Analytic Tableaux for Higher-Order Logic with Choice

Julian Backes, Chad E. Brown

Universität des Saarlandes

July 14, 2010

Introduction: (Some) Higher-Order Provers

Automated	Interactive
Isabelle-HOL	Isabelle-HOL
TPS	HOL family:
	HOL4
	HOL-Light
	ProofPower
Simple Type Theory	Even More Simple Types
+ Logical Constants	+ Logical Constants
	+ Extensionality
	+ Primitive Equality
	+ Choice
	+ Infinity

Andrews, Paulson, Nipkow, Gordon, . . .

Introduction: (Some) Higher-Order Provers

Automated	Interactive
Isabelle-HOL	Isabelle-HOL
TPS	HOL family:
LEO-II	HOL4
	HOL-Light
	ProofPower
Simple Type Theory	Even More Simple Types
+ Logical Constants	+ Logical Constants
+ Extensionality	+ Extensionality
+ Primitive Equality	+ Primitive Equality
	+ Choice
	+ Infinity

Andrews, Paulson, Nipkow, Gordon, ...

Kohlhase, Benzmüller, Brown, Smolka

Introduction: (Some) Higher-Order Provers

Automated	Interactive
Isabelle-HOL	Isabelle-HOL
TPS	HOL family:
LEO-II	HOL4
Satallax	HOL-Light
	ProofPower
Simple Type Theory	Even More Simple Types
+ Logical Constants	+ Logical Constants
+ Extensionality	+ Extensionality
+ Primitive Equality	+ Primitive Equality
+ Choice	+ Choice
	+ Infinity

Andrews, Paulson, Nipkow, Gordon, . . .
Kohlhase, Benzmüller, Brown, Smolka
Backes, Brown

Introduction: (Some) Higher-Order Provers

Automated	Interactive
Isabelle-HOL	Isabelle-HOL
TPS	HOL family:
LEO-II	HOL4
Satallax	HOL-Light
CASC THF	ProofPower
Simple Type Theory	Even More Simple Types
+ Logical Constants	+ Logical Constants
+ Extensionality	+ Extensionality
+ Primitive Equality	+ Primitive Equality
+ Choice	+ Choice
	+ Infinity

Andrews, Paulson, Nipkow, Gordon, . . .
Kohlhase, Benzmüller, Brown, Smolka
Backes, Brown

Introductory Example

Example 1: Let ε be a choice operator:

If p is nonempty, then εp is in p.

$$\forall px.px \longrightarrow p(\varepsilon p)$$

Introductory Example

Example 1: Let ε be a choice operator:

If p is nonempty, then εp is in p.

$$\forall px.px \rightarrow p(\varepsilon p)$$

Tableau Refutation of branch with

$$px$$
$$\neg p(\varepsilon p)$$

Introductory Example

Example 1: Let ε be a choice operator:

If p is nonempty, then εp is in p.

$$\forall px.px \longrightarrow p(\varepsilon p)$$

Tableau Refutation of branch with

$$px$$
$$\neg p(\varepsilon p)$$
$$\forall x.\neg px \quad \Big| \quad p(\varepsilon p)$$

Mints (JSL, 1999) gave sequent rules for choice.

Similar tableau rule: Split into p empty or $p(\varepsilon p)$.

Introductory Example

Example 1: Let ε be a choice operator:

If p is nonempty, then εp is in p.

$$\forall px.px \longrightarrow p(\varepsilon p)$$

Tableau Refutation of branch with

$$px$$

$$\neg p(\varepsilon p)$$

$$\forall x.\neg px \quad \Big| \quad p(\varepsilon p)$$

Mints (JSL, 1999) gave sequent rules for choice.

Similar tableau rule: Split into p empty or $p(\varepsilon p)$.

Introductory Example

Example 1: Let ε be a choice operator:

If p is nonempty, then εp is in p.

$$\forall p x. p x \longrightarrow p(\varepsilon p)$$

Tableau Refutation of branch with

$$px$$
$$\neg p(\varepsilon p)$$

$\forall x. \neg px$	$p(\varepsilon p)$
$\neg px$	

Mints (JSL, 1999) gave sequent rules for choice.

Similar tableau rule: Split into p empty or $p(\varepsilon p)$.

Introductory Example

Example 1: Let ε be a choice operator:

If p is nonempty, then εp is in p.

$$\forall px.px \longrightarrow p(\varepsilon p)$$

Tableau Refutation of branch with

$$px$$
$$\neg p(\varepsilon p)$$

$\forall x.\neg px$ $\neg px$	$p(\varepsilon p)$

Mints (JSL, 1999) gave sequent rules for choice.

Similar tableau rule: Split into p empty or $p(\varepsilon p)$.

Mints: Because εp is a subterm. Can we further restrict?

Outline

- Higher-Order Logic with Choice (ε)
- Higher-Order Tableau
- The Choice Rule
- Restrictions on Instantiations
- Examples

Higher Order Logic

Simple Types σ, τ:

o (propositions), ι (individuals), $\sigma \rightarrow \tau$ (functions)

Higher Order Logic

Simple Types σ, τ:

o (propositions), ι (individuals), $\sigma \rightarrow \tau$ (functions)

Variables x of each type.

Higher Order Logic

Simple Types σ, τ:

o (propositions), ι (individuals), $\sigma \to \tau$ (functions)

Variables x of each type.

Terms s, t :

$$x \mid c \mid (\lambda x.s) \mid (st)$$

Logical constants $c \ldots$ (next slide)

$[s]$ - normal form of s

Logical Constants

- $\bot : o$
- $\neg : o \longrightarrow o$
- $\vee : o \longrightarrow o \longrightarrow o$
- $=_\sigma : \sigma \longrightarrow \sigma \longrightarrow o$
- $\forall_\sigma : (\sigma \longrightarrow o) \longrightarrow o$

Logical Constants

- $\bot : o$
- $\neg : o \to o$
- $\vee : o \to o \to o$
- $=_\sigma : \sigma \to \sigma \to o$
- $\forall_\sigma : (\sigma \to o) \to o$
- $\varepsilon_\sigma : (\sigma \to o) \to \sigma$ - choice function on σ
  Notation: Write $\varepsilon x.s$ for $\varepsilon(\lambda x.s)$

Logical Constants

- $\bot : o$
- $\neg : o \to o$
- $\vee : o \to o \to o$
- $=_\sigma : \sigma \to \sigma \to o$
- $\forall_\sigma : (\sigma \to o) \to o$
- $\varepsilon_\sigma : (\sigma \to o) \to \sigma$ - choice function on σ
  Notation: Write $\varepsilon x.s$ for $\varepsilon(\lambda x.s)$
- $* : \iota$ - a default element of the nonempty type ι

Example 2

Example 2: Fix $p : \iota \to o$. Can we have $\forall_o q.\varepsilon p \neq \varepsilon x.q$?

Example 2

Example 2: Fix $p : \iota \to o$. Can we have $\forall_o q.\varepsilon p \neq \varepsilon x.q$?

εp must be different from $\varepsilon x.\bot$ and $\varepsilon x.\neg\bot$.

Example 2

Example 2: Fix $p : \iota \to o$. Can we have $\forall_o q.\varepsilon p \neq \varepsilon x.q$?

εp must be different from $\varepsilon x.\bot$ and $\varepsilon x.\neg\bot$.

Tableau procedure will decide whether or not this is satisfiable.

Higher Order Tableau

Brown, Smolka [LMCS 2010]
(Complete for Henkin models without Choice)

Higher Order Tableau

Brown, Smolka [LMCS 2010]
(Complete for Henkin models without Choice)
A few unsurprising rules...

$$\mathcal{T}_{\neg}\ \frac{s, \neg s}{\bot} \qquad \mathcal{T}_{\neq}\ \frac{s \neq_{\iota} s}{\bot} \qquad \mathcal{T}_{\neg\neg}\ \frac{\neg\neg s}{s} \qquad \mathcal{T}_{\vee}\ \frac{s \vee t}{s \mid t}$$

$$\mathcal{T}_{\neg\vee}\ \frac{\neg(s \vee t)}{\neg s, \neg t}$$

Higher Order Tableau

Brown, Smolka [LMCS 2010]
(Complete for Henkin models without Choice)
Mating and decomposition...

$$\mathcal{T}_{\text{MAT}} \quad \frac{\delta s\,,\ \neg\delta t}{s \neq t} \qquad\qquad \mathcal{T}_{\text{DEC}} \quad \frac{\delta s \neq_{\iota} \delta t}{s \neq t}$$

δ either a variable or an ε (also, for arity > 1)

Higher Order Tableau

Brown, Smolka [LMCS 2010]
(Complete for Henkin models without Choice)
...and more rules...extensionality, equality

$$\mathcal{T}_{\mathrm{CON}} \quad \frac{s =_\iota t \,,\, u \neq_\iota v}{s \neq u \,,\, t \neq u \mid s \neq v \,,\, t \neq v} \qquad \mathcal{T}_{\mathrm{BE}} \quad \frac{s \neq_o t}{s \,,\, \neg t \mid \neg s \,,\, t}$$

$$\mathcal{T}_{\mathrm{BQ}} \quad \frac{s =_o t}{s \,,\, t \mid \neg s \,,\, \neg t}$$

$$\mathcal{T}_{\mathrm{FE}} \quad \frac{s \neq_{\sigma\tau} t}{\neg[\forall x.sx = tx]} \quad x \notin \mathcal{V}s \cup \mathcal{V}t$$

$$\mathcal{T}_{\mathrm{FQ}} \quad \frac{s =_{\sigma\tau} t}{[\forall x.sx = tx]} \quad x \notin \mathcal{V}s \cup \mathcal{V}t$$

The Choice Rule

$$\mathcal{T}_\varepsilon \quad \frac{}{[\forall x.\neg(sx)] \;\mid\; [s(\varepsilon s)]} \; \varepsilon s \text{ accessible}, x \notin \mathcal{V}s$$

The Choice Rule

$$\mathcal{T}_\varepsilon \quad \frac{}{[\forall x.\neg(sx)] \;\mid\; [s(\varepsilon s)]} \; \varepsilon s \text{ accessible}, x \notin \mathcal{V}s$$

When is $\varepsilon_\sigma s$ accessible? Depends on σ

The Choice Rule

$$\mathcal{T}_\varepsilon \quad \frac{}{[\forall x.\neg(sx)] \;\mid\; [s(\varepsilon s)]} \; \varepsilon s \text{ accessible, } x \notin \mathcal{V}s$$

When is $\varepsilon_\sigma s$ accessible? Depends on σ

ι: $\varepsilon s \neq_\iota t$ or $t \neq_\iota \varepsilon s$ on the branch

The Choice Rule

$$\mathcal{T}_\varepsilon \quad \frac{}{[\forall x.\neg(sx)] \;\mid\; [s(\varepsilon s)]} \; \varepsilon s \text{ accessible, } x \notin \mathcal{V}s$$

When is $\varepsilon_\sigma s$ accessible? Depends on σ

ι: $\varepsilon s \neq_\iota t$ or $t \neq_\iota \varepsilon s$ on the branch

$\sigma_1 \to \cdots \to \sigma_n \to \iota$: $(\varepsilon s) u_1 \cdots u_n \neq_\iota t$ on the branch

The Choice Rule

$$\mathcal{T}_\varepsilon \quad \frac{}{[\forall x.\neg(sx)] \;\mid\; [s(\varepsilon s)]} \; \varepsilon s \text{ accessible, } x \notin \mathcal{V}s$$

When is $\varepsilon_\sigma s$ accessible? Depends on σ

ι: $\varepsilon s \neq_\iota t$ or $t \neq_\iota \varepsilon s$ on the branch

$\sigma_1 \to \cdots \to \sigma_n \to \iota$: $(\varepsilon s)u_1 \cdots u_n \neq_\iota t$ on the branch

o: εs or $\neg\varepsilon s$ on the branch

The Choice Rule

$$\mathcal{T}_\varepsilon \quad \frac{}{[\forall x.\neg(sx)] \;\mid\; [s(\varepsilon s)]} \; \varepsilon s \text{ accessible, } x \notin \mathcal{V}s$$

When is $\varepsilon_\sigma s$ accessible? Depends on σ

ι: $\varepsilon s \neq_\iota t$ or $t \neq_\iota \varepsilon s$ on the branch

$\sigma_1 \to \cdots \to \sigma_n \to \iota$: $(\varepsilon s)u_1 \cdots u_n \neq_\iota t$ on the branch

o: εs or $\neg\varepsilon s$ on the branch

$\sigma_1 \to \cdots \to \sigma_n \to o$: $(\varepsilon s)u_1 \cdots u_n$ or its negation on the branch

Restrictions on Instantiations

$$\mathcal{T}_{\forall} \quad \frac{\forall_\sigma s}{[st]} \quad t \in \mathcal{U}_\sigma$$

Restrict instantiations in the $\mathcal{T}_{\forall}$ rule based to $\mathcal{U}_\sigma$ on the type:

Restrictions on Instantiations

$$\mathcal{T}_\forall \quad \frac{\forall_\sigma s}{[st]} \quad t \in \mathcal{U}_\sigma$$

Restrict instantiations in the $\mathcal{T}_\forall$ rule based to $\mathcal{U}_\sigma$ on the type:

$\mathcal{U}_\iota$ Only terms s occurring as $s \neq t$ or $t \neq s$ on branch.
If there are none, use default $* : \iota$
Finitely many!

Restrictions on Instantiations

$$\mathcal{T}_\forall \quad \frac{\forall_\sigma s}{[st]} \quad t \in \mathcal{U}_\sigma$$

Restrict instantiations in the $\mathcal{T}_\forall$ rule based to $\mathcal{U}_\sigma$ on the type:

$\mathcal{U}_\iota$ Only terms s occurring as $s \neq t$ or $t \neq s$ on branch.
If there are none, use default $* : \iota$
Finitely many!

$\mathcal{U}_o$ Only $\bot$ and $\neg\bot$ (false and true)
Finitely many!

Restrictions on Instantiations

$$\mathcal{T}_\forall \quad \frac{\forall_\sigma s}{[st]} \quad t \in \mathcal{U}_\sigma$$

Restrict instantiations in the $\mathcal{T}_\forall$ rule based to $\mathcal{U}_\sigma$ on the type:

- $\mathcal{U}_\iota$ Only terms s occurring as $s \neq t$ or $t \neq s$ on branch.
  If there are none, use default $* : \iota$
  Finitely many!
- $\mathcal{U}_o$ Only $\bot$ and $\neg\bot$ (false and true)
  Finitely many!
- $\mathcal{U}_{\sigma\to\tau}$ Only normal terms using variables free on the branch
  Infinitely many, of course.

If only quantifiers at o and ι, the procedure sometimes terminates.

Example 1

$$px$$
$$\neg p(\varepsilon p)$$

Example 1

$$p x$$
$$\neg p(\varepsilon p)$$

Example 1

$$px$$
$$\neg p(\varepsilon p)$$
$$x \neq_\iota \varepsilon p$$

Example 1

$$px$$
$$\neg p(\varepsilon p)$$
$$x \neq_\iota \varepsilon p$$

Example 1

εp accessible - Choice Rule Activated

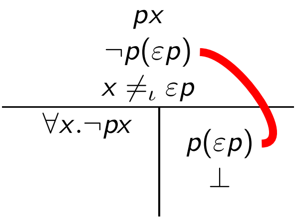
$$px$$
$$\neg p(\varepsilon p)$$
$$x \neq_\iota \varepsilon p$$

Example 1

$$px$$
$$\neg p(\varepsilon p)$$
$$x \neq_\iota \varepsilon p$$

$\forall x.\neg px$	$p(\varepsilon p)$

Example 1


px
$\neg p(\varepsilon p)$
$x \neq_\iota \varepsilon p$
$\forall x.\neg px$
$p(\varepsilon p)$
$\bot$

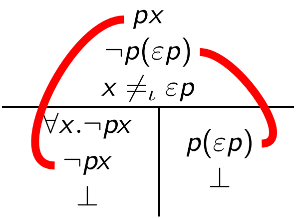
Example 1

x accessible - Legal Instantiation for $\forall$

$$px$$
$$\neg p(\varepsilon p)$$
$$x \neq_\iota \varepsilon p$$

$\forall x.\neg px$	$p(\varepsilon p)$
	$\bot$

Example 1



px
$\neg p(\varepsilon p)$
$x \neq_\iota \varepsilon p$
$\forall x. \neg px$
$\neg px$
$\bot$
$p(\varepsilon p)$
$\bot$

Example 2

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$

Example 2

Instantiate with $\bot$ and $\neg\bot$

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$

Example 2

Instantiate with $\bot$ and $\neg\bot$

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$

Example 2

Decompose

$$\forall_o q. \varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$

Example 2

Decompose

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$

Example 2

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$
$$\neg\forall x.px = \bot$$
$$\neg\forall x.px = \neg\bot$$

Example 2

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$
$$\neg\forall x.px = \bot$$
$$\neg\forall x.px = \neg\bot$$
$$px \neq \bot$$
$$py \neq \neg\bot$$

Example 2

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$
$$\neg\forall x.px = \bot$$
$$\neg\forall x.px = \neg\bot$$
$$px \neq \bot$$
$$py \neq \neg\bot$$

px $\neg\bot$	$\neg px$ $\bot$

Example 2

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$
$$\neg\forall x.px = \bot$$
$$\neg\forall x.px = \neg\bot$$
$$px \neq \bot$$
$$py \neq \neg\bot$$

px $\neg\bot$		$\neg px$ $\bot$
py $\neg\neg\bot$ $\bot$	$\neg py$	

Example 2

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$
$$\neg\forall x.px = \bot$$
$$\neg\forall x.px = \neg\bot$$
$$px \neq \bot$$
$$py \neq \neg\bot$$

<table>
<tr><td colspan="2">px
$\neg\bot$</td><td rowspan="2">$\neg px$
$\bot$</td></tr>
<tr><td>py
$\neg\neg\bot$
$\bot$</td><td>$\neg py$
$x \neq y$</td></tr>
</table>

Example 2

Accessible Choice Terms: εp, $\varepsilon x.\bot$, $\varepsilon x.\neg\bot$

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$
$$\neg\forall x.px = \bot$$
$$\neg\forall x.px = \neg\bot$$
$$px \neq \bot$$
$$py \neq \neg\bot$$

Branch 1:

$$px$$
$$\neg\bot$$

- Branch 1.1: py, $\neg\neg\bot$, $\bot$
- Branch 1.2: $\neg py$, $x \neq y$

Branch 2:

$$\neg px$$
$$\bot$$

Example 2

Accessible Choice Terms: εp, $\varepsilon x.\bot$, $\varepsilon x.\neg\bot$

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$
$$\neg\forall x.px = \bot$$
$$\neg\forall x.px = \neg\bot$$
$$px \neq \bot$$
$$py \neq \neg\bot$$

$$px$$
$$\neg\bot$$

$$py$$
$$\neg\neg\bot$$
$$\bot$$

$$\neg py$$
$$x \neq y$$

$$\forall x.\neg px$$
$$\vdots$$

$$p(\varepsilon p)$$
$$\varepsilon p \neq y$$

$$\forall x.\neg\bot$$

$$\bot$$

$$\neg px$$
$$\bot$$

Example 2

Evident (Hintikka) Set: Satisfiable

$$\forall_o q.\varepsilon p \neq \varepsilon x.q$$
$$\varepsilon p \neq \varepsilon x.\bot$$
$$\varepsilon p \neq \varepsilon x.\neg\bot$$
$$p \neq \lambda x.\bot$$
$$p \neq \lambda x.\neg\bot$$
$$\neg\forall x.px = \bot$$
$$\neg\forall x.px = \neg\bot$$
$$px \neq \bot$$
$$py \neq \neg\bot$$

Branch 1:

$$px$$
$$\neg\bot$$

Branch 1.1:

$$py$$
$$\neg\neg\bot$$
$$\bot$$

Branch 1.2:

$$\neg py$$
$$x \neq y$$

Branch 1.2.1:

$$\forall x.\neg px$$
$$\vdots$$

Branch 1.2.2:

$$p(\varepsilon p)$$
$$\varepsilon p \neq y$$

Branch 1.2.2.1:

$$\forall x.\neg\bot$$

Branch 1.2.2.2:

$$\bot$$

Branch 2:

$$\neg px$$
$$\bot$$

Completeness (1): Ideas

- Completeness: If a branch cannot be refuted, it is satisfiable by a Henkin model (with choice).

Completeness (1): Ideas

- Completeness: If a branch cannot be refuted, it is satisfiable by a Henkin model (with choice).
- In particular, if no more rules can be applied, the branch is satisfiable.

Completeness (1): Ideas

- Completeness: If a branch cannot be refuted, it is satisfiable by a Henkin model (with choice).
- In particular, if no more rules can be applied, the branch is satisfiable.
- Idea: Use a possible values relation (Prawitz 1968, Takahashi 1967+1968, Andrews 1971)

Completeness (1): Ideas

- Completeness: If a branch cannot be refuted, it is satisfiable by a Henkin model (with choice).
- In particular, if no more rules can be applied, the branch is satisfiable.
- Idea: Use a possible values relation (Prawitz 1968, Takahashi 1967+1968, Andrews 1971)
- and interpret ι by discriminants
  (compatible sets of discriminating terms - those used in disequations) - Brown, Smolka [LMCS 2010]

Completeness (1): Ideas

- Completeness: If a branch cannot be refuted, it is satisfiable by a Henkin model (with choice).
- In particular, if no more rules can be applied, the branch is satisfiable.
- Idea: Use a possible values relation (Prawitz 1968, Takahashi 1967+1968, Andrews 1971)
- and interpret ι by discriminants (compatible sets of discriminating terms - those used in disequations) - Brown, Smolka [LMCS 2010]
- and interpret Choice using a definition similar to Mints [JSL 1999]

Completeness (2): Possible Values

- Given: Branch A that is not refutable.
- Extend to E satisfying certain properties. ($A \subseteq E$)
- Let X be the free variables in E.
- Define $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$ by induction on types.

Completeness (2): Possible Values

- Given: Branch A that is not refutable.
- Extend to E satisfying certain properties. ($A \subseteq E$)
- Let X be the free variables in E.
- Define $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda^X_\sigma \times \mathcal{D}_\sigma$ by induction on types.
- $\mathcal{D}_o = \{0, 1\}$ (false and true)

Completeness (2): Possible Values

- Given: Branch A that is not refutable.
- Extend to E satisfying certain properties. ($A \subseteq E$)
- Let X be the free variables in E.
- Define $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$ by induction on types.
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- Interpret variables x such that $x \triangleright \mathcal{I}x$.

Completeness (2): Possible Values

- Given: Branch A that is not refutable.
- Extend to E satisfying certain properties. ($A \subseteq E$)
- Let X be the free variables in E.
- Define $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$ by induction on types.
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- Interpret variables x such that $x \triangleright \mathcal{I}x$.
- Interpret logical constants c appropriately and ensure $c \triangleright \mathcal{I}c$.

Completeness (2): Possible Values

- Given: Branch A that is not refutable.
- Extend to E satisfying certain properties. ($A \subseteq E$)
- Let X be the free variables in E.
- Define $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$ by induction on types.
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- Interpret variables x such that $x \triangleright \mathcal{I}x$.
- Interpret logical constants c appropriately and ensure $c \triangleright \mathcal{I}c$.
- Result will be a Henkin model of E.

Completeness (3): Forall

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)

Completeness (3): Forall

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- Interpretation of $\forall_\sigma$ is clear:

$$(\mathcal{I}\forall_\sigma)(f) = \begin{cases} 1 & \text{if } fa = 1 \text{ for all } a \\ 0 & \text{otherwise} \end{cases}$$

Completeness (3): Forall

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- Interpretation of $\forall_\sigma$ is clear:

$$(\mathcal{I}\forall_\sigma)(f) = \begin{cases} 1 & \text{if } fa = 1 \text{ for all } a \\ 0 & \text{otherwise} \end{cases}$$

- Easy to check $\forall_\sigma \triangleright \mathcal{I}\forall_\sigma$.

Completeness (4): Choice

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda^X_\sigma \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)

Completeness (4): Choice

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- To interpret ε_σ, let $f \in \mathcal{D}_{\sigma \to o}$.

Completeness (4): Choice

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- To interpret ε_σ, let $f \in \mathcal{D}_{\sigma \to o}$.
- First attempt:

$$(\mathcal{I}\varepsilon_\sigma)(f) = \begin{cases} \text{some } b & \text{such that } fb = 1 \text{ if such a } b \text{ exists.} \\ \text{some } a & \text{otherwise.} \end{cases}$$

Completeness (4): Choice

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda^X_\sigma \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- To interpret ε_σ, let $f \in \mathcal{D}_{\sigma\to o}$.
- First attempt:

$$(\mathcal{I}\varepsilon_\sigma)(f) = \begin{cases} \text{some } b & \text{such that } fb = 1 \text{ if such a } b \text{ exists.} \\ \text{some } a & \text{otherwise.} \end{cases}$$

- Problem: Cannot ensure $\varepsilon_\sigma \triangleright \mathcal{I}\varepsilon_\sigma$

Completeness (4): Choice

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda^X_\sigma \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- To interpret ε_σ, let $f \in \mathcal{D}_{\sigma\to o}$.
- Second attempt:

Completeness (4): Choice

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- To interpret ε_σ, let $f \in \mathcal{D}_{\sigma \to o}$.
- Second attempt:
- Define $f^0 := \{\varepsilon s \in \Lambda_\sigma^X | s \triangleright f$ and $\varepsilon[s]$ accessible in $E\}$.

Completeness (4): Choice

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda_\sigma^X \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- To interpret ε_σ, let $f \in \mathcal{D}_{\sigma\to o}$.
- Second attempt:
- Define $f^0 := \{\varepsilon s \in \Lambda_\sigma^X | s \triangleright f \text{ and } \varepsilon[s] \text{ accessible in } E\}$.
- Define

$$(\mathcal{I}\varepsilon_\sigma)(f) = \begin{cases} \text{some } b & \text{such that } fb = 1 \text{ if } f^0 \text{ is empty} \\ & \text{and such a } b \text{ exists.} \\ \text{some } a & \text{such that } \varepsilon s \triangleright a \text{ for every } \varepsilon s \in f^0. \end{cases}$$

Completeness (4): Choice

- $\mathcal{D}_\sigma$ and $\triangleright_\sigma \subseteq \Lambda^X_\sigma \times \mathcal{D}_\sigma$
- $\mathcal{D}_o = \{0, 1\}$ (false and true)
- To interpret ε_σ, let $f \in \mathcal{D}_{\sigma\to o}$.
- Second attempt:
- Define $f^0 := \{\varepsilon s \in \Lambda^X_\sigma | s \triangleright f$ and $\varepsilon[s]$ accessible in $E\}$.
- Define

$$(\mathcal{I}\varepsilon_\sigma)(f) = \begin{cases} \text{some } b & \text{such that } fb = 1 \text{ if } f^0 \text{ is empty} \\ & \text{and such a } b \text{ exists.} \\ \text{some } a & \text{such that } \varepsilon s \triangleright a \text{ for every } \varepsilon s \in f^0. \end{cases}$$

- $\mathcal{I}\varepsilon_\sigma$ is a choice function, and
- $\varepsilon_\sigma \triangleright \mathcal{I}\varepsilon_\sigma$.

Conclusion

- Directed, Cut-Free, Ground Tableau System for HOL (Brown, Smolka [LMCS 2010])
- Extended to include Choice (Backes, Brown [2010])
- Restricted Instantiations (Backes, Brown [2010])
- Similar techniques work for if-then-else and description (Backes [2010])
- Implementation: Satallax (Tableau + MiniSAT to search, competing in CASC)
- Can be used to determine both Unsatisfiability *and* (sometimes) Satisfiability.