Primitive recursive selection functions for existential assertions over abstract algebras

Jeffery Zucker (McMaster Univ.) and Thomas Strahm (Univ. Bern)

WPT, Münster, July, 2008

A classical result by Parsons, Mints and Takeuti

Let PA_1 denote Peano arithmetic with induction restricted to Σ^0_1 properties.

Theorem (Parsons, Mints, Takeuti)

Assume that for some Σ^0_1 formula P,

$$\mathsf{PA}_1 \vdash \forall \mathtt{x}\, \exists \mathtt{y}\, P(\mathtt{x}, \mathtt{y}).$$

Then for some primitive recursive function f*,*

$$\mathsf{PA}_1^{\mathsf{f}} \vdash \forall \mathtt{x}\, P(\mathtt{x}, \mathsf{f}(\mathtt{x})).$$

f *is called a selection function, realizing function or Skolem function.*

Aim of this work

- We want to generalize the Parson-Mints-Takeuti result to many-sorted abstract algebras, as part of an ongoing project of J. Zucker and J. Tucker on computation on abstract algebras.

Aim of this work

- We want to generalize the Parson-Mints-Takeuti result to many-sorted abstract algebras, as part of an ongoing project of J. Zucker and J. Tucker on computation on abstract algebras.
- Main difficulty encountered in carrying out this generalization: equality over these algebras may not be computable.

Aim of this work

- We want to generalize the Parson-Mints-Takeuti result to many-sorted abstract algebras, as part of an ongoing project of J. Zucker and J. Tucker on computation on abstract algebras.
- Main difficulty encountered in carrying out this generalization: equality over these algebras may not be computable.
- Solution: develop an appropriate concept of realizability of existential assertions over such algebras, generalized to *realizability of sequents of existential assertions*.

Aim of this work

- We want to generalize the Parson-Mints-Takeuti result to many-sorted abstract algebras, as part of an ongoing project of J. Zucker and J. Tucker on computation on abstract algebras.
- Main difficulty encountered in carrying out this generalization: equality over these algebras may not be computable.
- Solution: develop an appropriate concept of realizability of existential assertions over such algebras, generalized to *realizability of sequents of existential assertions*.
- In this way, the results can be seen to hold for classical proof systems.

Many-sorted abstract algebras

Given a signature Σ with finitely many *sorts* $s, \ldots$ and *function symbols*

$$\mathsf{F}\colon s_1 \times \cdots \times s_m \rightarrow s,$$

a Σ-algebra A consists of a carrier A_s for each Σ-sort s, and a total function

$$\mathsf{F}^A\colon A_{s_1} \times \cdots \times A_{s_m} \rightarrow A_s$$

for each Σ-function symbol F.

N-standard signatures and algebras

The signatures Σ and Σ-algebras A are said to be *N-standard* if they contain

N-standard signatures and algebras

The signatures Σ and Σ-algebras A are said to be *N-standard* if they contain

(a) the sort bool of *booleans* and the corresponding carrier $A_{\mathsf{bool}} = \mathbb{B} = \{\mathbb{t}, \mathbb{f}\}$, together with the standard boolean and boolean-valued operations, including the conditional at all sorts, and equality at certain sorts ("equality sorts");

N-standard signatures and algebras

The signatures Σ and Σ-algebras A are said to be *N-standard* if they contain

(*a*) the sort bool of *booleans* and the corresponding carrier $A_{\mathsf{bool}} = \mathbb{B} = \{\mathbb{t}, \mathbb{f}\}$, together with the standard boolean and boolean-valued operations, including the conditional at all sorts, and equality at certain sorts ("equality sorts");

(*b*) the sort nat of *natural numbers* and the corresponding carrier $A_{\mathsf{nat}} = \mathbb{N}$, together with the standard arithmetical operations of zero, successor, equality and order on $\mathbb{N}$.

Array signatures and algebras

Array signatures Σ^* and *array algebras* A^*, are formed from N-standard signatures Σ and algebras A by adding, for each sort s, an *array sort* s^*, with corresponding carrier A_s^* consisting of all arrays or finite sequences over A_s, with certain standard array operations.

Reason: Lack of effective coding in arbitrary data types.

$\mathbf{PR}(\mathbf{\Sigma})$ and $\mathbf{PR}^*(\mathbf{\Sigma})$ computation schemes

$\mathbf{PR}(\mathbf{\Sigma})$ and $\mathbf{PR}^*(\mathbf{\Sigma})$ computation schemes

(*i*) ***Primitive Σ-functions:***

$$\mathsf{f}(x) = \mathsf{F}(x)$$

$\mathbf{PR}(\mathbf{\Sigma})$ and $\mathbf{PR}^*(\mathbf{\Sigma})$ computation schemes

(*i*) ***Primitive Σ-functions:***

$$\mathsf{f}(x) \;=\; \mathsf{F}(x)$$

(*ii*) ***Projection:***

$$\mathsf{f}(\mathtt{x}) \;=\; \mathtt{x}_i$$

$\mathbf{PR}(\mathbf{\Sigma})$ and $\mathbf{PR}^*(\mathbf{\Sigma})$ computation schemes

(*i*) ***Primitive Σ-functions:***

$$\mathsf{f}(x) \;=\; \mathsf{F}(x)$$

(*ii*) ***Projection:***

$$\mathsf{f}(\mathtt{x}) \;=\; \mathtt{x}_i$$

(*iii*) ***Composition:***

$$\mathsf{f}(\mathtt{x}) \;=\; \mathsf{h}(\mathsf{g}_1(\mathtt{x}), \ldots, \mathsf{g}_m(\mathtt{x}))$$

$\mathbf{PR}(\mathbf{\Sigma})$ and $\mathbf{PR}^*(\mathbf{\Sigma})$ computation schemes

(*i*) ***Primitive Σ-functions:***

$$f(x) = F(x)$$

(*ii*) ***Projection:***

$$f(\mathtt{x}) = \mathtt{x}_i$$

(*iii*) ***Composition:***

$$f(\mathtt{x}) = h(g_1(\mathtt{x}), \ldots, g_m(\mathtt{x}))$$

(*iv*) ***Definition by cases:***

$$f(\mathtt{x}) = \begin{cases} g_1(\mathtt{x}) & \text{if } h(\mathtt{x}) = \mathbb{t} \\ g_2(\mathtt{x}) & \text{if } h(\mathtt{x}) = \mathbb{f} \end{cases}$$

$\mathbf{PR}(\boldsymbol{\Sigma})$ and $\mathbf{PR}^*(\boldsymbol{\Sigma})$ computation schemes

(*i*) ***Primitive Σ-functions:***

$$\mathsf{f}(x) \;=\; \mathsf{F}(x)$$

(*ii*) ***Projection:***

$$\mathsf{f}(\mathtt{x}) \;=\; \mathtt{x}_i$$

(*iii*) ***Composition:***

$$\mathsf{f}(\mathtt{x}) \;=\; \mathsf{h}(\mathsf{g}_1(\mathtt{x}), \ldots, \mathsf{g}_m(\mathtt{x}))$$

(*iv*) ***Definition by cases:***

$$\mathsf{f}(\mathtt{x}) \;=\; \begin{cases} \mathsf{g}_1(\mathtt{x}) & \text{if } \; \mathsf{h}(\mathtt{x}) = \mathbb{t} \\ \mathsf{g}_2(\mathtt{x}) & \text{if } \; \mathsf{h}(\mathtt{x}) = \mathbb{f} \end{cases}$$

(*v*) ***Simultaneous primitive recursion on $\mathbb{N}$:*** For $i = 1, \ldots, m$,

$$\begin{aligned} \mathsf{f}_i(0, \mathtt{x}) \;&=\; \mathsf{g}_i(\mathtt{x}) \\ \mathsf{f}_i(\mathsf{z}+1, \mathtt{x}) \;&=\; \mathsf{h}_i(\mathsf{z}, \mathtt{x}, \mathsf{f}_1(\mathsf{z}, \mathtt{x}), \ldots, \mathsf{f}_m(\mathsf{z}, \mathtt{x})) \end{aligned}$$

μPR computation schemes over Σ

Add to the PR schemes the scheme:

μPR computation schemes over Σ

Add to the PR schemes the scheme:

(*vi*) ***Least number*** or μ operator:

$$f(\mathrm{x}) \simeq \mu \mathrm{z}[g(\mathrm{x}, \mathrm{z}) = \mathbb{t}]$$

The interpretation of this is that $f^A(x) \downarrow z$ if, and only if, $g^A(x, y) \downarrow \mathbb{f}$ for each $y < z$ and $g^A(x, z) \downarrow \mathbb{t}$.

Generalization of Kleene partial recursive functions.

The language $Lang^*(\Sigma) = Lang(\Sigma^*)$

The language $\boldsymbol{Lang}^*(\Sigma) = \boldsymbol{Lang}(\Sigma^*)$

- *Atomic formulas* are *equations* between terms of the same sort, for all Σ-sorts, not just equality sorts!

UNIVERSITÄT BERN

The language $\boldsymbol{Lang}^*(\Sigma) = \boldsymbol{Lang}(\Sigma^*)$

- *Atomic formulas* are *equations* between terms of the same sort, for all Σ-sorts, not just equality sorts!
- A *BU (bounded universal) quantifier* is a quantifier of the form '$\forall \mathtt{k} < t$', where $\mathtt{k} : \mathsf{nat}$ and $t : \mathsf{nat}$.

The language $\boldsymbol{Lang}^*(\Sigma)= \boldsymbol{Lang}(\Sigma^*)$

- *Atomic formulas* are *equations* between terms of the same sort, for all Σ-sorts, not just equality sorts!
- A *BU (bounded universal) quantifier* is a quantifier of the form '$\forall \mathtt{k} < t$', where $\mathtt{k}$: nat and t : nat.
- A *BU equation* is an equation prefixed by BU quantifiers.

The language $\boldsymbol{Lang}^*(\Sigma) = \boldsymbol{Lang}(\Sigma^*)$

- *Atomic formulas* are *equations* between terms of the same sort, for all Σ-sorts, not just equality sorts!
- A *BU (bounded universal) quantifier* is a quantifier of the form '$\forall \mathtt{k} < t$', where $\mathtt{k}$: nat and t : nat.
- A *BU equation* is an equation prefixed by BU quantifiers.
- *Elementary formulas* are formed from equations by applying conjunctions, disjunctions, and BU quantifiers.

The language $\boldsymbol{Lang}^*(\Sigma) = \boldsymbol{Lang}(\Sigma^*)$

- *Atomic formulas* are *equations* between terms of the same sort, for all Σ-sorts, not just equality sorts!
- A *BU (bounded universal) quantifier* is a quantifier of the form '$\forall \mathtt{k} < t$', where $\mathtt{k}$: nat and t : nat.
- A *BU equation* is an equation prefixed by BU quantifiers.
- *Elementary formulas* are formed from equations by applying conjunctions, disjunctions, and BU quantifiers.
- $\boldsymbol{\Sigma_1^*}$ *formulas* are formed from equations by applying *conjunctions*, *disjunctions*, *BU quantification* and also (unbounded) *existential quantification* over any sort.

Defining the proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T)

In the following we will define a proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) for an abstract algebra A, where T is a suitable axiomatization of A.

Defining the proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T)

In the following we will define a proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) for an abstract algebra A, where T is a suitable axiomatization of A.

Assumption

The axioms of T are conditional BU equations, i.e., formulas of the form

$$Q_1 \wedge \cdots \wedge Q_n \rightarrow P$$

where Q_i and P are BU equations.

Correspondingly, a BU equantional sequent is a sequent of the form

$$Q_1, \ldots, Q_n \longmapsto P$$

where the Q_i and P are BU equations.

Defining the proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) (ctd.)

In the system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) we derive *classical* sequents $\Gamma \longmapsto \Delta$, where Γ and Δ are finite sequences of formulas of $\boldsymbol{Lang}^*(\Sigma)$.

Defining the proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) (ctd.)

In the system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) we derive *classical* sequents $\Gamma \longmapsto \Delta$, where Γ and Δ are finite sequences of formulas of $\boldsymbol{Lang}^*(\Sigma)$.

The system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) includes:

- Classical predicate calculus with equality (for signature Σ)

Defining the proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) (ctd.)

In the system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) we derive *classical* sequents $\Gamma \longmapsto \Delta$, where Γ and Δ are finite sequences of formulas of $\boldsymbol{Lang}^*(\Sigma)$.

The system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) includes:

- Classical predicate calculus with equality (for signature Σ)
- axioms for boolean operations and arrays

Defining the proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) (ctd.)

In the system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) we derive *classical* sequents $\Gamma \longmapsto \Delta$, where Γ and Δ are finite sequences of formulas of $\boldsymbol{Lang}^*(\Sigma)$.

The system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) includes:

- Classical predicate calculus with equality (for signature Σ)
- axioms for boolean operations and arrays
- Peano axioms for natural numbers

Defining the proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) (ctd.)

In the system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) we derive *classical* sequents $\Gamma \longmapsto \Delta$, where Γ and Δ are finite sequences of formulas of $\boldsymbol{Lang}^*(\Sigma)$.

The system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) includes:

- Classical predicate calculus with equality (for signature Σ)
- axioms for boolean operations and arrays
- Peano axioms for natural numbers
- $\mathbf{\Sigma_1^*}$ induction rule

$$\frac{\Gamma,\ P(\mathsf{a}) \ \longmapsto\ P(\mathsf{Sa}),\ \Delta}{\Gamma,\ P(0) \ \longmapsto\ P(t),\ \Delta}$$

Defining the proof system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) (ctd.)

In the system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) we derive *classical* sequents $\Gamma \longmapsto \Delta$, where Γ and Δ are finite sequences of formulas of $\boldsymbol{Lang}^*(\Sigma)$.

The system $\mathbf{\Sigma_1^*}$-Ind(Σ, T) includes:

- Classical predicate calculus with equality (for signature Σ)
- axioms for boolean operations and arrays
- Peano axioms for natural numbers
- $\mathbf{\Sigma_1^*}$ induction rule

$$\frac{\Gamma,\ P(\mathsf{a}) \ \longmapsto\ P(\mathsf{Sa}),\ \Delta}{\Gamma,\ P(0) \ \longmapsto\ P(t),\ \Delta}$$

- axioms of T as initial sequents

Selection theorem

Theorem (Selection theorem)

If $\mathbf{\Sigma}_1^*$-Ind(Σ, T) *proves the sequent*

$$\longmapsto \exists \mathrm{y} P(\mathrm{x}, \mathrm{y})$$

where $P(\mathrm{x}, \mathrm{y})$ *is an elementary formula, then there is a* PR^* *function* f *such that*

$$\longmapsto P(\mathrm{x}, \mathsf{f}(\mathrm{x}))$$

is provable in a suitable extension of $\mathbf{\Sigma}_1^*$-Ind(Σ, T) *(by defining equations for the function* f*).*

Main Lemma

Main Lemma (Tucker, Zucker, Leeds Proof Theory 1990)

Suppose that the $\mathbf{\Sigma_1^*}$ sequent

$$\exists z_1 Q_1(x, z_1),\ \ldots,\ \exists z_m Q_m(x, z_m) \ \longmapsto\ \exists y_1 P_1(x, y_1),\ \cdots,\ \exists y_n P_n(x, y_n)$$

is provable in $\mathbf{\Sigma_1^*}$-Ind(Σ, T). Then we can construct PR^* functions $f_1, \ldots, f_n$ such that

$$Q_1(x, z_1),\ \ldots,\ Q_m(x, z_m) \ \longmapsto\ P_1(x, f_1(x, z)),\ \cdots,\ P_n(x, f_n(x, z))$$

is provable.

Proof of Main Lemma

The proof proceeds by induction on the length of *quasi-cutfree derivations* (only $\mathbf{\Sigma_1^*}$ cuts). As expected, $\mathbf{\Sigma_1^*}$ induction uses PR^* functions for its interpretation.

Proof of Main Lemma

The proof proceeds by induction on the length of *quasi-cutfree derivations* (only $\boldsymbol{\Sigma_1^*}$ cuts). As expected, $\boldsymbol{\Sigma_1^*}$ induction uses PR^* functions for its interpretation.

Consider the case of contraction on the right hand side, Contr:R.

$$\frac{\ldots,\ \exists \mathrm{z}_j Q_j(\mathrm{x}, \mathrm{z}_j),\ \ldots \quad \longmapsto \quad \exists \mathrm{y} P(\mathrm{x}, \mathrm{y}),\ \exists \mathrm{y} P \mathrm{x}, \mathrm{y}),\ \ldots}{\ldots,\ \exists \mathrm{z}_j Q_j(\mathrm{x}, \mathrm{z}_j),\ \ldots \quad \longmapsto \quad \exists \mathrm{y} P(\mathrm{x}, \mathrm{y}), \ldots}\ .$$

Proof of Main Lemma

The proof proceeds by induction on the length of *quasi-cutfree derivations* (only $\mathbf{\Sigma_1^*}$ cuts). As expected, $\mathbf{\Sigma_1^*}$ induction uses PR^* functions for its interpretation.

Consider the case of contraction on the right hand side, Contr:R.

$$\frac{\ldots,\ \exists \mathrm{z}_j Q_j(\mathrm{x}, \mathrm{z}_j),\ \ldots \quad \longmapsto \quad \exists \mathrm{y} P(\mathrm{x}, \mathrm{y}),\ \exists \mathrm{y} P \mathrm{x}, \mathrm{y}),\ \ldots}{\ldots,\ \exists \mathrm{z}_j Q_j(\mathrm{x}, \mathrm{z}_j),\ \ldots \quad \longmapsto \quad \exists \mathrm{y} P(\mathrm{x}, \mathrm{y}), \ldots}\ .$$

By induction hypothesis there are PR^* functions $\mathsf{f}_1, \mathsf{f}_2$ such that

$$\ldots,\ Q_j(\mathrm{x}, \mathrm{z}_j),\ \ldots \longmapsto P(\mathrm{x}, \mathsf{f}_1(\mathrm{x}, \mathrm{z})),\ P(\mathrm{x}, \mathsf{f}_2(\mathrm{x}, \mathrm{z})),\ \ldots$$

is provable.

Proof of Main Lemma

The proof proceeds by induction on the length of *quasi-cutfree derivations* (only $\mathbf{\Sigma_1^*}$ cuts). As expected, $\mathbf{\Sigma_1^*}$ induction uses PR^* functions for its interpretation.

Consider the case of contraction on the right hand side, Contr:R.

$$\frac{\ldots,\ \exists \mathrm{z}_j Q_j(\mathrm{x},\mathrm{z}_j),\ \ldots\ \longmapsto\ \exists \mathrm{y} P(\mathrm{x},\mathrm{y}),\ \exists \mathrm{y} P\mathrm{x},\mathrm{y}),\ \ldots}{\ldots,\ \exists \mathrm{z}_j Q_j(\mathrm{x},\mathrm{z}_j),\ \ldots\ \longmapsto\ \exists \mathrm{y} P(\mathrm{x},\mathrm{y}),\ldots}\ .$$

By induction hypothesis there are PR^* functions $\mathsf{f}_1, \mathsf{f}_2$ such that

$$\ldots,\ Q_j(\mathrm{x},\mathrm{z}_j),\ \ldots\ \longmapsto\ P(\mathrm{x},\mathsf{f}_1(\mathrm{x},\mathrm{z})),\ P(\mathrm{x},\mathsf{f}_2(\mathrm{x},\mathrm{z})),\ \ldots$$

is provable. So define the PR function

$$\mathsf{f}(\mathrm{x},\mathrm{z}) \ = \ \begin{cases} \mathsf{f}_1(\mathrm{x},\mathrm{z}) & \text{if}\ \ P(\mathrm{x},\mathsf{f}_1(\mathrm{x},\mathrm{z})) \\ \mathsf{f}_2(\mathrm{x},\mathrm{z}) & \text{otherwise} \end{cases}$$

using *definition by cases*. Then f is a selection function for $\exists \mathrm{y} P$ in the conclusion.

PR decidability of equality

PR decidability of equality

The above case distincition uses primitive recursive decidability of elementary formulas!

A similar situation arises with the rules $\wedge$R and $\vee$L.

This assumption was needed in the Selection Theorem in [Tucker, Zucker: Proof Theory, Leeds, 1990].

However, many important algebras do not have decidable equality.

Attempted solution

Use intuitionistic instead of classical sequent calculus.

Reformulate selection theorem

Attempted solution

Use intuitionistic instead of classical sequent calculus.

Reformulate selection theorem

- without assuming decidability of equality

Attempted solution

Use intuitionistic instead of classical sequent calculus.

Reformulate selection theorem

- without assuming decidability of equality
- using intuitionistic version of $\mathbf{\Sigma_1^*}$-Ind(Σ, T)

Attempted solution

Use intuitionistic instead of classical sequent calculus.

Reformulate selection theorem

- without assuming decidability of equality
- using intuitionistic version of $\mathbf{\Sigma_1^*}$-Ind(Σ, T)

Attempted solution

Use intuitionistic instead of classical sequent calculus.

Reformulate selection theorem

- without assuming decidability of equality
- using intuitionistic version of $\mathbf{\Sigma_1^*}$-Ind(Σ, T)

The problem persists with the rule $\vee$L!

A way out is to use realizability and PR realizers instead of PR selectors (but still in an intuitionistic setting).

Realizability

Definition (Realizability of $\mathbf{\Sigma_1^*}$ formulas)

Let t be a Σ^*-term tuple, and P a $\mathbf{\Sigma_1^*}$ formula. We define the expression '$t \rhd P$' ("t realizes P") by structural induction on P:

(*i*) $t \rhd (t_1 = t_2) \quad \equiv \quad t_1 = t_2$

(*ii*) $\langle t_1, t_2 \rangle \rhd (P_1 \wedge P_2) \quad \equiv \quad (t_1 \rhd P_1) \wedge (t_2 \rhd P_2)$

(*iii*) $\langle t_0, t_1, t_2 \rangle \rhd (P_1 \vee P_2) \quad \equiv \quad (t_0 = \mathrm{true} \wedge t_1 \rhd P_1) \vee (t_0 = \mathrm{false} \wedge t_2 \rhd P_2)$

(*iv*) $t^* \rhd (\forall \mathrm{z} < t_0\, P) \quad \equiv \quad \forall \mathrm{z} < t_0 (t^*[\mathrm{z}] \rhd P)$

(*v*) $\langle t_0, t \rangle \rhd (\exists \mathrm{y} P) \quad \equiv \quad t \rhd P\langle \mathrm{y}/t_0 \rangle.$

UNIVERSITÄT BERN

Selection theorem: intuitionistic case

Main Lemma (Zucker, CiE 2006)

Suppose the $\mathbf{\Sigma_1^*}$ sequent

$$Q_1, \ldots, Q_m \quad \longmapsto \quad P$$

is provable in intuitionistic $\mathbf{\Sigma_1^*}$-$\mathrm{Ind_i}(\Sigma, T)$. Then for some tuple of PR functions f,

$$\mathrm{z}_1 \rhd Q_1,\ \ldots,\ \mathrm{z}_m \rhd Q_m \quad \longmapsto \quad \mathrm{f}(\mathrm{x}, \mathrm{z}_1, \ldots, \mathrm{z}_m) \rhd P$$

is provable.

The $\vee$L rule is no longer a problem in the setting!

Selection theorem: intuitionistic case

Main Lemma (Zucker, CiE 2006)

Suppose the $\mathbf{\Sigma}_1^*$ sequent

$$Q_1, \ldots, Q_m \quad \longmapsto \quad P$$

is provable in intuitionistic $\mathbf{\Sigma}_1^*$-$\mathsf{Ind_i}(\Sigma, T)$. Then for some tuple of PR functions f,

$$\mathrm{z}_1 \rhd Q_1, \; \ldots, \; \mathrm{z}_m \rhd Q_m \quad \longmapsto \quad \mathrm{f}(\mathrm{x}, \mathrm{z}_1, \ldots, \mathrm{z}_m) \rhd P$$

is provable.

The $\vee$L rule is no longer a problem in the setting!

Hence, we have a selection theorem

- w/o assuming decidable equality
- but with intuitionistic logic

The classical case

Is the Selection Theorem w/o the computable equality assumption but with classical logic true ?

The classical case

Is the Selection Theorem w/o the computable equality assumption but with classical logic true ?

Here is a proposed counterexample. Consider the algebra $\mathcal{R}$ of reals and the quantifier-free formula

$$P(\mathrm{x},\mathrm{y}) \quad \equiv_{df} \quad (\mathrm{x} \neq 0 \wedge \mathrm{y} = 0) \vee (\mathrm{x} = 0 \wedge \mathrm{y} = 1)$$

where x, y : real. Then

$$\forall \mathrm{x}\, \exists \mathrm{y}\, P(\mathrm{x},\mathrm{y})$$

is classically true and easily provable classically. But the (unique) selection function for this is *not continuous* on $\mathbb{R}$, and hence not PR^* computable on $\mathcal{R}$.

UNIVERSITÄT BERN

The classical case

Is the Selection Theorem w/o the computable equality assumption but with classical logic true ?

Here is a proposed counterexample. Consider the algebra $\mathcal{R}$ of reals and the quantifier-free formula

$$P(\mathrm{x},\mathrm{y}) \quad \equiv_{df} \quad (\mathrm{x} \neq 0 \wedge \mathrm{y} = 0) \vee (\mathrm{x} = 0 \wedge \mathrm{y} = 1)$$

where x,y : real. Then

$$\forall \mathrm{x}\, \exists \mathrm{y}\, P(\mathrm{x},\mathrm{y})$$

is classically true and easily provable classically. But the (unique) selection function for this is *not continuous* on $\mathbb{R}$, and hence not PR^* computable on $\mathcal{R}$.

Note, however, that P has a *negated equality*, and is therefore *not* elementary, according to our definition, or even $\mathbf{\Sigma_1^*}$!

Realizability extended

Solution: *Extend* the concept of realizability to realizability of *sequents*. Given a sequent

$$\Delta \;\equiv\; P_1, \ldots, P_n$$

of product type $u = u_1 \times \cdots \times u_n$, and a Σ^*-term tuple

$$\bar{r} \;=\; \langle r_0, r_1, \ldots, r_n \rangle$$

of "matching" type $\mathsf{nat} \times u_1 \times \cdots \times u_n$, we define

$$\bar{r} \;\rhd\!\rhd\; \Delta \qquad (\text{“}\,\bar{r} \text{ realizes } \Delta\,\text{”})$$

to mean

$$(r_0 = 1 \wedge r_1 \rhd P_1) \vee (r_0 = 2 \wedge r_2 \rhd P_2) \vee \ldots \vee (r_0 = n \wedge r_n \rhd P_n)$$

Selection theorem: classical case

Main Lemma (Zucker, Strahm (JLAP, to appear))

Suppose the $\mathbf{\Sigma_1^*}$ sequent

$$Q_1, \ldots, Q_m \quad \longmapsto \quad P_1, \ldots, P_n$$

is provable in $\mathbf{\Sigma_1^*}$-Ind(Σ, T). Then for some tuple of PR^* functions f,

$$\mathrm{z}_1 \triangleright Q_1, \; \ldots, \; \mathrm{z}_m \triangleright Q_m \quad \longmapsto \quad \mathrm{f}(\mathrm{x}, \mathrm{z}_1, \ldots, \mathrm{z}_m) \triangleright\!\triangleright (P_1, \ldots, P_n)$$

is provable.

The $\vee$L, Contr:R, and $\wedge$R rules are no longer a problem in this setting!

Selection theorem: classical case

Main Lemma (Zucker, Strahm (JLAP, to appear))

Suppose the $\mathbf{\Sigma}_1^*$ sequent

$$Q_1, \dots, Q_m \longmapsto P_1, \dots, P_n$$

is provable in $\mathbf{\Sigma}_1^*$-Ind(Σ, T). Then for some tuple of PR^* functions f,

$$\mathrm{z}_1 \rhd Q_1, \ \dots, \ \mathrm{z}_m \rhd Q_m \longmapsto \mathrm{f}(\mathrm{x}, \mathrm{z}_1, \dots, \mathrm{z}_m) \rhd\!\rhd (P_1, \dots, P_n)$$

is provable.

The $\vee$L, Contr:R, and $\wedge$R rules are no longer a problem in this setting!

Hence, we have a selection theorem

- w/o assuming decidable equality
- with classical logic

Contraction

The case of contraction on the right hand side, Contr:R:

$$\frac{\Gamma \quad \longmapsto \quad P, P, \Delta}{\Gamma \quad \longmapsto \quad P, \quad \Delta}$$

By induction hypothesis, there is a PR^* scheme tuple f such that

$$\mathrm{z} \;\rhd\; \Gamma \quad \longmapsto \quad \mathrm{f}(\mathrm{x}, \mathrm{z}) \;\rhd\!\rhd\; P, P, \Delta$$

is provable. Put $\mathrm{f}(\mathrm{x}, \mathrm{z}) = \langle r_0, r_1, r_2, \bar{r} \rangle$; $r_0 : \mathsf{nat}$, $r_1 : v$, $r_2 : v$ and $\bar{r} : w$ represent PR^* functions applied to x,z. Construct a PR^* tuple g with

$$\mathrm{g}(\mathrm{x}, \mathrm{z}) \;=\; \langle r_0', r_1', \bar{r} \rangle$$

where

$$r_0' \;=\; \begin{cases} 1 & \text{if} \;\; r_0 = 1 \vee r_0 = 2 \\ r_0 - 1 & \text{if} \;\; r_0 > 2 \end{cases}$$

and

$$r_1' \;=\; \begin{cases} r_1 & \text{if} \;\; r_0 = 1 \\ r_2 & \text{if} \;\; r_0 = 2 \\ \text{arbitrary} & \text{if} \;\; r_0 > 2. \end{cases}$$

Induction

The case of $\mathbf{\Sigma_1^*}$ *induction*:

$$\frac{\Gamma,\ P(\mathrm{a}) \ \longmapsto\ P(\mathrm{Sa}),\ \Delta}{\Gamma,\ P(0) \ \longmapsto\ P(t),\ \ \Delta}$$

By induction hypothesis there is a PR scheme f such that

$$\mathrm{z} \triangleright \Gamma,\ \mathrm{z}_0 \triangleright P(\mathrm{a}) \quad\longmapsto\quad \mathrm{f}(\mathrm{x},\mathrm{a},\mathrm{z},\mathrm{z}_0) \triangleright\!\triangleright P(\mathrm{Sa}), \Delta.$$

Put

$$\mathrm{f}(\mathrm{x},\mathrm{a},\mathrm{z},\mathrm{z}_0) \ =\ \langle r_0(\mathrm{a},\mathrm{z}_0),\ r_1(\mathrm{a},\mathrm{z}_0),\ r_2(\mathrm{a},\mathrm{z}_0),\ \ldots\rangle,$$

Now we construct a scheme g such that

$$\mathrm{g}(\mathrm{x},\mathrm{z},\mathrm{z}_0) \ =\ \langle r_0'(t,\mathrm{z}_0),\ r_1'(t,\mathrm{z}_0),\ r_2'(t,\mathrm{z}_0),\ \ldots\rangle$$

where the realizers $r_0', r_1', r_2', \ldots$ are defined by *simultaneous primitive recursion*:

Induction (ctd.)

Base case:

$$\begin{aligned} r'_i(0, \mathrm{z}_0) &= r_i(0, \mathrm{z}_0) \qquad \text{for } i \neq 1 \\ r'_1(0, \mathrm{z}_0) &= \mathrm{z}_0. \end{aligned}$$

Recursion step: For all $i = 0, 1, 2, \ldots$:

$$r'_i(\mathtt{n}+1, \mathrm{z}_0) = \begin{cases} r'_i(\mathtt{n}, \mathrm{z}_0) & \text{if } \; r'_0(\mathtt{n}, \mathrm{z}_0) > 1 \\ r_i(\mathtt{n}, \; r'_1(\mathtt{n}, \mathrm{z}_0)) & \text{if } \; r'_0(\mathtt{n}, \mathrm{z}_0) = 1 \end{cases}$$

As soon as the index points to a realizer in Δ, *i.e.*, $r'_0(\mathtt{n}, \mathrm{z}_0) > 1$, everything remains constant; otherwise we carry on inductively as expected.

Then g realizes the conclusion of the induction rule:

$$\mathrm{z} \rhd \Gamma, \; \mathrm{z}_0 \rhd P(0) \quad \longmapsto \quad \mathrm{g}(\mathtt{x}, \mathrm{z}, \mathrm{z}_0) \rhd\!\!\rhd P(t), \Delta$$

is provable by induction on (the value of) t.

Future work

- In this talk we have only considered *total* algebras.
- But partial functions occur naturally in some algebras.
- Hence, it would be of interest to extend the present work to a partial setting.