J Autom Reasoning (2011) 47:451–479
DOI 10.1007/s10817-011-9233-2

Analytic Tableaux for Higher-Order Logic with Choice

Julian Backes · Chad Edward Brown

Received: 9 June 2011 / Accepted: 15 June 2011 / Published online: 9 July 2011

Abstract While many higher-order interactive theorem provers include a choice operator, higher-order automated theorem provers so far have not. In order to support automated reasoning in the presence of a choice operator, we present a cut-free ground tableau calculus for Church's simple type theory with choice. The tableau calculus is designed with automated search in mind. In particular, the rules only operate on the top level structure of formulas. Additionally, we restrict the instantiation terms for quantifiers to a universe that depends on the current branch. At base types the universe of instantiations is finite. Both of these restrictions are intended to minimize the number of rules a corresponding search procedure is obligated to consider. We prove completeness of the tableau calculus relative to Henkin models.

Keywords Higher-order logic · Simple type theory · Tableaux · Completeness · Axiom of choice · Choice operators · Henkin models

1 Introduction

Interactive theorem provers based on classical higher-order logic (e.g., Isabelle-HOL [26], HOL88 [17], HOL-light [18], ProofPower [22] and HOL4 [28]) build in the axiom of choice by including a form of Hilbert's ε binder and appropriate

J. Backes · C. E. Brown (✉)
Saarland University, Saarbrücken, Germany
e-mail: cebrown@ps.uni-saarland.de

J. Backes
e-mail: julianbackes@gmail.com

rules. Church's formulation of the simple theory of types [15] included a selection operator (called ι) and an axiom of choice for this operator at each type. Henkin defined a general notion of a model of Church's type theory with choice and proved completeness [19]. A higher-order version of the TPTP has been under development the past few years [32]. In 2009 it was decided that Henkin models with choice would be the default semantics of the higher-order TPTP.

Automated theorem provers for classical higher-order logic (e.g., TPS [4] and LEO-II [10]) do not currently build in the axiom of choice. Completeness of such calculi is judged with respect to a variant of Henkin's models without choice [3, 9]. As Miller argues [23] Skolemization is unsound with respect to Henkin models without choice but is incomplete with respect to Henkin models with choice. For example, Skolemization makes the formula $(\forall x \exists y.rxy) \to \exists f \forall x.rx(fx)$ easy to prove, but Skolemization does not help one prove $\exists c \forall px.px \to p(cp)$. Miller gives a restriction which makes Skolemization sound with respect to Henkin models without choice.

What would be involved in adding support for choice? Assume a new logical constant ε_σ of type $(\sigma \to o) \to \sigma$ at each type σ is added to the syntax. We need new rules corresponding to this constant. The fundamental property ε_σ should satisfy is expressed by the formula

$$\forall p_{\sigma\to o} x_\sigma.px \to p(\varepsilon_\sigma p) \tag{1}$$

One option is to take a formula s we wish to prove and instead prove (1)$\to s$ using a cut-free proof calculus for higher-order logic without choice (e.g., the calculi in [11] and [14]). The first problem with this option is that (1) only expresses the axiom of choice at a single type σ. We could overcome this in principle by systematically considering $(1_{\sigma_1}) \wedge \cdots \wedge (1_{\sigma_n}) \to s$. for any finite set $\{\sigma_1, \ldots, \sigma_n\}$ of types. The second problem with this option is more serious. Even adding a single instance of (1) at any type σ allows one to simulate cut in the calculus (see Example 7.3 of [8]). This naïve idea for a cut-free calculus is doomed. As argued in [8] it is a general phenomenon that higher-order hypotheses destroy cut-freeness of a calculus. This phenomenon motivates trying to build the assumptions into the calculus in a cut-free (but complete) way.

Our purpose in this paper is to give a complete analytic tableau calculus for higher-order logic with choice that forms a basis for automated reasoning in the logic. Mints [24] has given a sequent calculus for relational higher-order logic with an ε-operator and proves cut-elimination. Mints' calculus does not include arbitrary function types and the corresponding simply typed λ-terms. We adapt Mints' rules for a simply typed formulation in the style of Church. We obtain tighter restrictions on when Mints' main choice rule (the ε-rule) needs to be applied. Furthermore, we show we can omit Mints' ε-extensionality rule altogether. These results are important for automated reasoning because these two rules would be highly branching in practice. In addition to including cut-free rules for the ε-operator, we give strong restrictions on the instantiation of universal quantifiers over base types analogous to those reported in [12].

In Section 2 we give a quick presentation of the syntax and semantics of simple type theory with choice. In Section 3 we present the tableau calculus. In Section 4 we define the notion of an evident set and prove that every evident set has a Henkin model. We define a notion of abstract consistency in Section 5. In Section 6 we prove completeness as well as compactness and the existence of countable models.

In Section 7 we extend the calculus to include an if-then-else operator. We discuss related work and conclude in Sections 8 and 9.

This article is an expanded version of [6].

2 Preliminaries

We start by giving the syntax for simple type theory with a choice operator in the style of Church [15]. Types (σ, τ, μ) are given inductively by the base type o (of truth values), ι (of individuals) and $\sigma \to \tau$ (of functions from σ to τ). For brevity, we will omit the arrow and write $\sigma\tau$ for $\sigma \to \tau$. Omitted parenthesis in types associate to the right: $\sigma\tau\mu$ means $\sigma(\tau\mu)$. The results in the paper generalize to the case where there are arbitrarily many base types of individuals. We use β to range over the base types o and ι.

For each type σ we assume a countably infinite set $\mathcal{V}_\sigma$ of variables of type σ. For each type σ we have logical constants $=_\sigma$ of type $\sigma\sigma o$, $\forall_\sigma$ of type $(\sigma o)o$ and ε_σ (the choice operator) of type $(\sigma o)\sigma$. Furthermore, we have logical constants for disjunction $\vee$ of type ooo, negation $\neg$ of type oo, false $\bot$ of type o and for a default individual $*$ of type ι. (The default individual is included only to act as an instantiation when no other instantiation of type ι is allowed by our calculus.) We use x, y to range over variables and c to range over logical constants. A *name* is either a variable or a logical constant. We use ν to range over names. Variables x and choice operators ε_σ are called *decomposable* names. We use w to range over decomposable names. Let $\mathcal{W}_\sigma$ be the set of decomposable names w of type σ.

The family of sets Λ_σ of terms of type σ are inductively defined. If ν is a name of type σ, then $\nu \in \Lambda_\sigma$. If $t \in \Lambda_{\sigma\tau}$ and $s \in \Lambda_\sigma$, then we have an application term $ts \in \Lambda_\tau$. If $x \in \mathcal{V}_\sigma$ and $t \in \Lambda_\tau$, then we have an abstraction term $\lambda x.s \in \Lambda_{\sigma\tau}$. A *formula* is a term $s \in \Lambda_o$.

Application associates to the left, so that stu means $(st)u$, with the exception that $\neg tu$ always means $\neg(tu)$. We use $\top$ as notation for $\neg\bot$. We use infix notation and write $s =_\sigma t$ (or $s = t$) for $=_\sigma st$ and write $s \vee t$ for $\vee st$. (Note that if s and t are different terms of type σ, then $s = t$ is a different term than $t = s$.) We write $s \neq_\sigma t$ (or $s \neq t$) for $\neg(s =_\sigma t)$. We also use binder notation to write $\forall x.s$ for $\forall_\sigma \lambda x.s$ and write $\varepsilon x.s$ for $\varepsilon_\sigma \lambda x.s$. We write $s \to t$, $s \wedge t$ and $\exists x.s$ as shorthands for $\neg s \vee t$, $\neg(\neg s \vee \neg t)$ and $\neg\forall x.\neg s$, respectively.

The set $\mathcal{V}t$ of *free variables of t* is defined as usual. For a set of variables X we write Λ_σ^X for the set of all terms $t \in \Lambda_\sigma$ such that $\mathcal{V}t \subseteq X$. Also, for a set A of terms, $\mathcal{V}A$ is $\bigcup\{\mathcal{V}s | s \in A\}$.

To describe our tableau calculus and to reason about it we will need to be able to refer to certain shallow occurrences of subterms within terms. For example, the choice rule may be applicable in the presence of a formula $\varepsilon_\iota px \neq_\iota y$ (where $p \in \mathcal{V}_{(\iota\iota)o}$ and $x, y \in \mathcal{V}_\iota$) because the subterm $\varepsilon_\iota p$ occurs as a subterm in a special position. To describe such positions, we define two notions of contexts (terms with holes).

An *elimination context* ($\mathcal{E}$) is a term with a hole $[]_\sigma$ defined inductively as follows (see [25]). $[]_\sigma$ is an elimination context of type σ. If $\mathcal{E}$ is an elimination context of type $\tau\mu$ and $s \in \Lambda_\tau$ then $\mathcal{E}s$ is an elimination context of type μ.

Let $\mathcal{E}$ be an elimination context of type σ which has a hole of type τ. We can apply $\mathcal{E}$ to a term $t \in \Lambda_\tau$ to get a term of type σ: $[\,][t] = t$ and $(\mathcal{E}\ s)[t] = \mathcal{E}[t]\ s$.

An *accessibility context* ($\mathscr{C}$) is a term with a hole $[]_\sigma$ of the form $\mathscr{E}$, $\neg\mathscr{E}$, $\mathscr{E} \neq_\iota s$ or $s \neq_\iota \mathscr{E}$ where $\mathscr{E}$ is an elimination context. We can apply an accessibility context $\mathscr{C}$ with a hole of type σ to a term $t \in \Lambda_\sigma$ to get a term of type o in the obvious way. A term s is *accessible* in a set A of formulas if there is an accessibility context $\mathscr{C}$ such that $\mathscr{C}[s] \in A$.

Let A be a set of formulas. A term s is *discriminating* in A if there is a term t such that $s \neq_\iota t \in A$ or $t \neq_\iota s \in A$. A *discriminant* Δ of A is a maximal set of discriminating terms such that there are no $s, t \in \Delta$ with $s \neq_\iota t \in A$. (Discriminants first appeared in [13].) Discriminating terms will be used to instantiate quantifiers over type ι, and discriminants will be used to interpret the type ι. Note that if there are no discriminating terms in A, then $\emptyset$ is the unique discriminant of A. Note also that s is accessible in A if and only if there is an elimination context $\mathscr{E}$ such that $\mathscr{E}[s] \in A$, $\neg\mathscr{E}[s] \in A$ or $\mathscr{E}[s]$ is discriminating in A.

We prove that compatible sets of discriminating terms can be collected into a common discriminant. This fact will be used more than once. This is the first of several places in the paper where we use the axiom of choice at the meta-level. In this particular case, we could use an enumeration of terms to avoid using the axiom of choice since we assumed the set $\mathscr{V}$ of variables is countably infinite.

Proposition 1 *Let A be a set of formulas and C be a set of terms that are discriminating in A. Suppose $s \neq t \notin A$ for every $s, t \in C$. There is some discriminant Δ of A such that $C \subseteq \Delta$.*

Proof Let P be the set of all sets D such that

1. $C \subseteq D$,
2. every term in D is discriminating in A, and
3. $s \neq t \notin A$ for every $s, t \in D$.

Note that P is partially ordered by $\subseteq$. For any totally ordered subset $Q \subseteq P$, $C \cup \left(\bigcup Q\right)$ is an upper bound of Q in P. By Zorn's Lemma, there is some maximal $\Delta \in P$. This Δ is a discriminant of A such that $C \subseteq \Delta$. □

We consider a simple example application of Proposition 1. Suppose $x_1, x_2, \ldots$ and $y_1, y_2, \ldots$ are enumerations of distinct variables of type ι. Let $A = \{x_1 \neq y_1, x_2 \neq y_2, \ldots\}$. Let C be a subset of $\{x_1, x_2, \ldots\} \cup \{y_1, y_2, \ldots\}$ such that for each i either $x_i \notin C$ or $y_i \notin C$. By Proposition 1 there is a discriminant Δ of A extending C.

Proposition 2 *For every set A of formulas, there is a discriminant Δ of A.*

Proof We obtain Δ by applying Proposition 1 with $C = \emptyset$. □

We now turn to a brief description of the semantics. Our notion of an interpretation is essentially that given by Henkin [19]. A *frame* $\mathscr{D}$ is a typed family of nonempty sets such that $\mathscr{D}_o = \{0, 1\}$ and $\mathscr{D}_{\sigma\tau}$ is a set of total functions from $\mathscr{D}_\sigma$ to $\mathscr{D}_\tau$. $\mathscr{D}_o$ is the set of booleans 0 (*false*) and 1 (*true*). An *assignment* into a frame $\mathscr{D}$ is a function $\mathscr{I}$ that maps every name ν of type σ to an element of $\mathscr{D}_\sigma$. We use $\mathscr{I}^x_a$ to denote the assignment that is like $\mathscr{I}$ but maps the variable x to a.

For each logical constant c of type σ we define a corresponding property $\mathfrak{L}_c(a)$ of elements $a \in \mathscr{D}_\sigma$ in Table 1. Essentially $\mathfrak{L}_c(a)$ holds if and only if a is an appropriate interpretation of c. An assignment $\mathscr{I}$ into $\mathscr{D}$ is *logical* if $\mathfrak{L}_c(\mathscr{I}c)$ holds for each logical constant c. A logical assignment $\mathscr{I}$ must map $\bot$ to 0, $\neg$ to the negation function, and so on. There is no restriction on the value of $\mathscr{I}*$ in $\mathscr{D}_\iota$. The most interesting case to consider is the choice operator ε_σ. For an assignment to be logical, $\mathscr{I}\varepsilon_\sigma$ must be a function Φ in $\mathscr{D}_{(\sigma o)\sigma}$ such that $f(\Phi(f)) = 1$ for every $f \in \mathscr{D}_{\sigma o}$ except when f is the constant 0 function. We call such a Φ a *choice function*. There may be many different choice functions in $\mathscr{D}_{(\sigma o)\sigma}$. (Of course, there may also be no choice function.)

We now turn to the interpretation of all typed terms. To do this we use induction on terms to lift each assignment $\mathscr{I}$ to a partial function $\hat{\mathscr{I}}$ on terms:

$$\begin{aligned} \hat{\mathscr{I}}(v) &:= \mathscr{I}(v) \\ \hat{\mathscr{I}}(st) &:= fa \qquad \text{if } \hat{\mathscr{I}}s = f \text{ and } \hat{\mathscr{I}}t = a \\ \hat{\mathscr{I}}(\lambda x.s) &:= f \qquad \text{if } \lambda x.s \in \Lambda_{\sigma\tau},\ f \in \mathscr{D}_{\sigma\tau} \text{ and } \forall a \in \mathscr{D}_\sigma\text{: } \widehat{\mathscr{I}^x_a}s = fa \end{aligned}$$

Note that $\hat{\mathscr{I}}(st)$ is undefined if either $\hat{\mathscr{I}}s$ or $\hat{\mathscr{I}}t$ is undefined. Similarly, $\hat{\mathscr{I}}(\lambda x.s)$ is undefined if either $\widehat{\mathscr{I}^x_a}s$ is undefined for some $a \in \mathscr{D}_\sigma$ or if the appropriate function $f : \mathscr{D}_\sigma \to \mathscr{D}_\tau$ is not in $\mathscr{D}_{\sigma\tau}$. If $\hat{\mathscr{I}}$ is a total function, then we say $\mathscr{I}$ is an *interpretation*.

We record the following useful fact which can be proven by an easy induction on terms.

Proposition 3 *Let $\mathscr{D}$ be a frame and $s \in \Lambda_\sigma$ be a term. If $\mathscr{I}$ and $\mathscr{J}$ are assignments into $\mathscr{D}$ such that $\mathscr{I}c = \mathscr{J}c$ for every logical constant c and $\mathscr{I}x = \mathscr{J}x$ for every $x \in \mathscr{V}s$, then either $s \notin \mathrm{Dom}(\hat{\mathscr{I}})$ and $s \notin \mathrm{Dom}(\hat{\mathscr{J}})$ or $s \in \mathrm{Dom}(\hat{\mathscr{I}})$, $s \in \mathrm{Dom}(\hat{\mathscr{J}})$ and $\hat{\mathscr{I}}s = \hat{\mathscr{J}}s$.*

A *(Henkin) model* $(\mathscr{D}, \mathscr{I})$ is a frame $\mathscr{D}$ and a logical interpretation $\mathscr{I}$ into $\mathscr{D}$. We say that a model $(\mathscr{D}, \mathscr{I})$ *satisfies* a formula s if and only if $\hat{\mathscr{I}}(s) = 1$. A formula is *satisfiable* if and only if there is a model $(\mathscr{D}, \mathscr{I})$ such that $\hat{\mathscr{I}}(s) = 1$. We say a model $(\mathscr{D}, \mathscr{I})$ is *countable* if $\mathscr{D}_\sigma$ is at most countable for every type σ. Note that even if $\mathscr{D}_\sigma$ is finite for every σ, the union $\bigcup_\sigma \mathscr{D}_\sigma$ will be countably infinite. Hence $(\mathscr{D}, \mathscr{I})$ is a countable model if and only if $\bigcup_\sigma \mathscr{D}_\sigma$ is countably infinite.

We say $(\mathscr{D}, \mathscr{I})$ is *a model of* a set of formulas A if $\hat{\mathscr{I}}(s) = 1$ for every $s \in A$. A set A of formulas is *satisfiable* if there is a model of A.

We assume a type preserving and total *normalization operator* $[\cdot]$ from terms to terms. A term is *normal* if and only if $[s] = s$. A set of terms is normal if every

Table 1 Properties of values of logical constants

Prop.	Where	Holds			For all
$\mathfrak{L}_*(a)$	$a \in \mathscr{D}_\iota$	always			
$\mathfrak{L}_\bot(a)$	$a \in \mathscr{D}_o$	when $a = 0$			
$\mathfrak{L}_\neg(n)$	$n \in \mathscr{D}_{oo}$	when $na = 1$	if and only if	$a = 0$	$a \in \mathscr{D}_o$
$\mathfrak{L}_\vee(d)$	$d \in \mathscr{D}_{ooo}$	when $dab = 1$	if and only if	$a = 1$ or $b = 1$	$a, b \in \mathscr{D}_o$
$\mathfrak{L}_{\forall_\sigma}(p)$	$p \in \mathscr{D}_{(\sigma o)o}$	when $pf = 1$	if and only if	$\forall a \in \mathscr{D}_\sigma\ fa = 1$	$f \in \mathscr{D}_{\sigma o}$
$\mathfrak{L}_{=_\sigma}(q)$	$q \in \mathscr{D}_{\sigma\sigma o}$	when $qab = 1$	if and only if	$a = b$	$a, b \in \mathscr{D}_\sigma$
$\mathfrak{L}_{\varepsilon_\sigma}(\Phi)$	$\Phi \in \mathscr{D}_{(\sigma o)\sigma}$	when $f(\Phi f) = 1$	if and only if	$\exists a \in \mathscr{D}_\sigma\ fa = 1$	$f \in \mathscr{D}_{\sigma o}$

element of this set is normal. Instead of committing to a specific operator such as β-normalization or $\beta\eta$-normalization, we require the following properties:

N1 $[[s]] = [s]$
N2 $[[s]t] = [st]$
N3 $[\nu s_1 \ldots s_n] = \nu[s_1] \ldots [s_n]$ if $\nu s_1 \ldots s_n \in \Lambda_\beta$ for some base type β and $n \geq 0$
N4 $\hat{\mathscr{I}}[s] = \hat{\mathscr{I}} s$ for every model $(\mathscr{D}, \mathscr{I})$
N5 $\mathscr{V}[s] \subseteq \mathscr{V} s$

Note that by N5 we know $[s] \in \Lambda_\sigma^X$ whenever $s \in \Lambda_\sigma^X$.

A *substitution* is a type preserving partial function from variables to terms. If θ is a substitution, x is a variable, and s is a term that has the same type as x, we write θ_s^x for the substitution that agrees everywhere with θ except on x where it yields s. For each substitution θ we assume there is a type preserving total function $\hat{\theta}$ from terms to terms such that the following properties hold:

S1 $\hat{\theta} x = \theta x$ for every $x \in \operatorname{Dom} \theta$
S2 $\hat{\theta}(st) = (\hat{\theta} s)(\hat{\theta} t)$
S3 $[(\hat{\theta}(\lambda x.s))t] = [\widehat{\theta_t^x} s]$
S4 $[\hat{\theta} s] = [s]$ if $\theta x = x$ for every $x \in \operatorname{Dom} \theta \cap \mathscr{V} s$
S5 $[\hat{\theta}[s]] = [\hat{\theta} s]$

The following two propositions demonstrate that abstract normalization and substitution satisfy two properties one would expect. The empty set $\emptyset$ is the substitution that is undefined on every variable.

Proposition 4 $[[\lambda x.s]t] = [\widehat{\emptyset_t^x} s]$

Proof $[[\lambda x.s]t] \overset{\text{S4}}{=} [[\hat{\emptyset}(\lambda x.s)]t] \overset{\text{N2}}{=} [(\hat{\emptyset}(\lambda x.s))t] \overset{\text{S3}}{=} [\widehat{\emptyset_t^x} s]$. □

Proposition 5 *Let $s \in \Lambda_{\sigma\tau}$, $x \in \mathscr{V}_\sigma$ and $t \in \Lambda_\sigma$. If $x \notin \mathscr{V} s$ and $x \notin \mathscr{V}[sx]$, then $[sx] = [st]$.*

Proof $[sx] \overset{\text{N1}}{=} [[sx]] \overset{\text{S4}}{=} [\widehat{\emptyset_t^x}[sx]] \overset{\text{S5}}{=} [\widehat{\emptyset_t^x}(sx)] \overset{\text{S2}}{=} [\widehat{\emptyset_t^x}(s)\widehat{\emptyset_t^x}(x)] \overset{\text{S1}}{=} [\widehat{\emptyset_t^x}(s)t] \overset{\text{N2}}{=} [[\widehat{\emptyset_t^x}(s)]t] \overset{\text{S4}}{=} [[s]t] \overset{\text{N2}}{=} [st]$. □

For each set A of formulas and type σ we define a nonempty set $\mathscr{U}_\sigma^A \subseteq \Lambda_\sigma$ as follows.

- Let $\mathscr{U}_o^A = \{\bot, \top\}$.
- Let $\mathscr{U}_\iota^A$ be the set of discriminating terms in A if there is some discriminating term in A and $\{*\}$ otherwise.
- Let $\mathscr{U}_{\sigma\tau}^A = \{[s] | s \in \Lambda_{\sigma\tau}, \mathscr{V} s \subseteq \mathscr{V} A\}$.

When trying to refute a set A of formulas, all our instantiations of type σ will come from the *universe* $\mathscr{U}_\sigma^A$. When the set A is clear in context, we write $\mathscr{U}_\sigma$.

3 Tableau Calculus

We now introduce a tableau calculus and define a notion of refutability. A *branch* is a finite set of normal formulas. A *step* is an $n+1$-tuple $\langle \mathcal{A}, \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle$ of branches where $n \geq 1$, $\bot \notin \mathcal{A}$ and $\mathcal{A} \subset \mathcal{A}_i$ for each $i \in \{1, \ldots, n\}$. The branch $\mathcal{A}$ is the *head* of the step $\langle \mathcal{A}, \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle$ and each $\mathcal{A}_i$ is an *alternative*. A *rule* is a set of steps, and is usually indicated by a schema. For example, the schema for $\mathcal{T}_{\text{BE}}$ given in Fig. 1 indicates the set of steps $\langle \mathcal{A}, \mathcal{A}_1, \mathcal{A}_2 \rangle$ where $(s \neq_o t) \in \mathcal{A}$, $\bot \notin \mathcal{A}$, $\{s, \neg t\} \not\subseteq \mathcal{A}$, $\{\neg s, t\} \not\subseteq \mathcal{A}$, $\mathcal{A}_1 = \mathcal{A} \cup \{s, \neg t\}$ and $\mathcal{A}_2 = \mathcal{A} \cup \{\neg s, t\}$. We say a rule *applies to* a branch $\mathcal{A}$ if some step in the rule has $\mathcal{A}$ as its head. A tableau calculus is also a set of steps. Let $\mathcal{T}$ be the tableau calculus given as the union of the rules in Fig. 1.

In the rules $\mathcal{T}_{\text{MAT}}$ (the mating rule) and $\mathcal{T}_{\text{DEC}}$ (the decomposition rule) we assume $n \geq 1$ and w is a decomposable name (a variable or a choice operator). In the rule $\mathcal{T}_\forall$ the instantiation term t must belong to the set $\mathcal{U}_\sigma^{\mathcal{A}}$ where $\mathcal{A}$ is the head of the step. In the rule $\mathcal{T}_{\neg\forall}$ the variable x must be *fresh* in the sense that it is not in $\mathcal{V}\mathcal{A}$ where $\mathcal{A}$ is the head of the step. We restrict the $\mathcal{T}_{\neg\forall}$ to apply only in the case where there is no decomposable name $w \in \mathcal{W}_\sigma$ such that $\neg[sw]$ is in the head $\mathcal{A}$. In the context of an automated prover, this restriction implies there is no need to apply the $\mathcal{T}_{\neg\forall}$ rule to a formula $\neg\forall s$ more than once. For example, if x and w are variables of type o, then $\mathcal{T}_{\neg\forall}$ does not apply to the branch $\{\neg w, \neg\forall x.x\}$. Without the restriction, we could continue to apply $\mathcal{T}_{\neg\forall}$ to add new formulas $\neg w_2, \neg w_3, \ldots$.

We explain the choice rule $\mathcal{T}_\varepsilon$. Whenever we must consider εs, either s corresponds to the empty set and hence $\forall x.\neg(sx)$ holds, or s represents a set containing at least one element and $s(\varepsilon s)$ holds. Note that we obtain a complete calculus even though we only apply the choice rule when εs occurs on the branch in the form $\mathcal{C}[\varepsilon s]$ for some accessibility context $\mathcal{C}$. That is, the choice rule only applies using εs when the branch contains a formula of the form $\varepsilon s t_1 \cdots t_n$, $\neg(\varepsilon s t_1 \cdots t_n)$, $(\varepsilon s t_1 \cdots t_n) \neq_\iota u$ or $u \neq_\iota (\varepsilon s t_1 \cdots t_n)$. This is a tighter restriction than the one given for the choice rule in [24].

The set of *refutable* branches is defined inductively as follows. If $\bot \in \mathcal{A}$, then $\mathcal{A}$ is refutable. If $\langle \mathcal{A}, \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle$ is a step in $\mathcal{T}$ and every alternative $\mathcal{A}_i$ is refutable, then $\mathcal{A}$ is refutable.

Proposition 6 (Soundness) *If $\mathcal{A}$ is refutable, then $\mathcal{A}$ is unsatisfiable.*

$$\mathcal{T}_\neg \; \frac{s, \neg s}{\bot} \qquad \mathcal{T}_{\neq} \; \frac{s \neq_\iota s}{\bot} \qquad \mathcal{T}_{\neg\neg} \; \frac{\neg\neg s}{s} \qquad \mathcal{T}_\vee \; \frac{s \vee t}{s \mid t} \qquad \mathcal{T}_{\neg\vee} \; \frac{\neg(s \vee t)}{\neg s, \neg t} \qquad \mathcal{T}_\forall \; \frac{\forall_\sigma s}{[st]} \; t \in \mathcal{U}_\sigma$$

$$\mathcal{T}_{\neg\forall} \; \frac{\neg\forall_\sigma s}{\neg[sx]} \; x \in \mathcal{V}_\sigma \text{ fresh} \qquad \mathcal{T}_\varepsilon \; \frac{}{[\forall x.\neg(sx)] \mid [s(\varepsilon s)]} \; \varepsilon s \text{ accessible}, x \notin \mathcal{V}s \qquad \mathcal{T}_{\text{BE}} \; \frac{s \neq_o t}{s, \neg t \mid \neg s, t}$$

$$\mathcal{T}_{\text{BQ}} \; \frac{s =_o t}{s, t \mid \neg s, \neg t} \qquad \mathcal{T}_{\text{FE}} \; \frac{s \neq_{\sigma\tau} t}{\neg[\forall x.sx = tx]} \; x \notin \mathcal{V}s \cup \mathcal{V}t \qquad \mathcal{T}_{\text{FQ}} \; \frac{s =_{\sigma\tau} t}{[\forall x.sx = tx]} \; x \notin \mathcal{V}s \cup \mathcal{V}t$$

$$\mathcal{T}_{\text{MAT}} \; \frac{ws_1 \ldots s_n, \neg wt_1 \ldots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n} \qquad \mathcal{T}_{\text{DEC}} \; \frac{ws_1 \ldots s_n \neq_\iota wt_1 \ldots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n} \qquad \mathcal{T}_{\text{CON}} \; \frac{s =_\iota t, u \neq_\iota v}{s \neq u, t \neq u \mid s \neq v, t \neq v}$$

Fig. 1 Tableau rules defining the tableau calculus $\mathcal{T}$

Proof It is enough to check for each step $\langle A, A_1, \cdots, A_n \rangle$ in $\mathcal{T}$ that if A is satisfiable, then A_i is satisfiable for some $i \in \{1, \ldots, n\}$. Each case is easy. For the steps involving the normalization operator, property N4 is used. For $\mathcal{T}_{\neg\forall}$ use Proposition 3. □

Example 1 Let $p \in \mathcal{V}_{\iota o}$. For this example, assume p and $\lambda x.\neg px$ are normal. We refute the set $\{p(\varepsilon x.\neg px), \neg p(\varepsilon p)\}$ using the rules $\mathcal{T}_{\text{MAT}}$, $\mathcal{T}_{\varepsilon}$, $\mathcal{T}_{\forall}$ and $\mathcal{T}_{\neg}$. Note that $\mathcal{T}_{\forall}$ is used with the instantiation term $\varepsilon x.\neg px$ which is a discriminating term when $(\varepsilon x.\neg px) \neq \varepsilon p$ is on the branch.

$$\begin{array}{c}
p(\varepsilon x.\neg px) \\
\neg p(\varepsilon p) \\
(\varepsilon x.\neg px) \neq \varepsilon p \\
\hline
\begin{array}{c|c}
\begin{array}{c}\forall x.\neg px \\ \neg p(\varepsilon x.\neg px) \\ \bot\end{array} & \begin{array}{c} p(\varepsilon p) \\ \bot \end{array}
\end{array}
\end{array}$$

4 Evident Sets and Model Existence

Let E be a set of normal formulas. We say E is *evident* if it satisfies the conditions in Fig. 2. We call these conditions *evidence conditions*. The evidence conditions are similar to conditions considered by Hintikka [20] in the context of predicate logic. For this reason, sets satisfying such conditions are sometimes called “Hintikka sets” (cf. [9]). Hintikka called such sets “model sets” in [20] because in predicate logic (without equality) each such set induces a model in a very natural way. In this

$\mathcal{E}_{\bot}$	$\bot$ is not in E.
$\mathcal{E}_{\neg}$	If $\neg s$ is in E, then s is not in E.
$\mathcal{E}_{\neq}$	$s \neq_\iota s$ is not in E.
$\mathcal{E}_{\neg\neg}$	If $\neg\neg s$ is in E, then s is in E.
$\mathcal{E}_{\vee}$	If $s \vee t$ is in E, then either s or t is in E.
$\mathcal{E}_{\neg\vee}$	If $\neg(s \vee t)$ is in E, then $\neg s$ and $\neg t$ are in E.
$\mathcal{E}_{\forall}$	If $\forall_\sigma s$ is in E, then $[st]$ is in E for every $t \in \mathcal{U}_\sigma^E$.
$\mathcal{E}_{\neg\forall}$	If $\neg\forall_\sigma s$ is in E, then $\neg[sw]$ is in E for some decomposable name $w \in \mathcal{W}_\sigma$.
$\mathcal{E}_{\text{MAT}}$	If $ws_1 \ldots s_n$ and $\neg wt_1 \ldots t_n$ are in E where $n \geq 1$, then $s_i \neq t_i$ is in E for some $i \in \{1, \ldots, n\}$.
$\mathcal{E}_{\text{DEC}}$	If $ws_1 \ldots s_n \neq_\iota wt_1 \ldots t_n$ is in E where $n \geq 1$, then $s_i \neq t_i$ is in E for some $i \in \{1, \ldots, n\}$.
$\mathcal{E}_{\text{CON}}$	If $s =_\iota t$ and $u \neq_\iota v$ are in E, then either $s \neq u$ and $t \neq u$ are in E or $s \neq v$ and $t \neq v$ are in E.
$\mathcal{E}_{\text{BE}}$	If $s \neq_o t$ is in E, then either s and $\neg t$ are in E or $\neg s$ and t are in E.
$\mathcal{E}_{\text{BQ}}$	If $s =_o t$ is in E, then either s and t are in E or $\neg s$ and $\neg t$ are in E.
$\mathcal{E}_{\text{FE}}$	If $s \neq_{\sigma\tau} t$ is in E, then $[sw] \neq [tw]$ is in E for some decomposable name $w \in \mathcal{W}_\sigma$.
$\mathcal{E}_{\text{FQ}}$	If $s =_{\sigma\tau} t$ is in E, then $[su] = [tu]$ is in E for every $u \in \mathcal{U}_\sigma^E$.
$\mathcal{E}_{\varepsilon}$	If $\varepsilon_\sigma s$ is accessible in E, then either $[s(\varepsilon s)]$ is in E or $\neg[st]$ is in E for every $t \in \mathcal{U}_\sigma^E$.

Fig. 2 Evidence conditions

section we will prove that each evident set E induces a Henkin model of E, though the construction in our higher-order setting is more involved than in the first-order setting.

Before continuing, we consider an additional property which an evident set might satisfy. We say E is *complete* if for every formula s such that $\mathcal{V}s \subseteq \mathcal{V}E$ either $s \in E$ or $\neg s \in E$. With the exception of the restriction of the free variables in s to those occurring free in E, the property was called "saturation" in [9] (cf. Definition 6.24) and [8]. The terminology changed to "complete" in [14]. It will turn out that if E is complete, then the model we construct will interpret each type as a set that is at most countable.

Most of the evidence conditions in Fig. 2 correspond directly to a tableau rule in Fig. 1. On the other hand, the conditions $\mathcal{E}_{\text{FE}}$, $\mathcal{E}_{\text{FQ}}$ and $\mathcal{E}_{\varepsilon}$ are formulated in a slightly different way than the corresponding tableau rules $\mathcal{T}_{\text{FE}}$, $\mathcal{T}_{\text{FQ}}$ and $\mathcal{T}_{\varepsilon}$. The tableau rules are formulated in a way that makes proof search more practical while the evidence conditions are formulated in a way that will ease the model construction. The next proposition demonstrates that these three evidence conditions could also be formulated differently. Later we will use the proposition to help prove certain sets are evident.

Proposition 7 *Let E be a set of normal formulas satisfying $\mathcal{E}_{\forall}$ and $\mathcal{E}_{\neg\forall}$.*

1. *For $s, t \in \Lambda_{\sigma\tau}$ and $x \in \mathcal{V}_\sigma \setminus (\mathcal{V}s \cup \mathcal{V}t)$, if $\neg[\forall x.sx =_\tau tx]$ is in E, then $[sw] \neq [tw]$ is in E for some decomposable name $w \in \mathcal{W}_\sigma$.*
2. *For $s, t \in \Lambda_{\sigma\tau}$ and $x \in \mathcal{V}_\sigma \setminus (\mathcal{V}s \cup \mathcal{V}t)$, if $[\forall x.sx =_\tau tx]$ is in E, then $[su] = [tu]$ is in E for every $u \in \mathcal{U}_\sigma^E$.*
3. *For $s \in \Lambda_{\sigma\tau}$ and $x \in \mathcal{V}_\sigma \setminus \mathcal{V}s$, if $[\forall x.\neg sx]$ is in E, then $\neg[st]$ is in E for every $t \in \mathcal{U}_\sigma^E$.*

Proof We prove only the first fact. The proofs of other two are similar. Assume $s, t \in \Lambda_{\sigma\tau}$, $x \in \mathcal{V}_\sigma \setminus (\mathcal{V}s \cup \mathcal{V}t)$ and $\neg[\forall x.sx =_\tau tx] \in E$. By N3 and $\mathcal{E}_{\neg\forall}$ we know $\neg[[\lambda x.sx =_\tau tx]w]$ is in E for some $w \in \mathcal{W}_\sigma$. By Proposition 4, S1, S2, S4 and N3 we know $[[\lambda x.sx =_\tau tx]w]$ is the same as $[sw] =_\tau [tw]$. Thus $[sw] \neq_\tau [tw]$ is in E as desired. □

Let E be an evident set and let X be the set $\mathcal{V}E$ of free variables in E. In the rest of this section we will construct a model of E. The construction is similar to the ones in [14] except for complications that arise from the inclusion of a choice operator and from instantiation restrictions.

We next define a binary relation $\triangleright_\sigma$ between terms $s \in \Lambda_\sigma^X$ and values $a \in \mathcal{D}_\sigma$. When the relation $s \triangleright_\sigma a$ holds we say *s can be a* or *a is a possible value of s*. A relation similar to $\triangleright$ was defined independently by Takahashi [33] and Prawitz [27] in order to prove cut-elimination for a higher-order calculus. (The phrase *possible value* was used by Prawitz [27].) Such a relation can also be found in [24] and [14]. An important difference between $\triangleright$ as defined here and the analogous relations defined in earlier works is that the present $\triangleright$ only relates terms in Λ^X to values. That is, we restrict our attention to terms that contain free variables from the set X, i.e., the variables free in E. This modification is necessary to obtain completeness in the presence of our restriction on instantiations.

We define $\rhd_\sigma$ by induction on types. For each σ, let $\mathcal{D}_\sigma$ be the range of $\rhd_\sigma$, i.e., the set of all a such that there is some $s \in \Lambda_\sigma^X$ with $s \rhd_\sigma a$.

- $s \rhd_o 0$ if $s \in \Lambda_o^X$ and $[s] \notin E$.
- $s \rhd_o 1$ if $s \in \Lambda_o^X$ and $\neg[s] \notin E$.
- $s \rhd_\iota \Delta$ if $s \in \Lambda_\iota^X$, Δ is a discriminant (of E), and either $[s]$ is not a discriminating term or $[s] \in \Delta$.
- $s \rhd_{\sigma\tau} f$ if $s \in \Lambda_{\sigma\tau}^X$, $f : \mathcal{D}_\sigma \to \mathcal{D}_\tau$ and $st \rhd_\tau fa$ whenever $t \rhd_\sigma a$.

Clearly we have $\rhd_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$. Also, by the definition of $\mathcal{D}$ for every $a \in \mathcal{D}_\sigma$ there is some $s \in \Lambda_\sigma^X$ such that $s \rhd_\sigma a$. For any set $T \subseteq \Lambda_\sigma^X$ we write $T \rhd a$ if $s \rhd a$ for every $s \in T$.

Note that if $s \in \Lambda^X$, then $[s] \in \Lambda^X$ by N5. Thus, it makes sense to ask in what circumstances $[s] \rhd a$ holds for such s. The following lemma answers this question.

Lemma 1 *For all types σ, terms $s \in \Lambda_\sigma^X$ and values $a \in \mathcal{D}_\sigma$, $s \rhd a$ if and only if $[s] \rhd a$.*

Proof This follows by an easy induction on types σ using N1, N2 and N5. The proof is essentially the same as that of Proposition 3.1 in [14]. □

The next proposition records a number of useful facts about $\rhd$ and $\mathcal{D}$. In particular, $\mathcal{D}$ is a frame and for every value $a \in \mathcal{D}_\sigma$ there is some $t \in \mathcal{U}_\sigma^E$ with possible value a. We need such a result to prove completeness since instantiations are restricted to terms in $\mathcal{U}_\sigma^E$.

Proposition 8

1. *$\bot \rhd 0$ and $\top \rhd 1$. In particular, $\mathcal{D}_o = \{0, 1\}$.*
2. *For every discriminant Δ, there is a term $t \in \mathcal{U}_\iota^E$ with possible value Δ. In particular, $\mathcal{D}_\iota$ is the set of all discriminants.*
3. *For all types σ and $a \in \mathcal{D}_\sigma$ there is a term $t \in \mathcal{U}_\sigma^E$ such that $t \rhd a$.*
4. *If $t \rhd_\mu b$ and $x \in \mathcal{V}_\tau \setminus \mathcal{V}t$, then $\lambda x.t \rhd K_b$ where $K_b : \mathcal{D}_\tau \to \mathcal{D}_\mu$ is the constant b function.*
5. *For all types σ, $\mathcal{D}_\sigma$ is nonempty.*
6. *$\mathcal{D}$ is a frame.*

Proof

1. By $\mathcal{E}_\bot$ and $\mathcal{E}_{\neg\neg}$ we know $\bot \notin E$ and $\neg\neg\bot \notin E$. By N3 $[\bot]$ is $\bot$ and $[\neg\bot]$ is $\neg\bot$. Hence $\bot \rhd 0$ and $\neg\bot \rhd 1$.
2. Let Δ be a discriminant. If there are no discriminating terms, then $*$ is in $\mathcal{U}_\iota^E$ and $* \rhd \Delta$. Suppose there is a discriminating term s. By $\mathcal{E}_{\neq}$ we know $s \neq_\iota s$ is not in E. Since $\emptyset \subset \{s\}$ and Δ is maximal, we know Δ must not be empty. Let $t \in \Delta$ be given. Clearly $t \in \mathcal{U}_\iota^E$ and $t \rhd \Delta$.
3. By case analysis on σ. The cases for base types follow directly from Proposition 8(1) and (2). Let σ be $\tau\mu$ and $f \in \mathcal{D}_{\tau\mu}$ be given. By definition there is some $s \in \Lambda_{\tau\mu}^X$ such that $s \rhd f$. Since X is $\mathcal{V}E$, $[s] \in \mathcal{U}_{\tau\mu}^E$. By Lemma 1 $[s] \rhd f$ and we are done.
4. Assume $\lambda x.t \not\rhd K_b$. By Lemma 1 $[\lambda x.t] \not\rhd K_b$. We know that there is a term $s \in \Lambda_\tau^X$ and a value $a \in \mathcal{D}_\tau$ such that $s \rhd a$ but $[\lambda x.t]s \not\rhd K_b a$. By the definition of K_b

and by Lemma 1 $[[\lambda x.t]s] \not\triangleright b$. We β-reduce according to Proposition 4 and use S4 and Lemma 1 to get $t \not\triangleright b$. This is a contradiction.

5. By induction on σ. The case for type o follows directly from Proposition 8(1). The case for type ι follows from Propositions 8(2) and 2. Let σ be $\tau\mu$. By induction there is some $b \in \mathscr{D}_\mu$. By definition there is some $t \in \Lambda_\mu^X$ such that $t \triangleright b$. Let $K_b : \mathscr{D}_\tau \to \mathscr{D}_\mu$ be the constant b function and let x be a variable not in $\mathscr{V}t$. By Proposition 8(4) $\lambda x.t \triangleright K_b$. By the definition of $\triangleright$ we know $K_b \in \mathscr{D}_\sigma$.
6. $\mathscr{D}_\sigma$ is nonempty for all σ by Proposition 8(5). $\mathscr{D}_o = \{0, 1\}$ by Proposition 8(1). $\mathscr{D}_{\tau\mu}$ only contains total functions from $\mathscr{D}_\tau$ to $\mathscr{D}_\mu$ for all μ and τ by the definition of $\triangleright$. □

The evident set E insists some terms must be interpreted differently. We use this information to define a relation $\sharp$. For $s, t \in \Lambda_\sigma^X$ we say $s \sharp t$ *holds* if either $s \neq t$ or $t \neq s$ is in E. We record a simple and useful fact.

Lemma 2 *For any variable $x \in \mathscr{V}_\sigma$, $[\lambda x.\bot]\sharp[\lambda x.\bot]$ does not hold.*

Proof Assume $[\lambda x.\bot]\sharp[\lambda x.\bot]$ holds. There is some $w \in \mathscr{W}_\tau$ such that $[(\lambda x.\bot)w]\sharp[(\lambda x.\bot)w]$ by $\mathscr{E}_{\text{FE}}$ and N2. We know $[(\lambda x.\bot)w]$ is $\bot$ by Proposition 4, S4 and N3. Hence $\bot\sharp\bot$. This contradicts $\mathscr{E}_{\text{BE}}$ and $\mathscr{E}_\bot$. □

We now turn to a notion of compatibility of terms.

Definition 1 For each type σ we define when two terms $s, t \in \Lambda_\sigma^X$ are *compatible* (written $s \parallel t$) by induction on types.

$\sigma = o$: $s \parallel t$ if $\{[s], \neg[t]\} \not\subseteq E$ and $\{\neg[s], [t]\} \not\subseteq E$.
$\sigma = \iota$: $s \parallel t$ if $[s]\sharp[t]$ does not hold.
$\sigma = \tau\mu$: $s \parallel t$ if for all $u, v \in \Lambda_\tau^X$ $u \parallel v$ implies $su \parallel tv$.

We say a set $T \subseteq \Lambda_\sigma^X$ is *compatible* if $s \parallel t$ for all $s, t \in T$.

The next lemma provides relationships between compatibility of terms and the presence of disequations in E. Note that part 2 of the lemma implies $\varepsilon_\sigma \parallel \varepsilon_\sigma$ for every type σ and $x \parallel x$ for every variable $x \in X$. The lemma is the same as Lemma 6.5 in [14] except for the restriction of free variables to X and the inclusion of ε_σ in part (2). The free variable restriction does cause a slight complication in the proof since the decomposable name w in the condition $\mathscr{E}_{\text{FE}}$ may be a variable not in X.

Lemma 3 *For all types σ we have the following:*

1. *For all $s, t \in \Lambda_\sigma^X$, if $[s]\sharp[t]$, then $s \nparallel t$.*
2. *For all $ws_1 \cdots s_n, wt_1, \cdots t_n \in \Lambda_\sigma^X$ where $n \geq 0$ and w is a decomposable name, either $ws_1 \cdots s_n \parallel wt_1, \cdots t_n$ or there is some $i \in \{1, \ldots, n\}$ such that $[s_i]\sharp[t_i]$.*

Proof By mutual induction on σ. The base cases for Lemma 3(1) follow from $\mathscr{E}_{\text{BE}}$ and the definition of compatibility. The base cases for Lemma 3(2) follow from N3, $\mathscr{E}_{\text{MAT}}$ and $\mathscr{E}_{\text{DEC}}$ since w is decomposable. The case for Lemma 3(2) when σ is $\tau\mu$ easily follows from the inductive hypotheses for Lemma 3(1) at τ and for Lemma 3(2) at μ.

The only complicated case is proving Lemma 3(1) when σ is $\tau\mu$. Assume $s \parallel t$ and $[s]\sharp[t]$ both hold. By $\mathscr{E}_{\text{FE}}$ and N2 $[sw]\sharp[tw]$ for some decomposable $w \in \mathscr{W}_\tau$. If $w \in X$ or w is a choice operator, then $w \parallel_\tau w$ by inductive hypothesis (2) and so $sw \parallel_\mu tw$, contradicting inductive hypothesis (1). Otherwise, $w \in \mathscr{V} \setminus X$. In particular, $w \notin \mathscr{V}s \cup \mathscr{V}t \cup \mathscr{V}[sw] \cup \mathscr{V}[tw]$. By Proposition 5 we know $[sw]$ is $[s(\varepsilon w.\bot)]$ and $[tw]$ is $[t(\varepsilon w.\bot)]$. By inductive hypothesis (2) and Lemma 2 we know $\varepsilon w.\bot \parallel \varepsilon w.\bot$. Hence $s(\varepsilon w.\bot) \parallel t(\varepsilon w.\bot)$, contradicting the inductive hypothesis (1). □

The next lemma relates compatibility to $\triangleright$. This lemma is very similar to Lemma 7.3 in [14]. Fortunately, the restriction of free variables to X does not cause complications in the proof. The axiom of choice is used twice in this proof: once directly and once indirectly via Proposition 1.

Lemma 4 *For all sets $T \subseteq \Lambda^X_\sigma$, T is compatible if and only if there is a value $a \in \mathscr{D}_\sigma$ such that $T \triangleright a$.*

Proof The proof is by induction on σ. Note that if T is empty then $T \triangleright a$ for all $a \in \mathscr{D}_\sigma$. (By Proposition 8(5) there is some $a \in \mathscr{D}_\sigma$.) In the cases below, we assume T is nonempty.

- $\sigma = \iota$, $\Rightarrow$. Let T be compatible. By Proposition 1 there exists a discriminant a that extends $\{\,[t] \text{ discriminating} \mid t \in T\,\}$. The claim follows since $T \triangleright a$.
- $\sigma = \iota$, $\Leftarrow$. Suppose $T \triangleright a$ and T is not compatible. Then there are terms $s, t \in T$ such that $([s]{\neq}[t]) \in E$. Thus $[s]$ and $[t]$ cannot be both in a. This contradicts $s, t \in T \triangleright a$ since $[s]$ and $[t]$ are discriminating.
- $\sigma = o$, $\Rightarrow$. By contraposition. Suppose $T \not\triangleright 0$ and $T \not\triangleright 1$. Then there are terms $s, t \in T$ such that $[s], \neg[t] \in E$. Thus $s \nparallel t$. Hence T is not compatible.
- $\sigma = o$, $\Leftarrow$. By contraposition. Suppose $s \nparallel_o t$ for $s, t \in T$. Then $[s], \neg[t] \in E$ without loss of generality. Hence $s \not\triangleright 0$ and $t \not\triangleright 1$. Thus $T \not\triangleright 0$ and $T \not\triangleright 1$.
- $\sigma = \tau\mu$, $\Rightarrow$. Let T be compatible. We define $T_a := \{\, ts \mid t \in T,\ s \triangleright_\tau a\,\}$ for every value $a \in \mathscr{D}_\tau$ and show that T_a is compatible. Let $t_1, t_2 \in T$ and $s_1, s_2 \triangleright_\tau a$. It suffices to show $t_1 s_1 \parallel t_2 s_2$. By the inductive hypothesis $s_1 \parallel_\tau s_2$. Since T is compatible, $t_1 \parallel t_2$. Hence $t_1 s_1 \parallel t_2 s_2$. By the inductive hypothesis we now know that for every $a \in \mathscr{D}_\tau$ there is a $b \in \mathscr{D}_\mu$ such that $T_a \triangleright_\mu b$. By the axiom of choice, there is a function $f \in \mathscr{D}_\sigma$ such that $T_a \triangleright_\mu fa$ for every $a \in \mathscr{D}_\tau$. Thus $T \triangleright_\sigma f$.
- $\sigma = \tau\mu$, $\Leftarrow$. Let $T \triangleright_\sigma f$ and $s, t \in T$. We will prove $s \parallel_\sigma t$. Let $u, v \in \Lambda^X_\tau$ be such that $u \parallel_\tau v$. It suffices to prove $su \parallel_\mu tv$. By the inductive hypothesis $u, v \triangleright_\tau a$ for some value a. Hence $su, tv \triangleright_\mu fa$. Thus $su \parallel_\mu tv$ by the inductive hypothesis. □

We now turn to the interpretation of the choice operators. We use a construction similar to that of Mints [24] adapted to our setting.

Let $f \in \mathscr{D}_{\sigma o}$ be a function and $w \in \mathscr{W}_{(\sigma o)\sigma}$ be a decomposable name. We write $f \propto ws$ (read f *chooses* ws) when $s \triangleright f$ and $w[s]$ is accessible in E. Let $f^w := \{ws \in \Lambda^X_\sigma \mid f \propto ws\}$.

Lemma 5 *For all $f \in \mathscr{D}_{\sigma o}$ and $w \in \mathscr{W}_{(\sigma o)\sigma} \cap \Lambda^X_{(\sigma o)\sigma}$, there is some $a \in \mathscr{D}_\sigma$ such that $f^w \triangleright a$.*

Proof We show that f^w is compatible. Lemma 4 gives us the claim. Let $ws, wt \in f^w$. By the definition of $\propto$, $s, t \rhd f$ and so $s \parallel t$ by Lemma 4. By Lemma 3(2) $w \parallel w$ and so $ws \parallel wt$. □

For each type σ we will now obtain a function $\Phi_\sigma : \mathscr{D}_{\sigma o} \to \mathscr{D}_\sigma$ that will serve as the interpretation of the choice operator ε_σ. For each σ we choose Φ_σ such that

$$\Phi_\sigma f = \begin{cases} \text{some } b & \text{such that } fb = 1 \text{ if } f^{\varepsilon_\sigma} \text{ is empty and such a } b \text{ exists.} \\ \text{some } a & \text{such that } f^{\varepsilon_\sigma} \rhd a. \end{cases}$$

The existence of an a in the second case follows from Lemma 5. Note that the second case includes the case in which f is the constant 0 function. In particular, if f is the constant 0 function and f^{ε_σ} is empty, then $\Phi_\sigma f$ can be any $a \in \mathscr{D}_\sigma$. The next three lemmas verify that Φ_σ can act as the interpretation of ε_σ.

Lemma 6 *Let ν be a name, $\nu t_1 \dots t_n \in \Lambda_\sigma^X$ and $a \in \mathscr{D}_\sigma$. If $\nu t_1 \dots t_n \not\rhd a$, then $\nu[t_1] \dots [t_n]$ is accessible in E.*

Proof We prove this by induction on σ.

- $\sigma = o$: Let $a = 0$. By the definition of $\rhd_o$ and N3, $\nu[t_1] \dots [t_n] \in E$. Let $a = 1$. Then, again by the definition of $\rhd_o$ and N3, $\neg\nu[t_1] \dots [t_n] \in E$.
- $\sigma = \iota$: By the definition of $\rhd_\iota$ and N3, we know that $\nu[t_1] \dots [t_n]$ is discriminating and hence accessible.
- $\sigma = \mu\tau$: By the definition of $\rhd_\sigma$, we know that there is some term $u \in \Lambda_\mu^X$ and some value $b \in \mathscr{D}_\mu$ such that $u \rhd b$ but $\nu t_1 \dots t_n u \not\rhd ab$. By the inductive hypothesis, we know that $\nu[t_1] \dots [t_n][u]$ is accessible in E. Hence, $\nu[t_1] \dots [t_n]$ is accessible. □

Lemma 7 *For any type σ we have $\varepsilon_\sigma \rhd \Phi_\sigma$.*

Proof Assume $\varepsilon \not\rhd \Phi$. Then, there are s, f such that $s \rhd f$ but $\varepsilon s \not\rhd \Phi f$. By Lemma 6 $\varepsilon[s]$ is accessible in E. Hence $\varepsilon s \in f^{\varepsilon_\sigma}$. There is some a such that $\Phi f = a$ and $f^{\varepsilon_\sigma} \rhd a$. Thus $\varepsilon s \rhd a$, a contradiction. □

Lemma 8 $\mathfrak{C}_{\varepsilon_\sigma}(\Phi_\sigma)$ *holds. That is, Φ as given above is a choice function.*

Proof Let $f \in \mathscr{D}_{\sigma o}$ be a function and $b \in \mathscr{D}_\sigma$ be such that $fb = 1$. Suppose $f(\Phi f) = 0$. Then f^{ε_σ} must be nonempty (by the definition of Φf). Choose some $\varepsilon s \in f^{\varepsilon_\sigma}$. By $\mathscr{E}_\varepsilon$ there are two possibilities:

1. $[s(\varepsilon s)] \in E$: In this case $s(\varepsilon s) \not\rhd 0$. On the other hand, $s \rhd f$ and $\varepsilon \rhd \Phi$ (by Lemma 7) and so $s(\varepsilon s) \rhd f(\Phi f)$. This contradicts our assumption that $f(\Phi f) = 0$.
2. $\neg[st] \in E$ for every $t \in \mathscr{U}_\sigma^E$: By Proposition 8(3) there is some term $t' \in \mathscr{U}_\sigma^E$ such that $t' \rhd b$. Hence $\neg[st'] \in E$. By the definition of $\rhd_o$, $st' \not\rhd 1$. On the other hand, we know $st' \rhd fb$ since $s \rhd f$ and $t' \rhd b$, contradicting the assumption that $fb = 1$.

□

The next lemma will ensure we can correctly interpret equality.

Lemma 9 *If $s \triangleright_\sigma a$, $t \triangleright_\sigma b$ and $s = t$ is in E, then $a = b$.*

Proof By contradiction and induction on σ. Assume $s \triangleright_\sigma a$, $t \triangleright_\sigma b$, $(s{=}t) \in E$, and $a \neq b$. Case analysis.

$\sigma = o$. By $\mathscr{E}_{\text{BQ}}$ either $s, t \in E$ or $\neg s, \neg t \in E$. Hence a and b are either both 1 or both 0. Contradiction.

$\sigma = \iota$. Since $a \neq b$, there must be discriminating terms of type ι. Since the discriminant a is maximal there is some $u \in a \setminus b$. Since b is also maximal, $b \cup \{u\}$ is not a discriminant. Hence there is some $v \in b$ such that $u \sharp v$. Since $(s{=}t) \in E$, we know by N3 that s and t are normal. By $\mathscr{E}_{\text{CON}}$ we know either $s \sharp u$ or $t \sharp v$. If $s \sharp u$, then s is discriminating and so $s \in a$, contradicting that a is a discriminant with $u \in a$. Likewise, if $t \sharp v$, then $t \in b$, contradicting $v \in b$.

$\sigma = \tau\mu$. Since $a \neq b$, there is some $d \in \mathscr{D}\tau$ such that $ad \neq bd$. By Proposition 8(3) there is some term $u \in \mathscr{U}_\tau^E$ such that $u \triangleright_\tau d$. Hence $su \triangleright ad$ and $tu \triangleright bd$. By Lemma 1 $[su] \triangleright_\mu ad$ and $[tu] \triangleright_\mu bd$. By $\mathscr{E}_{\text{FQ}}$ the equation $[su] = [tu]$ is in E, contradicting the inductive hypothesis. □

The next lemma will ensure we can correctly interpret universal quantifiers.

Lemma 10 *Let $s \in \Lambda_{\sigma o}^X$ be given. Let $f \in \mathscr{D}_{\sigma o}$ be such that $fb = 1$ for all $b \in \mathscr{D}_\sigma$. If $s \triangleright f$, then $\forall_\sigma s \triangleright 1$.*

Proof Assume $s \triangleright f$ and $\forall_\sigma s \not\triangleright 1$. Hence $\neg[\forall_\sigma s] \in E$. By N3, $\mathscr{E}_{\neg\forall}$ and N2 there is some $w \in \mathscr{W}_\sigma$ such that $\neg[sw]$ is in E. If $w \in X$ or w is a choice operator, then we obtain a contradiction using Lemma 3(2) and Lemma 4. Otherwise, $w \in \mathscr{V} \setminus X$. In particular, $w \notin \mathscr{V}s \cup \mathscr{V}[sw]$. By Proposition 5 $[sw]$ must be the same as $[s(\varepsilon w.\bot)]$. By Lemma 2 and Lemma 3(2) we know $\varepsilon x.\bot \parallel \varepsilon x.\bot$. By Lemma 4 there is some $b \in \mathscr{D}_\sigma$ such that $\varepsilon x.\bot \triangleright b$. Thus $s(\varepsilon x.\bot) \triangleright fb = 1$, contradicting that $\neg[s(\varepsilon x.\bot)]$ is in E. □

We now prove we can interpret every logical constant appropriately.

Proposition 9 *For each logical constant c of type σ there is some $a \in \mathscr{D}_\sigma$ such that $\mathfrak{L}_c(a)$ and $c \triangleright a$.*

Proof If c is a choice operator ε_σ, then we know $\varepsilon_\sigma \triangleright \Phi_\sigma$ and $\mathfrak{L}_{\varepsilon_\sigma}(\Phi_\sigma)$ by Lemmas 7 and 8. We know $\bot \triangleright 0$ by Proposition 8(1). If $*$ is not discriminating, then $* \triangleright \Delta$ for all $\Delta \in \mathscr{D}_\iota$. If $*$ is discriminating, then $* \triangleright \Delta$ for some $\Delta \in \mathscr{D}_\iota$ where $* \in \Delta$. Now, let $n : \mathscr{D}_o \to \mathscr{D}_o$ be the negation function and $d : \mathscr{D}_o \to \mathscr{D}_o \to \mathscr{D}_o$ be the disjunction function. For each σ let $p_\sigma : \mathscr{D}_{\sigma o} \to \mathscr{D}_o$ be the function such that $p_\sigma f = 1$ if and only if f is the constant 1 function. For each σ let $q_\sigma : \mathscr{D}_\sigma \to \mathscr{D}_\sigma \to \mathscr{D}_o$ be the function such that $q_\sigma ab = 1$ if and only if $a = b$. Each of the following statements is easily verified making extensive use of Lemma 1.

1. For all $s \in \Lambda_o^X$ and $a \in \mathscr{D}_o$, if $s \triangleright a$, then $\neg s \triangleright na$. (Use N3 and $\mathscr{E}_{\neg\neg}$.)
2. $\neg \triangleright n$. In particular, $n \in \mathscr{D}_{oo}$ and so $\mathfrak{L}_\neg(n)$. (Use definition of $\triangleright$ and (1).)
3. For all $s, t \in \Lambda_o^X$ and $a, b \in \mathscr{D}_o$, if $s \triangleright a$ and $t \triangleright b$, then $s \vee t \triangleright dab$. (Use N3, $\mathscr{E}_\vee$ and $\mathscr{E}_{\neg\vee}$.)
4. For all $s \in \Lambda_o^X$ and $a \in \mathscr{D}_o$, if $s \triangleright a$, then $(\vee s) \triangleright da$ and $da \in \mathscr{D}_{oo}$. (Use (3) and the definitions of $\triangleright$ and $\mathscr{D}_{oo}$.)

5. $d : \mathcal{D}_o \to \mathcal{D}_{oo}$. (Use (4).)
6. $\vee \rhd d$. In particular, $d \in \mathcal{D}_{ooo}$ and so $\mathfrak{L}_\vee(d)$. (Use (4) and the definitions of $\rhd$ and $\mathcal{D}_{oo}$.)
7. For all $s \in \Lambda^X_{\sigma o}$ and $f \in \mathcal{D}_{\sigma o}$, if $s \rhd f$, then $\forall_\sigma s \rhd p_\sigma f$. (Use Lemma 10 if f is the constant 1 function. Otherwise, use Proposition 8(3) and $\mathcal{E}_\forall$.)
8. $\forall_\sigma \rhd p_\sigma$. In particular, $p_\sigma \in \mathcal{D}_{(\sigma o)o}$ and so $\mathfrak{L}_{\forall_\sigma}(p_\sigma)$. (Use (7) and the definitions of $\rhd$ and $\mathcal{D}_{(\sigma o)o}$.)
9. For all $s, t \in \Lambda^X_\sigma$ and $a, b \in \mathcal{D}_\sigma$, if $s \rhd a$ and $t \rhd b$, then $s =_\sigma t \rhd q_\sigma ab$. (Use N3, Lemmas 4, 3(1), and 9.)
10. For all $s \in \Lambda^X_\sigma$ and $a \in \mathcal{D}_\sigma$, if $s \rhd a$, then $(=_\sigma s) \rhd q_\sigma a$ and $q_\sigma a \in \mathcal{D}_{\sigma o}$. (Use (9) and the definitions of $\rhd$ and $\mathcal{D}_{\sigma o}$.)
11. $q_\sigma : \mathcal{D}_\sigma \to \mathcal{D}_{\sigma o}$. (Use (10).)
12. $=_\sigma \rhd q_\sigma$. In particular, $q_\sigma \in \mathcal{D}_{\sigma\sigma o}$ and so $\mathfrak{L}_{=_\sigma}(q_\sigma)$. (Use (11) and the definitions of $\rhd$ and $\mathcal{D}_{\sigma\sigma o}$.) □

We say an assignment $\mathcal{I}$ into $\mathcal{D}$ is *admissible* if $c \rhd \mathcal{I}c$ for all logical constants c.

Lemma 11 *Let s be a term, θ be a substitution and $\mathcal{I}$ be an admissible assignment into $\mathcal{D}$. Suppose for every $x \in \mathcal{V}s$, $x \in \mathrm{Dom}\,\theta$ and $\theta x \rhd \mathcal{I}x$. Then $s \in \mathrm{Dom}\,\hat{\mathcal{I}}$ and $\hat{\theta}s \rhd \hat{\mathcal{I}}s$.*

Proof By induction on s. If s is a variable x, then $x \in \mathrm{Dom}\,\theta$ and $\theta x \rhd \mathcal{I}x$ by assumption and so $\hat{\theta}s \rhd \hat{\mathcal{I}}s$ by S1. If s is a logical constant c, then $\hat{\theta}s \rhd \hat{\mathcal{I}}s$ by admissibility of $\mathcal{I}$, S4 and Lemma 1. The case where s is an application term follows from the inductive hypotheses, S2 and the definitions of $\hat{\mathcal{I}}$ and $\rhd$. Finally, suppose s is of the form $\lambda x.t$ where $x \in \mathcal{V}_\sigma$ and $t \in \Lambda_\tau$. Let $u \rhd_\sigma a$ be given. We prove $(\hat{\theta}(\lambda x.t))u \rhd (\hat{\mathcal{I}}(\lambda x.t))a$. Applying the inductive hypothesis to t with θ^x_u and $\mathcal{I}^x_a$, we have that $t \in \mathrm{Dom}\,\widehat{\theta^x_u}$ and $\widehat{\theta^x_u}t \rhd \widehat{\mathcal{I}^x_a}t$. By S3 $[(\hat{\theta}(\lambda x.t))u]$ is $[\widehat{\theta^x_u}t]$. Two applications of Lemma 1 complete the proof. □

Using the tools above, we can obtain a logical, admissible interpretation. We prove this fact in a slightly more general form than we need here. The extra strength will be useful in a later section. Recall that X is $\mathcal{V}E$.

Lemma 12 *Let θ_0 be a substitution and $\mathcal{I}_0$ be an assignment such that $\theta_0 x \rhd \mathcal{I}_0 x$ for every $x \in \mathrm{Dom}\,\theta_0$. There is a substitution θ and a logical, admissible interpretation $\mathcal{I}$ such that $\hat{\theta}s \rhd \hat{\mathcal{I}}s$ for all $s \in \Lambda_\sigma$ and $\theta x = \theta_0 x$ and $\mathcal{I}x = \mathcal{I}_0 x$ for every $x \in \mathrm{Dom}\,\theta_0$.*

Proof We define an assignment $\mathcal{I}$ as follows. For each logical constant c we can choose $\mathcal{I}c$ such that $c \rhd \mathcal{I}c$ and $\mathfrak{L}_c(\mathcal{I}c)$ by Proposition 9. This ensures we will have a logical, admissible assignment. For each variable $x \in \mathrm{Dom}\,\theta_0$ let $\theta x := \theta_0 x$ and $\mathcal{I}x := \mathcal{I}_0 x$. For each variable $x \in \mathcal{V}_\sigma \setminus (\mathrm{Dom}\,\theta_0 \cup X)$ we take $\theta x := \varepsilon_\sigma y.\bot \in \Lambda^X_\sigma$ and $\mathcal{I}x := \Phi_\sigma K_0$ where K_0 is the constant 0 function. By Lemma 7 and Proposition 8 we know that $\varepsilon_\sigma y.\bot \rhd \Phi K_0$ and hence $\theta x \rhd \mathcal{I}x$ for every variable x. By Lemma 11 we know every $s \in \mathrm{Dom}\,\hat{\mathcal{I}}$ and $\hat{\theta}s \rhd \hat{\mathcal{I}}s$ for every term s. In particular, $\mathcal{I}$ is an interpretation. □

Now we can prove the model existence theorem for evident sets.

Theorem 1 (Model Existence) *Every evident set E has a model $(\mathcal{D}, \mathcal{I})$. Furthermore, we have the following:*

1. *If E is finite, then $\mathcal{D}_\sigma$ is finite for all types σ.*
2. *If E is complete, then $(\mathcal{D}, \mathcal{I})$ is a countable model.*

Proof Let E be an evident set and X be $\mathcal{V}E$. Take $\triangleright$ and $\mathcal{D}$ as defined in this section. We start by defining an assignment $\mathcal{I}_0$ and a substitution θ_0. We define $\theta_0 x := x$ for every $x \in X$. Note that $\mathrm{Dom}\,\theta_0 = X$. For each variable $x \in X$ we know $x \parallel x$ by Lemma 3(2) and so we can use Lemma 4 to choose $\mathcal{I}_0 x$ such that $x \triangleright \mathcal{I}_0 x$. For variables $x \in \mathcal{V}_\sigma \setminus X$ take $\mathcal{I}_0 x \in \mathcal{D}_\sigma$ arbitrarily, using Proposition 8(5). Using Lemma 12 we obtain a substitution θ and a logical, admissible interpretation $\mathcal{I}$ such that $\hat{\theta} s \triangleright \hat{\mathcal{I}} s$ for all $s \in \Lambda_\sigma$ and $\theta x = x$ for all $x \in X$. For every $s \in \Lambda_\sigma^X$ by S4 we know $[\hat{\theta} s] = [s]$ and so $s \triangleright \hat{\mathcal{I}} s$ by Lemma 1. Note that $(\mathcal{D}, \mathcal{I})$ is a model. For any $s \in E$, we know $s \triangleright_o \hat{\mathcal{I}} s$ and $s \not\triangleright 0$, and so $\hat{\mathcal{I}} s = 1$. Hence $(\mathcal{D}, \mathcal{I})$ is a model of E.

1. Assume E is finite. There are only finitely many discriminants of E. Hence $\mathcal{D}_\iota$ is finite. The fact that each $\mathcal{D}_\sigma$ is finite follows from an easy induction on types.
2. Since the set Λ_σ^X is countable, it is enough to give a surjective function from Λ_σ^X onto $\mathcal{D}_\sigma$. We will prove that $\triangleright_\sigma$ is such a surjective function. For every $s \in \Lambda_\sigma^X$ we know $s \triangleright \hat{\mathcal{I}} s$, so that $\triangleright_\sigma$ is total. To prove $\triangleright_\sigma$ is functional, suppose $s \in \Lambda_\sigma^X$, $s \triangleright a$ and $s \triangleright b$. Note that $[s = s]$ is $[s] = [s]$ by N3 and that $\mathcal{V}[s] \subseteq \mathcal{V}E$ by N5. Since we already know E is satisfiable, $[s] \neq [s]$ is not in E. Since E is complete, $[s] = [s]$ must be in E. Hence $a = b$ by Lemmas 1 and 9. Finally, $\triangleright_\sigma$ is surjective by the definition of $\mathcal{D}_\sigma$. □

We can now prove that if the tableau calculus $\mathcal{T}$ cannot make progress on a branch, then this branch is satisfiable and in fact has a model with finitely many individuals.

Corollary 1 *Let A be a branch. Suppose $\bot \notin A$ and A is not the head of any step in the calculus $\mathcal{T}$. Then A is evident and there is a model $(\mathcal{D}, \mathcal{I})$ of A where $\mathcal{D}_\sigma$ is finite for each type σ.*

Proof By Theorem 1, it suffices to prove A is evident. The evidence condition $\mathcal{E}_\bot$ follows from the assumption that $\bot \notin A$. The conditions $\mathcal{E}_\neg$ and $\mathcal{E}_{\neq}$ follow from $\bot \notin A$ and the assumption that the rules $\mathcal{T}_\neg$ and $\mathcal{T}_{\neq}$ do not apply to A. Except for $\mathcal{E}_{\mathrm{FE}}$, $\mathcal{E}_{\mathrm{FQ}}$ and $\mathcal{E}_\varepsilon$, the remaining evidence conditions follow immediately from the assumption that the corresponding rule does not apply. After we know $\mathcal{E}_\forall$ and $\mathcal{E}_{\neg\forall}$ hold for A, we can conclude that $\mathcal{E}_{\mathrm{FE}}$, $\mathcal{E}_{\mathrm{FQ}}$ and $\mathcal{E}_\varepsilon$ hold for A using Proposition 7 and the assumption that the corresponding rule does not apply. □

Example 2 Let $p \in \mathcal{V}_{\iota o}$ and $q \in \mathcal{V}_o$. For this example assume $[s] = s$ for all $\beta\eta$-normal forms s. We prove $\forall_o q.\varepsilon_{\iota o} p \neq \varepsilon_{\iota o} x.q$ is satisfiable. Consider the partial tableau shown in Fig. 3. Let A be the branch ending with $\forall x.\neg\bot$. It is easy to check that no more rules apply to A. In particular, consider the rule $\mathcal{T}_\varepsilon$. There are three accessible terms to consider: εp, $\varepsilon x.\bot$ and $\varepsilon x.\top$. The rule does not apply with εp since $p(\varepsilon p)$ is on the branch. The rule does not apply with $\varepsilon x.\bot$ since $\forall x.\neg\bot$ is on the branch. The rule

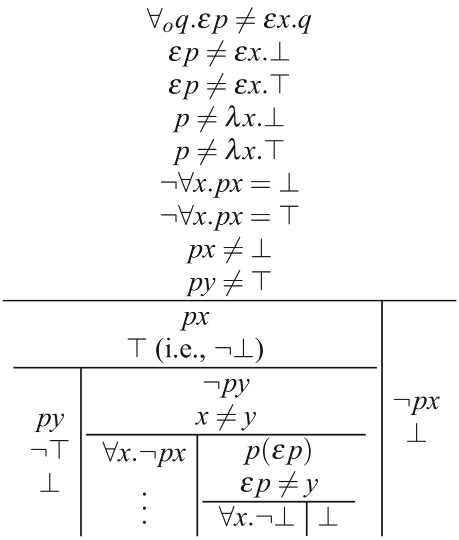

Fig. 3 A tableau with an evident branch

does not apply with $\varepsilon x.\top$ since $\top$ (the normal form of $(\lambda x.\top)(\varepsilon x.\top)$) is on the branch. By Corollary 1 the branch A is satisfiable.

5 Abstract Consistency and Completeness

We now lift the model existence theorem for evident sets to a model existence theorem for abstractly consistent sets. This will allow us to prove completeness of the tableau calculus $\mathcal{T}$. The use of abstract consistency to prove completeness was first used by Smullyan [29, 30] and later used by several authors in various higher-order settings [2, 9, 14, 21]. To prove completeness of the tableau calculus, it is enough to consider branches (finite sets of normal formulas) as in [6]. To obtain a more general result which will imply compactness and the existence of countable models, we also consider sets A of normal formulas which may be infinite.

A set Γ of sets of normal formulas is an *abstract consistency class* if it satisfies the conditions in Fig. 4 for every $A \in \Gamma$. We say Γ is *complete* if for every $A \in \Gamma$ and every formula $s \in \Lambda_o^{\mathcal{V}A}$ either $A \cup \{s\} \in \Gamma$ or $A \cup \{\neg s\} \in \Gamma$. As with evident sets, this property (without the restriction on free variables of s) was called "saturation" in earlier work [8, 9]. A strong connection between admissibility of cut in a sequent calculus and the existence of complete abstract consistency classes was shown in Theorems 3.5 and 3.8 in [8]. Indeed, Smullyan discusses the property in [30] and calls it the *cut* condition.

In Lemma 14 we will prove that every member of an abstract consistency class can be extended to an evident set. In order to verify the $\mathcal{E}_\forall$ condition we will need the following lemma relating universes for different sets of formulas.

Lemma 13 *Let $\mathscr{A}$ be a nonempty set of sets of normal formulas and let E be $\bigcup \mathscr{A}$. Suppose for every finite set $B \subseteq E$ there is some $A \in \mathscr{A}$ such that $B \subseteq A$. Then for every $t \in \mathscr{U}_\sigma^E$ there is some $A \in \mathscr{A}$ such that $t \in \mathscr{U}_\sigma^A$.*

$\mathscr{C}_\bot$	$\bot$ is not in A.
$\mathscr{C}_\neg$	If $\neg s$ is in A, then s is not in A.
$\mathscr{C}_{\neq}$	$s \neq_\iota s$ is not in A.
$\mathscr{C}_{\neg\neg}$	If $\neg\neg s$ is in A, then $A \cup \{s\}$ is in Γ.
$\mathscr{C}_\vee$	If $s \vee t$ is in A, then $A \cup \{s\}$ or $A \cup \{t\}$ is in Γ.
$\mathscr{C}_{\neg\vee}$	If $\neg(s \vee t)$ is in A, then $A \cup \{\neg s, \neg t\}$ is in Γ.
$\mathscr{C}_\forall$	If $\forall_\sigma s$ is in A, then $A \cup \{[st]\}$ is in Γ for every $t \in \mathscr{U}_\sigma^A$.
$\mathscr{C}_{\neg\forall}$	If $\neg\forall_\sigma s$ is in A, then $A \cup \{\neg[sw]\}$ is in Γ for some decomposable $w \in \mathscr{W}_\sigma$.
$\mathscr{C}_{\text{MAT}}$	If $ws_1 \dots s_n$ is in A and $\neg wt_1 \dots t_n$ is in A, then $n \geq 1$ and $A \cup \{s_i \neq t_i\}$ is in Γ for some $i \in \{1, \dots, n\}$.
$\mathscr{C}_{\text{DEC}}$	If $ws_1 \dots s_n \neq_\iota wt_1 \dots t_n$ is in A, then $n \geq 1$ and $A \cup \{s_i \neq t_i\}$ is in Γ for some $i \in \{1, \dots, n\}$.
$\mathscr{C}_{\text{CON}}$	If $s =_\iota t$ and $u \neq_\iota v$ are in A, then either $A \cup \{s \neq u, t \neq u\}$ or $A \cup \{s \neq v, t \neq v\}$ is in Γ.
$\mathscr{C}_{\text{BE}}$	If $s \neq_o t$ is in A, then either $A \cup \{s, \neg t\}$ or $A \cup \{\neg s, t\}$ is in Γ.
$\mathscr{C}_{\text{BQ}}$	If $s =_o t$ is in A, then either $A \cup \{s, t\}$ or $A \cup \{\neg s, \neg t\}$ is in Γ.
$\mathscr{C}_{\text{FE}}$	If $s \neq_{\sigma\tau} t$ is in A, then $A \cup \{\neg[\forall x. sx =_\tau tx]\}$ is in Γ for some $x \in \mathscr{V}_\sigma \setminus (\mathscr{V}s \cup \mathscr{V}t)$.
$\mathscr{C}_{\text{FQ}}$	If $s =_{\sigma\tau} t$ is in A, then $A \cup \{[\forall x. sx =_\tau tx]\}$ is in Γ for some $x \in \mathscr{V}_\sigma \setminus (\mathscr{V}s \cup \mathscr{V}t)$.
$\mathscr{C}_\varepsilon$	If $\varepsilon_\sigma s$ is accessible in A, then either $A \cup \{[s(\varepsilon s)]\}$ is in Γ or there is some $x \in \mathscr{V}_\sigma \setminus \mathscr{V}s$ such that $A \cup \{[\forall x. \neg sx]\}$ is in Γ.

Fig. 4 Abstract consistency conditions (must hold for every $A \in \Gamma$)

Proof Let $t \in \mathscr{U}_\sigma^E$ be given. If σ is o, then choose $A \in \mathscr{A}$ and note $\mathscr{U}_o^A = \{\bot, \neg\bot\} = \mathscr{U}_o^E$.

Suppose σ is ι. First assume E has no discriminating terms. In this case t must be $*$. We choose $A \in \mathscr{A}$ and note that $t \in \mathscr{U}_\sigma^A$ since A also has no discriminating terms. Next assume E has discriminating terms. In this case t is a discriminating term of E. There is some s such that $t \neq s$ or $s \neq t$ is in E. There is some $A \in \mathscr{A}$ such that $t \neq s$ or $s \neq t$ is in A. Clearly $t \in \mathscr{U}_\iota^A$ as desired.

Finally suppose σ is $\tau\mu$. Let X be $\mathscr{V}E$. We know t is normal and in Λ_σ^X. For each $x \in \mathscr{V}t$, choose some $s_x \in E$ such that $x \in \mathscr{V}s_x$. Since the set $\{s_x | x \in \mathscr{V}t\}$ is finite, there is some $A \in \mathscr{A}$ such that $s_x \in A$ for every $x \in \mathscr{V}t$. Hence $\mathscr{V}t \subseteq \mathscr{V}A$ and so t is in $\mathscr{U}_\sigma^A$. □

We can now prove the desired extension lemma.

Lemma 14 (Extension Lemma) *Let Γ be an abstract consistency class and $A \in \Gamma$. There is an evident set E such that $A \subseteq E$. Furthermore, if Γ is complete, then E is complete.*

Proof Let $u^0, u^1, \ldots$ be an enumeration of all normal formulas. We will construct a sequence $A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots$ of branches such that every $A_n \in \Gamma$. Let $A_0 := A$. We define A_{n+1} by cases. If there is no $B \in \Gamma$ such that $A_n \cup \{u_n\} \subseteq B$, then let $A_{n+1} := A_n$. Otherwise, choose some $B \in \Gamma$ such that $A_n \cup \{u_n\} \subseteq B$. We consider six subcases.

1. If u_n is of the form $\neg\forall_\sigma s$, then choose A_{n+1} to be $B \cup \{\neg[sw]\} \in \Gamma$ for some decomposable $w \in \mathscr{W}_\sigma$. This is possible since Γ satisfies $\mathscr{C}_{\neg\forall}$.
2. If u_n is of the form $s \neq_{\sigma\tau} t$, then choose A_{n+1} to be $B \cup \{\neg[\forall x.sx =_\tau tx]\} \in \Gamma$ for some $x \in \mathscr{V}_\sigma \setminus ([s] \cup [t])$. This is possible by $\mathscr{C}_{\text{FE}}$.
3. If u_n is of the form $s =_{\sigma\tau} t$, then choose A_{n+1} to be $B \cup \{[\forall x.sx =_\tau tx]\} \in \Gamma$ for some $x \in \mathscr{V}_\sigma \setminus ([s] \cup [t])$. This is possible by $\mathscr{C}_{\text{FQ}}$.
4. Suppose u_n is of the form $\mathscr{E}_1[\varepsilon_\sigma s] \neq_\iota \mathscr{E}_2[\varepsilon_\tau t]$ for elimination contexts $\mathscr{E}_1$ and $\mathscr{E}_2$. We define A_{n+1} according to the first of the following possibilities that applies.

   (a) Let A_{n+1} be $B \cup \{[s(\varepsilon s)], [t(\varepsilon t)]\}$ if it is in Γ.
   (b) Let A_{n+1} be $B \cup \{[\forall x.\neg sx], [t(\varepsilon t)]\}$ if it is in Γ.
   (c) Let A_{n+1} be $B \cup \{[s(\varepsilon s)], [\forall y.\neg ty]\}$ if it is in Γ.
   (d) Let A_{n+1} be $B \cup \{[\forall x.\neg sx], [\forall y.\neg ty]\}$ if it is in Γ.

   Applying $\mathscr{C}_\varepsilon$ twice, we know one of the four possibilities above must hold.
5. Suppose u_n is of the form $\mathscr{C}[\varepsilon_\sigma s]$ where $\mathscr{C}$ is an accessibility context, but the previous case does not apply. (Since the previous case does not apply, the accessibility context $\mathscr{C}$ is uniquely determined.) By $\mathscr{C}_\varepsilon$ either $B \cup \{[s(\varepsilon s)]\}$ is in Γ or there is some $x \in \mathscr{V}_\sigma \setminus \mathscr{V}s$ such that $B \cup \{[\forall x.\neg sx]\}$ is in Γ. If $B \cup \{[s(\varepsilon s)]\}$ is in Γ, then let A_{n+1} be $B \cup \{[s(\varepsilon s)]\}$. Otherwise, choose A_{n+1} to be $B \cup [\forall x.\neg sx] \in \Gamma$ for some $x \in \mathscr{V}_\sigma \setminus \mathscr{V}s$.
6. If no previous case applies, then let A_{n+1} be B.

Let $E := \bigcup_{n\in\mathbb{N}} A_n$. We prove E satisfies the evidence conditions.

$\mathscr{E}_\perp$ If $\perp$ is in E, then $\perp$ is in A_n for some n, contradicting $\mathscr{C}_\perp$.

$\mathscr{E}_\neg$ Assume s and $\neg s$ are both in E. Let r be such that $\{s, \neg s\} \subseteq A_r$. This contradicts $\mathscr{C}_\neg$.

$\mathscr{E}_{\neq}$ Assume $s \neq_\iota s$ is in E. There is some r such that $s \neq s$ is in A_r, contradicting $\mathscr{C}_{\neq}$.

$\mathscr{E}_{\neg\neg}$ Assume $\neg\neg s$ is in E. Let n be such that $u_n = s$. Let $r \geq n$ be such that $\neg\neg s$ is in A_r. By $\mathscr{C}_{\neg\neg}$, $A_r \cup \{s\} \in \Gamma$. Since $A_n \cup \{s\} \subseteq A_r \cup \{s\}$, we have $s \in A_{n+1} \subseteq E$.

$\mathscr{E}_\vee$ Assume $s \vee t$ is in E. Let n, m be such that $u_n = s$ and $u_m = t$. Let $r \geq n, m$ be such that $s \vee t$ is in A_r. By $\mathscr{C}_\vee$, $A_r \cup \{s\} \in \Gamma$ or $A_r \cup \{t\} \in \Gamma$. In the first case, $A_n \cup \{s\} \subseteq A_r \cup \{s\} \in \Gamma$, and so $s \in A_{n+1} \subseteq E$. In the second case, $A_m \cup \{t\} \subseteq A_r \cup \{t\} \in \Gamma$, and so $t \in A_{m+1} \subseteq E$. Hence either s or t is in E.

$\mathscr{E}_{\neg\vee}$ Assume $\neg(s \vee t)$ is in E. Let n, m be such that $u_n = \neg s$ and $u_m = \neg t$. Let $r \geq n, m$ be such that $\neg(s \vee t)$ is in A_r. By $\mathscr{C}_{\neg\vee}$, $A_r \cup \{\neg s, \neg t\} \in \Gamma$ and so $\neg s$ and $\neg t$ are in E.

$\mathscr{E}_\forall$ Assume $\forall_\sigma s$ is in E. Let $t \in \mathscr{U}_\sigma^E$ be a normal term. Let n be such that $u_n = [st]$. By Lemma 13 (taking $\mathscr{A}$ to be $\{A_r | r \geq n \text{ and } \forall_\sigma s \in A_r\}$) there is some $r \geq n$ such that $t \in \mathscr{U}_\sigma^{A_r}$ and $\forall_\sigma s$ is in A_r. By $\mathscr{C}_\forall$, $A_r \cup \{[st]\}$ is in Γ. Since $A_n \cup \{u_n\} \subseteq A_r \cup \{[st]\}$, we have $[st] = u_n \in A_{n+1} \subseteq E$.

$\mathscr{E}_{\neg\forall}$ Assume $\neg\forall_\sigma s$ is in E. Let n be such that $u_n = \neg\forall s$. Let $r \geq n$ be such that $\neg\forall s$ is in A_r. This A_r witnesses that there is some $B \in \Gamma$ such that $A_n \cup \{u_n\} \subseteq B$. By definition $\neg[sw] \in A_{n+1} \subseteq E$ for some $w \in \mathscr{W}_\sigma$.

$\mathscr{E}_{\text{MAT}}$ Assume $xs_1 \ldots s_n$ and $\neg xt_1 \ldots t_n$ are in E where $n \geq 1$. For each $i \in \{1, \ldots, n\}$, let m_i be such that u_{m_i} is $s_i \neq t_i$. Let $r \geq m_1, \ldots, m_n$ be such that $xs_1 \ldots s_n$ and $\neg xt_1 \ldots t_n$ are in A_r. By $\mathscr{C}_{\text{MAT}}$ there is some $i \in \{1, \ldots, n\}$ such that $A_r \cup \{s_i \neq t_i\} \in \Gamma$. Since $A_{m_i} \cup \{s_i \neq t_i\} \subseteq A_r \cup \{s_i \neq t_i\}$, we have $(s_i \neq t_i) \in A_{m_i+1} \subseteq E$.

$\mathscr{E}_{\text{DEC}}$ Similar to $\mathscr{E}_{\text{MAT}}$.

$\mathscr{E}_{\text{CON}}$ Assume $s =_\iota t$ and $u \neq_\iota v$ are in E. Let n, m, j, k be such that u_n is $s \neq u$, u_m is $t \neq u$, u_j is $s \neq v$ and u_k is $t \neq v$. Let $r \geq n, m, j, k$ be such that $s =_\iota t$ and $u \neq_\iota v$ are in A_r. By $\mathscr{C}_{\text{CON}}$ either $A_r \cup \{s \neq u, t \neq u\}$ or $A_r \cup \{s \neq v, t \neq v\}$ is in Γ. If $A_r \cup \{s \neq u, t \neq u\}$ is in Γ, then $s \neq u$ and $t \neq u$ are in E. If $A_r \cup \{s \neq v, t \neq v\}$ is in Γ, then $s \neq v$ and $t \neq v$ are in E.

$\mathscr{E}_{\text{BE}}$ Assume $s \neq_o t$ is in E. Let n, m, j, k be such that $u_n = s$, $u_m = t$, $u_j = \neg s$ and $u_k = \neg t$. Let $r \geq n, m, j, k$ be such that $s \neq_o t$ is in A_r. By $\mathscr{C}_{\text{BE}}$ either $A_r \cup \{s, \neg t\}$ or $A_r \cup \{\neg s, t\}$ is in Γ. If $A_r \cup \{s, \neg t\}$ is in Γ, then s and $\neg t$ are in E. If $A_r \cup \{\neg s, t\}$ is in Γ, then $\neg s$ and t are in E.

$\mathscr{E}_{\text{BQ}}$ Similar to $\mathscr{E}_{\text{BE}}$.

$\mathscr{E}_{\text{FE}}$ Assume $s \neq_{\sigma\tau} t$ is in E. Let n be such that u_n is $s \neq_{\sigma\tau} t$. Let $r \geq n$ be such that u_n is in A_r. Since $A_n \cup \{u_n\} \subseteq A_r$, by the definition of A_{n+1} there is some $x \in \mathscr{V}_\sigma \setminus (\mathscr{V}s \cup \mathscr{V}t)$ such that $[\neg\forall x.sx =_\tau tx]$ is in A_{n+1} and hence in E. By Proposition 7(1) there is some $w \in \mathscr{W}_\sigma$ such that $[sw] \neq_\tau [tw]$ is in E.

$\mathscr{E}_{\text{FQ}}$ Assume $s =_{\sigma\tau} t$ is in E and let $u \in \mathscr{U}_\sigma^E$ be given. Let n be such that u_n is $s =_{\sigma\tau} t$. Let $r \geq n$ be such that u_n is in A_r. This A_r witnesses that there is some $B \in \Gamma$ such that $A_n \cup \{u_n\} \subseteq B$. By the definition of A_{n+1} we know there is some $x \in \mathscr{V}_\sigma \setminus (\mathscr{V}s \cup \mathscr{V}t)$ such that $[\forall x.sx =_\tau tx]$ is in A_{n+1} and hence in E. By Proposition 7(2) we know $[su] \neq_\tau [tu]$ is in E.

$\mathscr{E}_\varepsilon$ Assume $\varepsilon_\sigma s$ is accessible in E. Then there is some accessibility context $\mathscr{C}$ such that $\mathscr{C}[\varepsilon_\sigma s]$ is in E. Let n be such that u_n is $\mathscr{C}[\varepsilon_\sigma s]$. Let $r \geq n$ be such that u_n is in A_r. By the definition of A_{n+1} either $[s(\varepsilon s)]$ is in A_{n+1} or $[\forall x.\neg(sx)]$ is in A_{n+1} for some $x \in \mathscr{V}_\sigma \setminus \mathscr{V}s$. In the first case we are done. In the second case let $x \in \mathscr{V}_\sigma \setminus \mathscr{V}s$ be such that $[\forall x.\neg(sx)]$ is in E. Let $t \in \mathscr{U}_\sigma^E$ be given. By Proposition 7(3) we know $\neg[st]$ is in E.

It remains to show that E is complete if Γ is complete. Let Γ be complete and s be a normal formula such that $\mathscr{V}s \subseteq \mathscr{V}E$. Since $\mathscr{V}s$ is a finite set, there is some k such that $\mathscr{V}s \subseteq \mathscr{V}(A_k)$. Let m, n be such that $u_m = s$ and $u_n = \neg s$. Consider $r \geq m, n, k$. Since Γ is complete, $A_r \cup \{s\}$ is in Γ or $A_r \cup \{\neg s\}$ is in Γ. If $A_r \cup \{s\}$ is in Γ, then $s \in E$. If $A_r \cup \{\neg s\}$ is in Γ, then $\neg s \in E$. □

Using the extension lemma we can lift the model existence theorem for evident sets to a model existence theorem for abstract consistency classes.

Theorem 2 (Model Existence) *Let Γ be an abstract consistency class. Every $A \in \Gamma$ is satisfiable. If Γ is complete, then every $A \in \Gamma$ has a countable model.*

Proof Let $A \in \Gamma$ be given. By Lemma 14 there is an evident set E such that $A \subseteq E$ such that E is complete if Γ is complete. We finish the proof with an appeal to Theorem 1. □

6 Completeness, Compactness and Countable Models

We can now prove completeness of the tableau calculus $\mathcal{T}$. Let $\Gamma_{\mathcal{T}}$ be the set of all branches A which are not refutable. We will first prove $\Gamma_{\mathcal{T}}$ is an abstract consistency class and then use Model Existence to prove completeness.

Lemma 15 *$\Gamma_{\mathcal{T}}$ is an abstract consistency class.*

Proof It is easy to check each condition in Fig. 4 using the corresponding tableau rule in $\mathcal{T}$. For example, we check $\mathcal{C}_{\varepsilon}$. Suppose $A \in \Gamma_{\mathcal{T}}$, $\varepsilon_{\sigma}s$ is accessible in A, $A \cup \{[s(\varepsilon s)]\}$ is not in $\Gamma_{\mathcal{T}}$ and $A \cup \{[\forall x.\neg(sx)]\}$ is not in $\Gamma_{\mathcal{T}}$ for every $x \in \mathcal{V}_{\sigma} \setminus \mathcal{V}s$. Choose some $x \in \mathcal{V}_{\sigma} \setminus \mathcal{V}s$. We know $A \cup \{[s(\varepsilon s)]\}$ and $A \cup \{[\forall x.\neg(sx)]\}$ are refutable. Hence A is refutable using $\mathcal{C}_{\varepsilon}$, contradicting $A \in \Gamma_{\mathcal{T}}$. □

Completeness now follows directly from Lemma 15 and Theorem 2.

Theorem 3 (Completeness) *Let A be a branch. If A is unsatisfiable, then A is refutable.*

We can also apply Theorem 2 to prove a combined form of the compactness theorem and the (downward) Löwenheim–Skolem theorem. Such a combination was proven for first-order logic in an analogous way in [29].

A set A of normal formulas is *sufficiently pure* if for each type σ the set $\mathcal{V}_{\sigma} \setminus \mathcal{V}A$ is infinite. In other words, A is sufficiently pure if there are infinitely many variables (of each type) that are not free in (any formula in) A.

Let Γ_{C} be the set of all sufficiently pure sets A of normal formulas such that every finite subset of A is satisfiable. The following lemma helps verify Γ_{C} is an abstract consistency class (cf. Lemma 10.1 in [14]).

Lemma 16 *Let $A \in \Gamma_{\mathrm{C}}$. If $B_1, \ldots, B_n$ are branches such that $A \cup B_i \notin \Gamma_{\mathrm{C}}$ for all $i \in \{1, \ldots, n\}$, then there is some finite $A' \subseteq A$ such that $A' \cup B_i$ is unsatisfiable for all $i \in \{1, \ldots, n\}$.*

Proof Consider $(C_1 \cup \cdots \cup C_n) \cap A$ where each C_i is an unsatisfiable finite subset of $A \cup B_i$. □

Lemma 17 *Γ_{C} is a complete abstract consistency class.*

Proof Most of the proof is the same as the proof of Lemma 10.2 in [14]. We show a two representative cases and leave the rest to the reader.

$\mathcal{C}_{\neg\forall}$ Suppose $\neg\forall_{\sigma}s$ is in $A \in \Gamma_{\mathrm{C}}$. Since A is sufficiently pure, there is some variable $x \in \mathcal{V}_{\sigma} \setminus \mathcal{V}A$. Note that x is decomposable. Assume $A \cup \{\neg[sx]\} \notin \Gamma_{\mathrm{C}}$. By Lemma 16 there is some finite $A' \subseteq A$ such that $A' \cup \{\neg[sx]\}$ is unsatisfiable. On the other hand, $A' \cup \{\neg\forall_{\sigma}s\}$ has a model $(\mathcal{D}, \mathcal{I})$ since it is a finite subset of A. By $\mathfrak{L}_{\forall_{\sigma}}(\mathcal{I}(\forall_{\sigma}))$ and $\mathfrak{L}_{\neg}(\mathcal{I}\neg)$ there is some $a \in \mathcal{D}_{\sigma}$ such that $\hat{\mathcal{I}}sa = 0$. We will prove $(\mathcal{D}, \mathcal{I}^{x}_{a})$ is a model of $A' \cup \{\neg[sx]\}$, giving a contradiction. By

Proposition 3 we know $(\mathcal{D}, \mathcal{I}^x_a)$ is a model of A' and that $\widehat{\mathcal{I}^x_a}(s) = \hat{\mathcal{I}}(s)$. Hence $\widehat{\mathcal{I}^x_a}(sx) = 0$. By N4 and $\mathfrak{L}_\neg(\mathcal{I}\neg)$ we are done.

$\mathcal{C}_\varepsilon$ Suppose $\varepsilon_\sigma s$ is accessible in $A \in \Gamma_C$. Choose some $x \in \mathcal{V}_\sigma \setminus \mathcal{V}s$. Assume neither $A \cup \{[s(\varepsilon s)]\}$ nor $A \cup \{[\forall x.\neg(sx)]\}$ is in Γ_C. By Lemma 16 there is some finite A' such that $A' \cup \{[s(\varepsilon s)]\}$ and $A' \cup \{[\forall x.\neg(sx)]\}$ are unsatisfiable. As a finite subset of A, A' has some model $(\mathcal{D}, \mathcal{I})$. By N4 and $\mathfrak{L}_{\varepsilon_\sigma}(\mathcal{I}(\varepsilon_\sigma))$, we must either have $\hat{\mathcal{I}}[s(\varepsilon s)] = 1$ (contradicting unsatisfiability of $A' \cup \{[s(\varepsilon s)]\}$) or for every $a \in \mathcal{D}_\sigma$ such that $\hat{\mathcal{I}}sa = 0$. In the latter case, it is easy to prove $\hat{\mathcal{I}}[\forall x.\neg(sx)] = 1$ (contradicting unsatisfiability of $A' \cup \{[\forall x.\neg(sx)]\}$) using $\mathfrak{L}_{\forall_\sigma}(\mathcal{I}(\forall_\sigma))$, $\mathfrak{L}_\neg(\mathcal{I}\neg)$ and Proposition 3. □

Theorem 4 (Compactness, Countable Models) *Let A be a set of formulas such that every finite subset of A is satisfiable. Then A has a countable model.*

We delay the proof. Note that if A is sufficiently pure, then we know there is a countable model of A by Lemma 17 and Theorem 2. In the remainder of this section we elaborate how to reduce the general case to the case in which A is sufficiently pure. A simple idea is to rename the variables free in A until it is sufficiently pure. We can rename the variables in such a way using an infinite substitution. The following lemma relates substitutions and interpretations and will be useful to prove Theorem 4.

Lemma 18 *Let $\mathcal{D}$ be a frame, $s \in \Lambda_\sigma$ be a term, θ be a substitution, $\mathcal{I}$ be an interpretation into $\mathcal{D}$ and $\mathcal{J}$ be an assignment into $\mathcal{D}$. Suppose $\mathcal{I}c = \mathcal{J}c$ for every logical constant c and $\hat{\mathcal{I}}(\hat{\theta}x) = \mathcal{J}x$ for every $x \in \mathcal{V}s$. Then $s \in$ Dom $\hat{\mathcal{J}}$ and $\hat{\mathcal{I}}(\hat{\theta}s) = \hat{\mathcal{J}}s$.*

Proof By induction on s. The base cases follow by assumption. If s is tu, then we compute

$$\hat{\mathcal{I}}(\hat{\theta}(tu)) \overset{\text{S2}}{=} (\hat{\mathcal{I}}(\hat{\theta}t))(\hat{\mathcal{I}}(\hat{\theta}u)) \overset{\text{IH}}{=} (\hat{\mathcal{J}}t)(\hat{\mathcal{J}}u) = \hat{\mathcal{J}}(tu).$$

Finally, suppose s is $\lambda y.u$ of type $\tau\mu$. We must prove $(\lambda y.u) \in$ Dom $\hat{\mathcal{J}}$ and $\hat{\mathcal{I}}(\hat{\theta}(\lambda y.u)) = \hat{\mathcal{J}}(\lambda y.u)$. Let $a \in \mathcal{D}_\tau$ be given. Let $z \in \mathcal{V}_\tau$ be a variable such that $z \notin \hat{\theta}(\lambda y.u)$ and $z \notin \mathcal{V}(\hat{\theta}x)$ for each $x \in \mathcal{V}(u) \setminus \{y\}$. By Proposition 3 and our choice of z we know

$$\widehat{\mathcal{I}^z_a}(\hat{\theta}(\lambda y.u)) = \hat{\mathcal{I}}(\hat{\theta}(\lambda y.u)) \text{ and } \widehat{\mathcal{I}^z_a}(\hat{\theta}x) = \hat{\mathcal{I}}(\hat{\theta}x) \text{ for all } x \in \mathcal{V}(u) \setminus \{y\}$$

We can apply the inductive hypothesis with θ^y_z, $\mathcal{I}^z_a$ and $\mathcal{J}^y_a$ since $\widehat{\mathcal{I}^z_a}(\widehat{\theta^y_z}y) = a = \mathcal{J}^y_a y$ and

$$\widehat{\mathcal{I}^z_a}(\widehat{\theta^y_z}x) = \widehat{\mathcal{I}^z_a}(\hat{\theta}x) = \hat{\mathcal{I}}(\hat{\theta}x) = \mathcal{J}x = \mathcal{J}^y_a x$$

for each $x \in \mathcal{V}u \setminus \{y\}$. Hence $u \in$ Dom $\widehat{\mathcal{J}^y_a}$ and

$$\hat{\mathcal{I}}(\hat{\theta}(\lambda y.u))a = \widehat{\mathcal{I}^z_a}(\hat{\theta}(\lambda y.u))a = \widehat{\mathcal{I}^z_a}(\hat{\theta}(\lambda y.u)z) \overset{\text{N4,S3}}{=} \widehat{\mathcal{I}^z_a}(\widehat{\theta^y_z}u) \overset{\text{IH}}{=} \widehat{\mathcal{J}^y_a}(u)$$

Generalizing over a, we know $(\lambda y.u) \in$ Dom $\hat{\mathcal{J}}$ and $\hat{\mathcal{I}}(\hat{\theta}(\lambda y.u))a = \hat{\mathcal{J}}(\lambda y.u)a$. □

Proof (Theorem 4) Since there are infinitely many variables of each type, we can find an infinite, injective substitution θ (with $\mathrm{Dom}\,\theta = \mathcal{V}$) such that $\hat{\theta}A$ is sufficiently pure (where $\hat{\theta}A := \{\hat{\theta}s | s \in A\}$). Since θ is injective, there is a substitution ψ such that $\psi(\theta x) = x$ for all $x \in \mathcal{V}$. Every finite subset of $\hat{\theta}A$ is of the form $\hat{\theta}B$ for some finite subset B of A. Let such a finite subset B be given. By assumption B has a model $(\mathcal{D}, \mathcal{J})$. Let $\mathcal{I}c := \mathcal{J}c$ for each logical constant c and $\mathcal{I}x := \hat{\mathcal{J}}(\hat{\psi}x)$ for each variable x. By Lemma 18 with the ψ as the substitution and with the roles of $\mathcal{I}$ and $\mathcal{J}$ reversed, we can conclude that $\mathcal{I}$ is an interpretation. Note that for each variable x we have $\mathcal{I}(\theta x) = \mathcal{J}(\psi(\theta x)) = \mathcal{J}x$. Let $\hat{\theta}s \in \hat{\theta}B$ be given. By Lemma 18 with θ, we know $\hat{\mathcal{I}}(\hat{\theta}s) = \hat{\mathcal{J}}s = 1$. Hence $(\mathcal{D}, \mathcal{I})$ is a model of $\hat{\theta}B$ and so $\hat{\theta}A$ is in Γ_C.

By Theorem 2 there is a countable model $(\mathcal{D}, \mathcal{I})$ of $\hat{\theta}A$. Let $\mathcal{J}c := \mathcal{I}c$ for each logical constant c and $\mathcal{J}x := \hat{\mathcal{I}}(\hat{\theta}x)$ for each variable x. By Lemma 18 we know $\mathcal{J}$ is an interpretation and for each $s \in A$ we know $\hat{\mathcal{J}}s = \hat{\mathcal{I}}(\hat{\theta}s) = 1$. Therefore, $(\mathcal{D}, \mathcal{J})$ is a countable model of A. □

7 Including If-Then-Else

We now extend the calculus to include an if-then-else operator if_σ of type $o\sigma\sigma\sigma$ for each type σ. This operator should satisfy the following formula:

$$\forall x_\sigma y_\sigma.(\mathsf{if}_\sigma \top xy = x) \wedge (\mathsf{if}_\sigma \bot xy = y) \tag{2}$$

A simple way to obtain such an if-then-else operator is to consider if_σ to be shorthand for the term $\lambda pxy.\varepsilon_\sigma z.p \wedge z = x \vee \neg p \wedge z = y$ and then reason using the tableau calculus $\mathcal{T}$.

An alternative is to consider each if_σ as a variable and include formulas of the form (2) on the branch to refute. The main problem with this approach is that the instantiation rule $\mathcal{T}_\forall$ applies to such formulas. Suppose σ is ιo and $\forall x_{\iota o} y_{\iota o}.(\mathsf{if}_{\iota o} \top xy = x) \wedge (\mathsf{if}_{\iota o} \bot xy = y)$ is on the branch A we wish to refute. Let s be a normal formula only using variables in $\mathcal{V}A$ and choose some $z \in \mathcal{V}_\iota \setminus \mathcal{V}A$. Since $[\lambda z.s] \in \mathcal{U}^A_{\iota o}$ we can apply $\mathcal{T}_\forall$ twice with $[\lambda z.s]$, followed by $\mathcal{T}_{\neg\vee}$ and $\mathcal{T}_{\neg\neg}$, to obtain $\mathsf{if}_{\iota o} \top [\lambda z.s][\lambda z.s] =_{\iota o} [\lambda z.s]$ on the branch. Choose some $t \in \mathcal{U}^A_\iota$. By Proposition 4 and S4 we know $[[\lambda z.s]t] = s$. Applying $\mathcal{T}_{\mathrm{FQ}}$ and then $\mathcal{T}_\forall$ with t we have $\mathsf{if}_{\iota o} \top [\lambda z.s][\lambda z.s]t =_o s$ on the branch. After applying $\mathcal{T}_{\mathrm{BE}}$ we see that we have reduced the problem of refuting A to the problem of refuting two branches extending A, one containing s and the other containing $\neg s$. That is, we have used the formula (2) at type ιo to simulate application of a cut rule with a formula s.

There are many examples of higher-order assumptions that allow one to simulate cut (see [8]). In such cases, one must somehow build the assumptions into the calculus itself in order to remain cut-free. In fact, this was one of the motivations for building the choice operator into the calculus $\mathcal{T}$.

With the above discussion in mind, we now give a tableau calculus $\mathcal{T}^{\mathsf{if}}$ extending $\mathcal{T}$ to include a rule for an if-then-else operator. We also prove its completeness.

7.1 Tableau Calculus and Evidence

For each type σ choose a variable $\mathsf{if}_\sigma \in \mathscr{V}_{o\sigma\sigma\sigma}$. Note that there are infinitely many variables in $\mathscr{V}_\sigma$ that are not chosen. From now on, when we speak of a variable being *fresh* we will also assume it is not one of the variables if_σ. Let T^{if} be the set of formulas of the form of (2) for each type σ. That is, $T^{\mathsf{if}} := \{\forall x_\sigma y_\sigma.(\mathsf{if}_\sigma \top xy = x) \wedge (\mathsf{if}_\sigma \bot xy = y) | \sigma \text{ type}\}$.

A model $(\mathscr{D}, \mathscr{I})$ is a T^{if}-*model* if it is a model of T^{if}. Suppose $(\mathscr{D}, \mathscr{I})$ is a T^{if}-model. Each $\mathscr{I}\mathsf{if}_\sigma$ must be a function $I \in \mathscr{D}_{o\sigma\sigma\sigma}$ such that $I1b_1b_0 = b_1$ and $I0b_1b_0 = b_0$ for every $b_1, b_0 \in \mathscr{D}_\sigma$. We call such an I an *if-then-else* function. Note that there is at most one if-then-else function in $\mathscr{D}_{o\sigma\sigma\sigma}$. Conversely, we know a model $(\mathscr{D}, \mathscr{I})$ is a T^{if}-model if every $\mathscr{I}\mathsf{if}_\sigma$ is an if-then-else function.

We define a tableau calculus $\mathscr{T}^{\mathsf{if}}$ by taking the union of $\mathscr{T}$ and the following rule:

$$\mathscr{T}_{\text{IF}} \quad \frac{\mathscr{C}[\mathsf{if}_\sigma stu]}{s,\ [\mathscr{C}[t]] \mid \neg s,\ [\mathscr{C}[u]]} \ \mathscr{C} \text{ accessibility context}$$

We say a set E is T^{if}-*evident* if it is evident and satisfies the following additional evidence condition:

$\mathscr{E}_{\text{IF}}$ If $\mathscr{C}[\mathsf{if}\, s\, t\, u]$ is in E and $\mathscr{C}$ is an accessibility context, then s and $[\mathscr{C}[t]]$ are in E or $\neg s$ and $[\mathscr{C}[u]]$ are in E.

7.2 Model Existence

We prove that every T^{if}-evident set E has a T^{if}-model. Let E be T^{if}-evident and X be $\mathscr{V}E$. Let $\|, \rhd, \mathscr{D}$ and Φ_σ be defined as in Section 4. The construction of a model is similar to the one in Section 4 except that we must choose the interpretations of the variables if_σ to obtain a T^{if}-model. Two lemmas suffice for this purpose.

Lemma 19 *For each type σ there is a term $s \in \Lambda^{\emptyset}_{o\sigma\sigma\sigma}$ and an if-then-else function $I \in \mathscr{D}_{o\sigma\sigma\sigma}$ such that $s \rhd I$.*

Proof By Lemma 12 (with the empty substitution and an arbitrary assignment) there is a substitution θ and a logical, admissible interpretation $\mathscr{I}$ such that $\hat{\theta}s \rhd \hat{\mathscr{I}}s$ for all $s \in \Lambda_\tau$. For each $s \in \Lambda^{\emptyset}_\tau$, $[\hat{\theta}s] = [s]$ by S4 and so $s \rhd \hat{\mathscr{I}}s$ by Lemma 1. Choose distinct variables $x \in \mathscr{V}_o$, $y_0, y_1, z \in \mathscr{V}_\sigma$. Let $s \in \Lambda^{\emptyset}_{o\sigma\sigma\sigma}$ be $\lambda xy_1y_0.\varepsilon_\sigma z.x \wedge z = y_1 \vee \neg x \wedge z = y_0$ and let I be $\hat{\mathscr{I}}s$. Clearly, $s \rhd I$. We need only check that I is an if-then-else function.

Let $b_1, b_0 \in \mathscr{D}_\sigma$ be given. It is easy to check that for each $i \in \{0, 1\}$ and $b \in \mathscr{D}_\sigma$

$$\widehat{\mathscr{I}^{x,y_1,y_0}_{i,b_1,b_0}}(\lambda z.x \wedge z = y_1 \vee \neg x \wedge z = y_0)b = 1 \text{ if and only if } b = b_i.$$

Thus $Iib_1b_0 = \Phi_\sigma(\widehat{\mathscr{I}^{x,y_1,y_0}_{i,b_1,b_0}}(\lambda z.x \wedge z = y_1 \vee \neg x \wedge z = y_0)) = b_i$. □

Lemma 20 *If $\mathsf{if}_\sigma \in X$, then there is an if-then-else function $I \in \mathscr{D}_{o\sigma\sigma\sigma}$ such that $\mathsf{if}_\sigma \rhd I$.*

Proof By Lemma 19 there is an if-then-else function $I \in \mathscr{D}_{o\sigma\sigma\sigma}$. We need only check that $\mathsf{if}_\sigma \rhd I$. Assume not. There must be terms $s, t, u, v_1, \cdots, v_n \in \Lambda^X$ and values $a, b, c, d_1, \ldots, d_n$ such that $\mathsf{if}_\sigma stuv_1 \cdots v_n \not\rhd_\beta Iabcd_1 \cdots d_n$ (for a base type β), $s \rhd a$, $t \rhd b$, $u \rhd c$, $v_1 \rhd d_1$, ..., $v_n \rhd d_n$. We also have $tv_1 \cdots v_n \rhd_\beta bd_1 \cdots d_n$ and $uv_1 \cdots v_n \rhd_\beta cd_1 \cdots d_n$. We can split into three cases: Either (1) $\beta = o$ and $Iabcd_1 \cdots d_n = 0$, or (2) $\beta = o$ and $Iabcd_1 \cdots d_n = 1$, or (3) $\beta = \iota$ and $Iabcd_1 \cdots d_n$ is a discriminant not containing the discriminating term $\mathsf{if}_\sigma[s][t][u][v_1] \cdots [v_n]$. In each case we can apply $\mathscr{E}_{\text{IF}}$ with an appropriately chosen context $\mathscr{C}$ and split into two subcases based on whether $[s]$ and $\mathscr{C}[[[t][v_1] \cdots [v_n]]]$ are in E or $\neg[s]$ and $\mathscr{C}[[[u][v_1] \cdots [v_n]]]$ are in E. In each subcase one can determine whether a is 0 or 1 and hence whether $Iabc$ is b or c. It is straightforward, though tedious, to check that each subcase yields a contradiction. □

Theorem 5 (Model Existence for T^{if}) *Every T^{if}-evident set E has a T^{if}-model.*

Proof We first define a substitution θ_0 with $\mathrm{Dom}\,\theta_0 = X \cup \mathscr{V}T$ and an assignment $\mathscr{I}_0$. For each $x \in X \setminus \mathscr{V}T$, let $\theta_0 x := x$ and $\mathscr{I}_0 x$ be such that $x \rhd \mathscr{I}_0 x$, which is possible by Lemmas 3(2) and 4. For each $\mathsf{if}_\sigma \in X \cap \mathscr{V}T$, let $\theta_0 \mathsf{if}_\sigma := \mathsf{if}_\sigma$ and $\mathscr{I}_0 \mathsf{if}_\sigma$ be the if-then-else function $I \in \mathscr{D}_{o\sigma\sigma\sigma}$ where $\mathsf{if}_\sigma \rhd I$, which is possible by Lemma 20. For each $\mathsf{if}_\sigma \in \mathscr{V}T \setminus X$, let $\theta_0 \mathsf{if}_\sigma$ be $s \in \Lambda^{\emptyset}_{o\sigma\sigma\sigma} \subseteq \Lambda^X_{o\sigma\sigma\sigma}$ and $\mathscr{I}_0 \mathsf{if}_\sigma$ be the if-then-else function $I \in \mathscr{D}_{o\sigma\sigma\sigma}$ where $s \rhd I$, which is possible by Lemma 19. By Lemma 12 there is a substitution θ and a logical, admissible interpretation $\mathscr{I}$ such that $\hat{\theta} s \rhd \hat{\mathscr{I}} s$ for all $s \in \Lambda_\sigma$, $\theta x = \theta_0 x$ and $\mathscr{I} x = \mathscr{I}_0 x$ for all $x \in X \cup \mathscr{V}T$. In particular, $\theta x = x$ for all $x \in X$. The fact that $(\mathscr{D}, \mathscr{I})$ is a model of E follows as in the proof of Theorem 1. Since $\mathscr{I} \mathsf{if}_\sigma = \mathscr{I}_0 \mathsf{if}_\sigma$ is an if-then-else function for every σ, we know $(\mathscr{D}, \mathscr{I})$ is a T^{if}-model. □

7.3 Completeness

A set Γ of sets of normal formulas is a T^{if}*-abstract consistency class* if it is an abstract consistency class and satisfies the following condition:

$\mathscr{C}_{\text{IF}}$ If $\mathscr{C}[\mathsf{if}\,s\,t\,u]$ is in A and $\mathscr{C}$ is an accessibility context, then $A \cup \{s, [\mathscr{C}[t]]\}$ is in Γ or $A \cup \{\neg s, [\mathscr{C}[u]]\}$ is in Γ.

Lemma 21 (Extension Lemma for T^{if}) *Let Γ be a T^{if}-abstract consistency class and $A \in \Gamma$. There is an T^{if}-evident set E such that $A \subseteq E$.*

Proof Recall the construction of E given in the proof of Lemma 14. We have an enumeration $u^0, u^1, \ldots$ of all normal formulas and define a sequence of $A_n \in \Gamma$ such that $A = A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots$ and then define E to be $\bigcup_n A_n$. We already know E is evident from Lemma 14. We need only check that $\mathscr{E}_{\text{IF}}$ holds. Suppose $\mathscr{C}$ is an accessibility context and $\mathscr{C}[\mathsf{if}\,s\,t\,u]$ is in E. Choose n, m, j, k such that u_n is s, u_m is $\neg s$, u_j is $[\mathscr{C}[t]]$ and u_k is $[\mathscr{C}[u]]$. Let $r \geq n, m, j, k$ be such that $\mathscr{C}[\mathsf{if}\,s\,t\,u]$ is in A_r. By $\mathscr{C}_{\text{IF}}$ either $A_r \cup \{u_n, u_j\}$ or $A_r \cup \{u_m, u_k\}$ is in Γ. Hence either u_n and u_j are in E or u_m and u_k are in E, as desired. □

Theorem 6 (Completeness of $\mathscr{T}^{\mathsf{if}}$) *Let A be a branch. If A is T^{if}-unsatisfiable, then A is $\mathscr{T}^{\mathsf{if}}$-refutable.*

Proof Let $\Gamma_{\mathscr{T}^{\mathsf{if}}}$ be the set of all branches which are not $\mathscr{T}^{\mathsf{if}}$-refutable. As with Lemma 15, it is easy to check that $\Gamma_{\mathscr{T}^{\mathsf{if}}}$ is a T^{if}-abstract consistency class. Assume A is not $\mathscr{T}^{\mathsf{if}}$-refutable. By Lemma 21 there is a T^{if}-evident set E such that $A \subseteq E$. By Theorem 5 there is a T^{if}-model of E. Hence A is T^{if}-satisfiable. □

8 Related Work

This work is an extension of two lines of research. First, we have extended the tableau calculus of Brown and Smolka [14] to support a choice operator and an if-then-else operator at every type. Second, we have obtained tighter restrictions on the instantiations of quantifiers than were available before.

In [12] Brown and Smolka give a complete tableau calculus for a first-order subsystem (EFO) of higher-order logic. Quantifiers are only allowed at type ι there and the instantiations are restricted to discriminating terms. We have maintained this restriction on instantiations for quantifiers at type ι. In addition we have proven that it is enough to instantiate quantifiers at type o with the two terms $\bot$ and $\top$. As for quantifiers at function types, we have proven that these instantiations need not consider variables that do not already occur free on the branch.

Mints gives sequent rules for choice in [24]. The choice rule given in this paper is similar to Mints' ε-rule. Our proof of Henkin-completeness was constructed by adapting the relevant parts of Mints' cut-elimination proof [24] to our setting. We briefly sketch a comparison between our rules and the rules of Mints.

Translating into our language, Mints' ε-rule could be represented as

$$(\textsc{Mints' } \varepsilon)\ \frac{}{[\neg(st)] \ \mid\ [s(\varepsilon s)]}\ \varepsilon s \text{ occurs on the branch}$$

By *εs occurs on the branch* we simply mean that εs appears as any subterm where none of the free variables of s are captured by a λ-binder. Note that this rule could apply more often than our $\mathscr{T}_\varepsilon$ rule. Our $\mathscr{T}_\varepsilon$ rule cannot be applied until εs appears on the branch in one of the forms $\varepsilon st_1 \cdots t_n$, $\neg(\varepsilon st_1 \cdots t_n)$, $(\varepsilon st_1 \cdots t_n) \neq_\iota u$ or $u \neq_\iota (\varepsilon st_1 \cdots t_n)$. Furthermore, in Mints' system the ε-rule would need to be applied for each new instantiation term t. In practice this could lead to the need to refute branches with $[s(\varepsilon s)]$ multiple times. We have avoided this by using the quantified formula $[\forall x.\neg(sx)]$ on the left branch.

Mints also includes an ε-extensionality rule in [24]. In our context, his rule could be realized as

$$(\textsc{Mints' ext } \varepsilon)\ \frac{}{s \neq t \ \mid\ (\varepsilon s) = (\varepsilon t)}\ \varepsilon_\sigma s \text{ and } \varepsilon_\sigma t \text{ occur on the branch}$$

In words, whenever $\varepsilon_\sigma s$ and $\varepsilon_\sigma t$ both occur on the branch, we must consider the case where s and t are different, and the case where εs and εt are the same. This rule could be highly branching in practice. When n different terms of the form εs occur on the branch, then the rule must be applied $\frac{n^2-n}{2}$ times. Furthermore, it has the disadvantage that it adds a positive equation to the branch. If σ is a function type, this will lead to the need to perform instantiations. We were able to omit such a rule entirely from our system and still prove completeness. It seems that Mints needed such a rule because the extensionality rule in [24] is not liberal enough. Translated into our context, the extensionality rule in [24] includes the rule

$$\text{(Special Case of Mints' extensionality)}\quad \frac{\varepsilon s s_1 \ldots s_n\,,\ \neg\varepsilon s t_1 \ldots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n}\quad n \geq 1$$

This corresponds to our mating rule, except that we have liberalized the rule to include the case when the corresponding first arguments of ε are different.

$$\text{(Special Case of } \mathcal{T}_{\text{MAT}}\text{)}\quad \frac{\varepsilon s_1 \ldots s_n\,,\ \neg\varepsilon t_1 \ldots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n}\quad n \geq 1$$

Combinations of λ-calculus and if-then-else operators have been considered before. Beeson [7] considered the unification problem for λ-calculus with a (slightly different) if-then-else operator. Altenkirch and Uustalu [1] study the simply typed λ-calculus with if-then-else as the elimination construct for the two element type.

The first author has considered choice operators, description operators and if-then-else operators in his Master's thesis [5]. Similar rules (using restrictions to accessible terms) can be used to incorporate description operators and a similar model construction (using discriminants and possible values) can be used to prove completeness.

9 Conclusion

We have presented a cut-free tableau calculus for Church's simple type theory with a choice operator. The calculus is designed with automated proof search in mind. In particular, only accessible terms on the branch need to be considered in order to apply a rule. Furthermore, instantiation terms are restricted according to the type and the formulas on the branch. At type o only instantiations corresponding to true and false are considered. At the base type ι only discriminating terms on the branch need to be considered (except when there are no discriminating terms in which case a default element can be used). Note that this means only finitely many instantiations at type ι need to be considered at each stage of the search. At function types, the set of instantiations is infinite, but we have at least proven that we do not need to consider instantiations with free variables that do not occur on the current branch. We have also given an extension of the calculus to include if-then-else operators.

The second author has implemented a higher-order automated theorem prover, Satallax, based on the ground calculus in this paper. Satallax encodes tableau steps of the ground calculus as propositional clauses and uses the SAT-solver MiniSat [16] to decide if there is a refutation using the steps considered so far. Satallax competed in the higher-order division of the CASC system competition [31]. Out of 200 problems, LEO-II [10] solved 125, Satallax solved 120, Isabelle [26] solved 101 and TPS [4] solved 80.

References

1. Altenkirch, T., Uustalu, T.: Normalization by evaluation for $\lambda^{\to 2}$. In: Kameyama, Y., Stuckey, P.J. (eds.) Functional and Logic Programming, 7th International Symposium, FLOPS 2004, Proceedings. LNCS, vol. 2998, pp. 260–275. Springer (2004)

2. Andrews, P.B.: Resolution in type theory. J. Symb. Log. **36**, 414–432 (1971)
3. Andrews, P.B.: General models and extensionality. J. Symb. Log. **37**, 395–397 (1972)
4. Andrews, P.B., Brown, C.E.: TPS: a hybrid automatic-interactive system for developing proofs. J. Appl. Logic **4**(4), 367–395 (2006)
5. Backes, J.: Tableaux for higher-order logic with if-then-else, description and choice. Master's thesis, Universität des Saarlandes (2010)
6. Backes, J., Brown, C.E.: Analytic tableaux for higher-order logic with choice. In: Giesl, J., Hähnle, R. (eds.) Automated Reasoning: 5th International Joint Conference, IJCAR 2010, Proceedings. LNCS/LNAI, vol. 6173, pp. 76–90. Springer (2010)
7. Beeson, M.: Unification in lambda-calculi with if-then-else. In: Kirchner, C., Kirchner, H. (eds.) Proceedings of the 15th International Conference on Automated Deduction. LNAI, vol. 1421, pp. 103–118. Springer, Lindau, Germany (1998)
8. Benzmueller, C., Brown, C.E., Kohlhase, M.: Cut-simulation and impredicativity. LMCS **5**(1:6), 1–21 (2009)
9. Benzmüller, C., Brown, C.E., Kohlhase, M.: Higher-order semantics and extensionality. J. Symb. Log. **69**, 1027–1088 (2004)
10. Benzmüller, C., Theiss, F., Paulson, L., Fietzke, A.: LEO-II — a cooperative automatic theorem prover for higher-order logic. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) Fourth International Joint Conference on Automated Reasoning, IJCAR'08. LNAI, vol. 5195. Springer (2008)
11. Brown, C.E.: Automated Reasoning in Higher-Order Logic: Set Comprehension and Extensionality in Church's Type Theory. College Publications (2007)
12. Brown, C.E., Smolka, G.: Extended first-order logic. In: Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Proceedings. LNCS, vol. 5674, pp. 164–179. Springer (2009)
13. Brown, C.E., Smolka, G.: Terminating tableaux for the basic fragment of simple type theory. In: Giese, M., Waaler, A. (eds.) Automated Reasoning with Analytic Tableaux and Related Methods: 18th International Conference, TABLEAUX 2009, Proceedings. LNCS/LNAI, vol. 5607, pp. 138–151. Springer (2009)
14. Brown, C.E., Smolka, G.: Analytic tableaux for simple type theory and its first-order fragment. LMCS **6**(2), 1–33 (2010)
15. Church, A.: A formulation of the simple theory of types. J. Symb. Log. **5**, 56–68 (1940)
16. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) Theory and Applications of Satisfiability Testing, LNCS, vol. 2919, pp. 333–336. Springer, Berlin/Heidelberg (2004)
17. Gordon, M., Melham, T.: Introduction to HOL: A Theorem-Proving Environment for Higher-Order Logic. Cambridge University Press (1993)
18. Harrison, J.: HOL light: a tutorial introduction. In: Srivas, M., Camilleri, A. (eds.) Proceedings of the First International Conference on Formal Methods in Computer-Aided Design (FM-CAD'96), LNCS, vol. 1166, pp. 265–269. Springer (1996)
19. Henkin, L.: Completeness in the theory of types. J. Symb. Log. **15**, 81–91 (1950)
20. Hintikka, K.J.J.: Form and content in quantification theory. Two papers on symbolic logic. Acta Philos. Fenn. **8**, 7–55 (1955)
21. Huet, G.P.: Constrained resolution: a complete method for higher order logic. PhD thesis, Case Western Reserve University (1972)
22. King, D.J., Arthan, R.D.: Development of practical verification tools. ICL Systems J. **11**(1) (1996)
23. Miller, D.A.: A compact representation of proofs. Stud. Log. **46**(4), 347–370 (1987)
24. Mints, G.: Cut-elimination for simple type theory with an axiom of choice. J. Symb. Log. **64**(2), 479–485 (1999)
25. Mitchell, J.C., Hoang, M., Howard, B.T.: Labeling techniques and typed fixed-point operators. In: Higher Order Operational Techniques in Semantics, pp. 137–174. Cambridge University Press, New York (1998)
26. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL — a proof assistant for higher-order logic. LNCS, vol. 2283. Springer (2002)
27. Prawitz, D.: Hauptsatz for higher order logic. J. Symb. Log. **33**, 452–457 (1968)
28. Slind, K., Norrish, M.: A brief overview of HOL4. In: Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics. LNCS, vol. 5170, pp. 28–32. Springer, Berlin, Heidelberg (2008)
29. Smullyan, R.M.: A unifying principle in quantification theory. Proc. Natl. Acad. Sci. U.S.A. **49**, 828–832 (1963)

30. Smullyan, R.M.: First-Order Logic. Springer (1968)
31. Sutcliffe, G.: The 5th IJCAR automated theorem proving system competition - CASC-J5. AI Commun. **24**(1), 75–89 (2011)
32. Sutcliffe, G., Benzmüller, C., Brown, C.E., Theiss, F.: Progress in the development of automated theorem proving for higher-order logic. In: Schmidt, R.A. (ed.) Automated Deduction - CADE-22. 22nd International Conference on Automated Deduction, Proceedings. LNCS, vol. 5663, pp. 116–130. Springer (2009)
33. Takahashi, M.: Simple type theory of gentzen style with the inference of extensionality. Proc. Jpn. Acad. **44**(2), 43–45 (1968)