the new rules $\mathcal{G}(\mathcal{F}_{\alpha\beta})$ and $\mathcal{G}(\mathcal{B})$ is called $\mathcal{G}^E_\beta$. Soundness of the the new rules is easy to verify: In [5](4.3) we show that $\mathcal{G}(\mathcal{F}_{\alpha\beta})$ and $\mathcal{G}(\mathcal{B})$ are valid for Henkin models.

Replacing the Extensionality Axioms. In Figure 3 we define alternative extensionality rules which correspond to those developed for resolution and sequent calculi in [4] and [8]. Calculus $\mathcal{G}_\beta$ enriched by $\mathcal{G}(\mathfrak{f})$ and $\mathcal{G}(\mathfrak{b})$ is called $\mathcal{G}^-_{\beta\mathfrak{fb}}$. Soundness of $\mathcal{G}(\mathfrak{f})$ and $\mathcal{G}(\mathfrak{b})$ for Henkin semantics is again easy to show.

Our aim is to develop a machine-oriented sequent calculus for automating Henkin complete proof search. We argue that for this purpose $\mathcal{G}(\mathfrak{f})$ and $\mathcal{G}(\mathfrak{b})$ are more suitable rules than $\mathcal{G}(\mathcal{F}_{\alpha\beta})$ and $\mathcal{G}(\mathcal{B})$.

Our next step now is to show Henkin completeness for $\mathcal{G}^E_\beta$. This will be relatively easy since we can employ cut-simulation. Then we analyze whether calculus $\mathcal{G}^-_{\beta\mathfrak{fb}}$ has the same deductive power as $\mathcal{G}^E_\beta$.

First we extend Theorem 3. The proof is given in [7].

Theorem 22 *Let $\mathcal{G}$ be a sequent calculus such that $\mathcal{G}(Inv^\neg)$ and $\mathcal{G}(\neg)$ are admissible.*

1. *If $\mathcal{G}(\mathfrak{f})$ and $\mathcal{G}(\Pi^c_+)$ are admissible, then $\Gamma^{\mathcal{G}}_\Sigma$ satisfies $\nabla_\mathfrak{f}$.*
2. *If $\mathcal{G}(\mathfrak{b})$ is admissible, then $\Gamma^{\mathcal{G}}_\Sigma$ satisfies $\nabla_\mathfrak{b}$.*

Theorem 23 *The sequent calculus $\mathcal{G}^E_\beta$ is Henkin complete and the rule $\mathcal{G}(cut)$ is* 12*-admissible.*

Proof: $\mathcal{G}(cut)$ can be effectively simulated and hence eliminated in $\mathcal{G}^E_\beta$ by combining rule $\mathcal{G}(\mathcal{F}_{\alpha\beta})$ with the 11-step derivation presented in the proof of Theorem 21.

Let $\Gamma^{\mathcal{G}^E_\beta}_\Sigma$ be defined as in Definition 1. We prove Henkin completeness of $\mathcal{G}^E_\beta$ by showing that the class $\Gamma^{\mathcal{G}^E_\beta}_\Sigma$ is a saturated abstract consistency class in

$$\frac{\Delta * (\forall X_\alpha \mathbf{.} \mathbf{A}X \doteq^\beta \mathbf{B}X){\downarrow_\beta}}{\Delta * (\mathbf{A} \doteq^{\alpha\to\beta} \mathbf{B})}\,\mathcal{G}(\mathfrak{f}) \qquad \frac{\Delta * \neg\mathbf{A} * \mathbf{B} \quad \Delta * \neg\mathbf{B} * \mathbf{A}}{\Delta * (\mathbf{A} \doteq^o \mathbf{B})}\,\mathcal{G}(\mathfrak{b})$$

Fig. 3. Proper Extensionality Rules

$\mathfrak{Acc}_{\beta\mathfrak{fb}}$. We here only analyze the crucial conditions $\nabla_\mathfrak{b}$, $\nabla_\mathfrak{f}$ and ∇_{sat}. For the other conditions we refer to Theorem 3. Note that 0-admissibility of $\mathcal{G}(Inv^\neg)$ and $\mathcal{G}(weak)$ can be shown for $\mathcal{G}^E_\beta$ by a suitable induction on derivations as in Lemma 9.

$\nabla_\mathfrak{f}$ $\mathcal{G}(\Pi^c_+)$ is a rule of $\mathcal{G}^E_\beta$ and thus admissible. According to Theorem 22 it is thus sufficient to ensure admissibility of rule $\mathcal{G}(\mathfrak{f})$ to show $\nabla_\mathfrak{f}$. This is justified by the following derivation where $\mathbf{N} := \mathbf{A} \doteq^{\alpha\to\beta} \mathbf{B}$ and $\mathbf{M} := (\forall X_\alpha \mathbf{.} \mathbf{A}X \doteq^\beta \mathbf{B}X){\downarrow_\beta}$ (for β-normal $\mathbf{A}, \mathbf{B}$).

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\Delta * (\forall X_\alpha \mathbf{.} \mathbf{A}X \doteq^\beta \mathbf{B}X){\downarrow_\beta}}{\Delta * \mathbf{N} * \mathbf{M}}\,\mathcal{G}(weak)}{\Delta * \mathbf{N} * \neg\neg\mathbf{M}}\,\mathcal{G}(\neg) \qquad \overset{\text{derivable}}{\vdots}\atop \Delta * \mathbf{N} * \neg\mathbf{N}}{\Delta * \mathbf{N} * \neg(\neg\mathbf{M} \vee \mathbf{N})}\,\mathcal{G}(\vee_-)}{\Delta * \mathbf{N} * \neg\mathcal{F}_{\alpha\beta}}\,\mathcal{G}(\Pi^\mathbf{A}_-), \mathcal{G}(\Pi^\mathbf{B}_-)}{\Delta * \mathbf{A} \doteq^{\alpha\to\beta} \mathbf{B}}\,\mathcal{G}(\mathcal{F}_{\alpha\beta})}{}$$

$\nabla_\mathfrak{b}$ With a similar derivation using $\mathcal{G}(\mathcal{B})$ we can show that $\mathcal{G}(\mathfrak{b})$ is admissible. We conclude $\nabla_\mathfrak{b}$ by Theorem 22.

∇_{sat} Since $\mathcal{G}(cut)$ is admissible we get saturation by Theorem 4. □

Does $\mathcal{G}^-_{\beta\mathfrak{fb}}$ have the same deductive strength as $\mathcal{G}^E_\beta$? I.e., is $\mathcal{G}^-_{\beta\mathfrak{fb}}$ Henkin complete? We show this is not yet the case.

Theorem 24 *The sequent calculus $\mathcal{G}^-_{\beta\mathfrak{fb}}$ is not complete for Henkin semantics.*

We illustrate the problem by a counterexample.

Example 25 *Consider the sequent $\Delta := \{\neg a, \neg b, \neg(qa), (qb)\}$ where $a_o, b_o, q_{o\to o} \in \Sigma$ are parameters. For any $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon) \in \mathfrak{M}_{\beta\mathfrak{fb}}$, either $\upsilon(\mathcal{E}(a)) \equiv \mathtt{F}$, $\upsilon(\mathcal{E}(b)) \equiv \mathtt{F}$ or $\mathcal{E}(a) \equiv \mathcal{E}(b)$ by property $\mathfrak{b}$. Hence sequent Δ is valid for every $\mathcal{M} \in \mathfrak{M}_{\beta\mathfrak{fb}}$. However, $\Vdash_{\mathcal{G}^-_{\beta\mathfrak{fb}}} \Delta$ does not hold. By inspection, Δ cannot be the conclusion of any rule.*

In order to reach Henkin completeness and to show cut-elimination we thus need to add further rules. Our example motivates the two rules presented in Figure 4. $\mathcal{G}(Init^{\doteq})$ introduces Leibniz equations such as $qa \doteq^o qb$ as is needed in our example and $\mathcal{G}(d)$ realizes the required decomposition into $a \doteq^o b$.

$$\frac{\Delta * (\mathbf{A} \doteq^{o} \mathbf{B}) \quad (\dagger)}{\Delta * \neg\mathbf{A} * \mathbf{B}}\,\mathcal{G}(\mathit{Init}^{\doteq}) \qquad \frac{\Delta * (\mathbf{A}^1 \doteq^{\alpha_1} \mathbf{B}^1) \ \cdots \ \Delta * (\mathbf{A}^n \doteq^{\alpha_n} \mathbf{B}^n) \quad (\ddagger)}{\Delta * (h\overline{\mathbf{A}^n} \doteq^{\beta} h\overline{\mathbf{B}^n})}\,\mathcal{G}(d)$$

($\dagger$) $\mathbf{A}$,$\mathbf{B}$ atomic ($\ddagger$) $n \geq 1, \beta \in \{o, \iota\}, h_{\overline{\alpha^n} \to \beta} \in \Sigma$ parameter

Fig. 4. Additional Rules $\mathcal{G}(\mathit{Init}^{\doteq})$ and $\mathcal{G}(d)$

We thus extend sequent calculus $\mathcal{G}^{-}_{\beta\mathfrak{f}\mathfrak{b}}$ to $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ by adding the decomposition rule $\mathcal{G}(d)$ and the rule $\mathcal{G}(\mathit{Init}^{\doteq})$ which generally checks if two atomic sentences of opposite polarity are provably equal (as opposed to syntactically equal).

Is $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ complete for Henkin semantics? We will show in the next Section that this indeed holds (cf. Theorem 28).

With $\mathcal{G}^{E}$ and $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ we have thus developed two Henkin complete calculi and both calculi are cut-free. However, as our exploration shows "cut-freeness" is not a well-chosen criterion to differentiate between their suitability for proof search automation: $\mathcal{G}^{E}$ inherently supports effective cut-simulation and thus cut-freeness is meaningless.

The criterion we propose for the analysis of calculi in impredicative logics is "freeness of effective cut-simulation".

Other Rules for Other Model Classes. In [6] we developed respective complete and cut-free sequent calculi not only for Henkin semantics but for five of the eight model classes. In particular, no additional rules are required for the β, $\beta\eta$ and $\beta\xi$ case. Meanwhile, the $\beta\mathfrak{f}$ case requires additional rules allowing η-conversion. The limited space does not allow us to present and analyze these cases here.

6 Acceptability Conditions

We now turn our attention again to the existence of saturated extension of abstract consistency classes.

As illustrated by the Example 6, we need some extra abstract consistency properties to ensure the existence of saturated extensions. We call these extra properties **acceptability conditions**. They actually closely correspond to additional rules $\mathcal{G}(\mathit{Init}^{\doteq})$ and $\mathcal{G}(d)$.

Definition 26 (Acceptability Conditions) *Let Γ_{Σ} be an abstract consistency class in $\mathfrak{Acc}_{\beta\mathfrak{f}\mathfrak{b}}$. We define the following properties:*

∇_m *If $\mathbf{A}, \mathbf{B} \in \mathit{cwff}_o(\Sigma)$ are atomic and $\mathbf{A}, \neg\mathbf{B} \in \Phi$, then $\Phi * \neg(\mathbf{A} \doteq^{o} \mathbf{B}) \in \Gamma_{\Sigma}$.*

∇_d *If $\neg(h\overline{\mathbf{A}^n} \doteq^{\beta} h\overline{\mathbf{B}^n}) \in \Phi$ for some types α_i where $\beta \in \{o, \iota\}$ and $h_{\overline{\alpha^n} \to \beta} \in \Sigma$ is a parameter, then there is an i $(1 \leq i \leq n)$ such that $\Phi * \neg(\mathbf{A}^i \doteq^{\alpha^i} \mathbf{B}^i) \in \Gamma_{\Sigma}$.*

We now replace the strong saturation condition used in [5] by these acceptability conditions.

Definition 27 (Acceptable Classes) *An abstract consistency class $\Gamma_\Sigma \in \mathfrak{Acc}_{\beta\mathfrak{fb}}$ is called **acceptable** in $\mathfrak{Acc}_{\beta\mathfrak{fb}}$ if it satisfies the conditions ∇_m and ∇_d.*

One can show a model existence theorem for acceptable abstract consistency classes in $\mathfrak{Acc}_{\beta\mathfrak{fb}}$ (cf. [6](8.1)). From this model existence theorem, one can conclude $\mathcal{G}_{\beta\mathfrak{fb}}$ is complete for $\mathfrak{M}_{\beta\mathfrak{fb}}$ (hence for Henkin models) and that cut is admissible in $\mathcal{G}_{\beta\mathfrak{fb}}$.

Theorem 28 *The sequent calculus $\mathcal{G}_{\beta\mathfrak{fb}}$ is complete for Henkin semantics and the rule $\mathcal{G}(cut)$ is admissible.*

Proof: The argumentation is similar to Theorem 10 but here we employ the acceptability conditions ∇_m and ∇_d. □

One can further show the **Saturated Extension Theorem** (cf. [6](9.3)):

Theorem 29 *There is a saturated abstract consistency class in $\mathfrak{Acc}_{\beta\mathfrak{fb}}$ that is an extension of all acceptable Γ_Σ in $\mathfrak{Acc}_{\beta\mathfrak{fb}}$.*

Given Theorem 7, one can view the Saturated Extension Theorem as an abstract cut-elimination result.

The proof of a model existence theorem employs Hintikka sets and in the context of studying Hintikka sets we have identified a phenomenon related to cut-strength which we call the **Impredicativity Gap**. That is, a Hintikka set $\mathcal{H}$ is saturated if any cut-strong formula $\mathbf{A}$ (e.g. a Leibniz equation $\mathbf{C} \doteq \mathbf{D}$) is in $\mathcal{H}$. Hence we can reasonably say there is a "gap" between saturated and unsaturated Hintikka sets. Every Hintikka set is either saturated or contains no cut-strong formulae.

7 Conclusion

We have shown that adding cut-strong formulae to a calculus for an impredicative logic is like adding cut. For machine-oriented automated theorem proving in impredicative logics — such as classical type theory — it is therefore not recommendable to naively add cut-strong axioms to the search space. In addition to the comprehension principle and the functional and Boolean extensionality axioms as elaborated in this paper the list of cut-strong axioms includes:

Other Forms of Defined Equality Formulas $\mathbf{A} \ddot{=}^{\alpha} \mathbf{B}$ are 4-cut-strong in $\mathcal{G}_\beta$ where $\ddot{=}^{\alpha}$ is $\lambda X_\alpha . \lambda Y_\alpha . \forall Q_{\alpha\to\alpha\to o} . (\forall Z_\alpha . (Q\ Z\ Z)) \Rightarrow (Q\ X\ Y)$ (cf. [3]).
Proof: Instantiate Q with $\lambda X_\alpha . \lambda Y_\alpha . \mathbf{C}$. □

Axiom of Induction The axiom of induction for the naturals $\forall P_{\iota\to o} . P0 \wedge (\forall X_\iota . PX \Rightarrow P(sX)) \Rightarrow \forall X_\iota . PX$ is 18-cut-strong in $\mathcal{G}_\beta$. (Other well-founded ordering axioms are analogous.)
Proof: Instantiate P with $\lambda X_\iota . a \doteq^{o} a$ for some parameter a_o. □

Axiom of Choice $\exists I_{(\alpha\to o)\to o}.\forall Q_{\alpha\to o}.\exists X_\alpha.QX \Rightarrow Q(IQ)$ is 7-cut-strong in $\mathcal{G}_\beta$.

Proof: Instantiate Q with $\lambda X_\alpha.\mathbf{C}$. $\square$

Axiom of Description The description axiom $\exists I_{(\alpha\to o)\to o}.\forall Q_{\alpha\to o}.(\exists_1 Y_\alpha.QY) \Rightarrow Q(IQ)$ (see [2]), where $\exists_1 Y_\alpha.QY$ stands for $\exists Y_\alpha.QY \wedge (\forall Z_\alpha.QZ \Rightarrow Y \doteq Z)$ is 25-cut-strong in $\mathcal{G}_\beta$.

Proof: Instantiate Q with $\lambda X_\alpha.a \doteq^\alpha X$ for some parameter a_α. $\square$

As Example 15 shows, comprehension axioms can be cut-strong. Church's formulation of type theory (cf. [9]) used typed λ-calculus to build comprehension principles into the language. One can view Church's formulation as a first step in the program to eliminate the need for cut-strong axioms. For the extensionality axioms a start has been made by the sequent calculi in this paper (and [6]), for resolution in [4] and for sequent calculi and extensional expansion proofs in [8]. The extensional systems in [8] also provide a complete method for using primitive equality instead of Leibniz equality. For improving the automation of higher-order logic our exploration thus motivates the development of higher-order calculi which directly include reasoning principles for equality, extensionality, induction, choice, description, etc., without using cut-strong axioms.

References

1. P. B. Andrews. Resolution in type theory. *Journal of Symbolic Logic*, 36(3):414–432, 1971.
2. P. B. Andrews. General models and extensionality. *Journal of Symbolic Logic*, 37(2):395–397, 1972.
3. P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.* Kluwer Academic Publishers, second edition, 2002.
4. C. E. Benzmüller. *Equality and Extensionality in Automated Higher-Order Theorem Proving.* PhD thesis, Saarland University, 1999.
5. C. E. Benzmüller, C. E. Brown, and M. Kohlhase. Higher-order semantics and extensionality. *Journal of Symbolic Logic*, 69(4):1027–1088, 2004.
6. C. E. Benzmüller, C. E. Brown, and M. Kohlhase. Semantic techniques for higher-order cut-elimination. Seki Report SR-2004-07, Saarland University, 2004.
7. C. E. Benzmüller, C. E. Brown, and M. Kohlhase. Cut-simulation in impredicative logics (extended version). Seki Report SR-2006-01, Saarland University, 2006.
8. C. E. Brown. *Set Comprehension in Church's Type Theory.* PhD thesis, Department of Mathematical Sciences, Carnegie Mellon University, 2004.
9. A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.
10. K. J. J. Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:7–55, 1955.
11. G. P. Huet. A mechanization of type theory. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, pages 139–146, 1973.
12. B. Russell. Mathematical logic as based on the theory of types. *American Journal of Mathematics*, 30:222–262, 1908.
13. R. M. Smullyan. A unifying principle for quantification theory. *Proc. Nat. Acad Sciences*, 49:828–832, 1963.
14. R. M. Smullyan. *First-Order Logic.* Springer, 1968.