Operators for Computation over Partially Ordered Structures

Pedro Francisco Valencia Vizcaíno

University of Greifswald

November 18, 2016
Oaxaca, México.

Outline

By *ex falso sequitur quodlibet*, or simply *ex falso* for short, we mean intuitively the idea that we can derive anything from a contradiction. In slightly more formal terms, we mean a principle that allows us to derive all formulas of the form

$$\neg C \supset (C \supset A)$$

where A and B are arbitrary formulæ.

By *tertium non datur* we mean a principle that allows us to derive all formulas of the form

$$B \vee \neg B$$

where A is an arbitrary formula, and by *double negation elimination* we mean a principle that allows us to derive all formulas of the form

$$\neg\neg D \supset D.$$

Lemma

Over ML it is equivalent to have $\rightarrow \neg A \supset (A \supset B)$ *as initial sequents for all formulas* A *and* B *and to have the rule weakening:right.*

Lemma

Over ML it is equivalent to have $\to \neg A \supset (A \supset B)$ as initial sequents for all formulas A and B and to have the rule weakening:right.

Proof.

If we add weakening:right to ML we can prove ex-falso quodlibet in the following straightforward way:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \to A}{\neg A, A \to}}{\neg A, A \to B}\,\text{w:r}}{A, \neg A \to B}}{\neg A \to A \supset B}}{\to \neg A \supset (A \supset B)}$$

□

Equivalence between ex falso and weakening right

Perhaps a bit less obvious is the fact that with this form of ex-falso quodlibet ML proves *Weakening-right*:

$$
\begin{prooftree}
\AxiomC{$A \to A$}
\AxiomC{$A \to$}
\UnaryInfC{$\to \neg A$}
\UnaryInfC{$A \to \neg A$}
\BinaryInfC{$A \to A \wedge \neg A$}
\AxiomC{$\to \neg A \supset (A \supset B)$}
\AxiomC{$\neg A \to \neg A$}
\AxiomC{$A \to A$}
\AxiomC{$B \to B$}
\BinaryInfC{$A \supset B, A \to B$}
\BinaryInfC{$\neg A \supset (A \supset B), \neg A, A \to B$}
\RightLabel{Cut}
\BinaryInfC{$\neg A, A \to B$}
\UnaryInfC{$A \wedge \neg A, A \to B$}
\UnaryInfC{$A, A \wedge \neg A \to B$}
\UnaryInfC{$A \wedge \neg A, A \wedge \neg A \to B$}
\UnaryInfC{$A \wedge \neg A \to B$}
\RightLabel{Cut}
\BinaryInfC{$A \to B$}
\end{prooftree}
$$

$$\frac{A, \Gamma \to \Delta \qquad \neg A, \Gamma \to \Delta}{\Gamma \to \Delta}$$

Theorem

Cut elimination holds for ML.

Theorem

Cut elimination holds for ML.

Lemma

Let B be an atomic formula. If the sequent $\Gamma \to B$ is provable in ML^+ then there is a formula F which has B as a subformula not occurring in the scope of a negation such that $F \in \Gamma$.

Theorem

Cut elimination holds for ML.

Lemma

Let B be an atomic formula. If the sequent $\Gamma \to B$ is provable in ML^+ then there is a formula F which has B as a subformula not occurring in the scope of a negation such that $F \in \Gamma$.

Corollary

If A and B are atomic formulas then

$$ML \nvdash \neg A, A \to B$$

Russell's Paradox

$$
\dfrac{
  \dfrac{
    A \to A
    \qquad
    \dfrac{
      \to A \supset \neg A
      \qquad
      \dfrac{A \to A \qquad \neg A \to \neg A}{A \supset \neg A,\, A \to \neg A}
    }{A \to \neg A}
  }{A \to A \wedge \neg A}
  \qquad
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{A \to A}{\neg A,\, A \to}
        }{A \wedge \neg A,\, A \to}
      }{A,\, A \wedge \neg A \to}
    }{A \wedge \neg A,\, A \wedge \neg A \to}
  }{A \wedge \neg A \to}
}{
  \dfrac{A \to}{\to \neg A}
}\ \mathrm{Cut}
$$

Russell's Paradox

$$
\dfrac{
\dfrac{
\dfrac{
\to \neg A \supset A
\qquad
\dfrac{
\dfrac{\dfrac{\dfrac{A \to A}{\neg A, A \to}}{A, \neg A \to}}{\neg A \to \neg A}
\qquad
A \to A
}{\neg A \supset A, \neg A \to A}
}{\neg A \to A}\,\text{Cut}
\qquad
\dfrac{\dfrac{\dfrac{A \to A}{\neg A, A \to}}{A, \neg A \to}}{\neg A \to \neg A}
}{\neg A \to A \wedge \neg A}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \to A}{\neg A, A \to}}{A \wedge \neg A, A \to}}{A, A \wedge \neg A \to}}{A \wedge \neg A, A \wedge \neg A \to}}{A \wedge \neg A \to}
}{\neg A \to}\,\text{Cut}
$$

Russell's Paradox

$$\frac{P_1 \qquad P_2}{\rightarrow}$$

Some Other Principles

Usually the principle $\neg\neg A \wedge \neg\neg B \supset \neg\neg(A \wedge B)$ is considered intuitionistic, but we can prove it in ML.

Das nicht nichtet sich selbst

We have been able to separate simple negation, formally, in the sequent calculus, as follows:

$$A \supset B \wedge \neg B \Rightarrow \neg\neg(A \supset B \wedge \neg B) \Rightarrow A \supset \neg C \wedge \neg\neg C \Rightarrow \neg\neg(A \supset \neg C \wedge \neg\neg C) \Rightarrow \neg A$$

Theorem

The following hold on the basis of minimal logic:

1. *Double negation elimination implies Ex falso and Tertium non datur.*
2. *Ex falso + Tertium non datur imply Double negation elimination.*
3. *Ex falso does not imply Tertium non datur.*
4. *Tertium non datur does not impy Ex falso.*
5. *Tertium non datur does not imply Double negation elimination.*

Omniscience Principles

Corollary

ML^+ does not prove $\neg\forall x \neg A(x) \to \exists x A(x)$ if A is an atomic formula.

Theorem

$\neg\forall x A(x) \to \exists x \neg A(x)$ is not provable in ML^+.

Omniscience Principles

Corollary

ML^+ does not prove $\neg\forall x \neg A(x) \to \exists x A(x)$ if A is an atomic formula.

Theorem

$\neg\forall x A(x) \to \exists x \neg A(x)$ is not provable in ML^+.

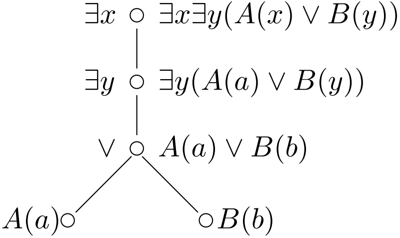
This means in particular that one shouldn't blame the excluded middle for the derivability of the omniscience principle, it's both excluded middle and ex falso that are responsible for the omniscience principle, and hence, on the basis of minimal logic, it is double negation elimination that is responsible for the omniscience principle and not tertium non datur.

Definition

A first order formula is geometric if it uses only $\exists, \wedge, \vee$.

Lemma (Pulling up the $\exists$)

Let C be a geometric formula. Then there is a geometric formula D such that D has all its existential quantifiers from the top down of its parsing tree and such that $D \to C$ and $C \to D$.


$\exists x$
$\exists x \exists y (A(x) \vee B(y))$
$\exists y$
$\exists y (A(a) \vee B(y))$
$\vee$
$A(a) \vee B(b)$
$A(a)$
$B(b)$

Take

$$\Gamma \to \Delta$$

First, we transform each side into an infinitary disjunction. (Just as in the finitary case except that now we may have infinitely many disjuncts.)
Next we can write the succeedent Δ in the desired form (the only difference being that we might get an infinitary disjunction instead of a finitary one).
After that we break up the disjunction on the left into (possibly) infinitely many pieces. And then we deal with each piece individually. Finally we get the desired canonical form, because after this, it is just a matter of rearranging the $\exists$'s and pulling them out, just like in the finitary case.

Theorem

The new inference rules scheme for infinitary geometric theories is equivalent to the addition of initial sequents corresponding to the infinitary geometric axioms.

Definition

The infinitary non-geometric degree of a formula is defined as

1. $\partial_{\neg G}(A) = 0$ *if A is geometric*
2. $\partial_{\neg G}(\forall x A) = \partial_{\neg G}(A) + 1$
3. $\partial_{\neg G}(\neg A) = \partial_{\neg G}(A) + 1$
4. $\partial_{\neg G}(\bigvee A_i) = \sup(\{\partial_{\neg G}(A_i)\})$
5. $\partial_{\neg G}(\exists x A) = \partial_{\neg A}(A)$
6. $\partial_{\neg G}(A \wedge B) = \sup(\partial_{\neg G}(A), \partial_{\neg G}(B))$
7. $\partial_{\neg G}(A \supset B) = \sup(\partial_{\neg G}(A), \partial_{\neg G}(B)) + 1$

Lemma

Let C be a formula of infinitary non-geometric degree $\alpha > 0$ and $\mathbf{P_1}$ and $\mathbf{P_2}$ proofs of $\Gamma \to \Delta$ and $\Pi \to \Lambda$ of infinitary non-geometric degrees less than α. Then we can make a proof of $\Gamma, \Pi - C \to \Delta - C, \Lambda$ of infinitary non-geometric degree less than α.

Theorem

Let T be an infinitary geometric theory and $\Gamma \to \Delta$ an infinitary geometric sequent, such that T proves $\Gamma \to \Delta$ in classical logic. Then T proves $\Gamma' \to \bigvee \Delta'$ in intuitionistic logic, where Γ' and Δ' are alphabetically equivalent to Γ and Δ, respectively, ie, they may have different free variables, and they do not exhaust the free variables in the language, in fact they leave infinitely many free variables unused.

Basic Concepts

Let $\mathcal{A} = (\mathcal{U}_{\mathcal{A}}; D; f_1, \ldots, f_n; R_1, \ldots, R_m, =, \leq)$ be an algebraic structure with two different constants $a, b \in D$. An **$\mathcal{A}$-machine** $\mathfrak{M}$ (see [**?**]) has registers that may be occupied by elements of $\mathcal{U}_{\mathcal{A}}$; index registers which may only be occupied by positive natural numbers, and $\mathfrak{M}$ may, according to its program, execute any of the following types of instructions:

- Computation instructions:

  $\ell\colon\ Z_j := d_k$ $\qquad\qquad (d_k \in D)$

  $\ell\colon\ Z_j := f_k(Z_{j_1}, \ldots, Z_{j_{m_k}})$ $\qquad\qquad$ (e.g. $\ell\colon\ Z_j := Z_{j_1} + Z_{j_2}$)

- Branching instructions:

  $\ell\colon$ if $Z_i = Z_j$ then goto ℓ_1 else goto ℓ_2

  $\ell\colon$ if $R_k(Z_{j_1}, \ldots, Z_{j_{n_k}})$ then goto ℓ_1 else goto ℓ_2

  $\ell\colon$ if $Z_{j_1} \leq Z_{j_{n_k}}$ then goto ℓ_1 else goto ℓ_2

- Copy instructions:

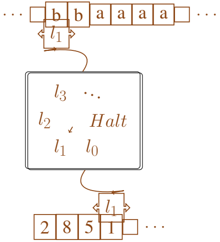
  $\ell\colon\ Z_{I_j} := Z_{I_k}$

  For indirect addressing we have the following index instructions:

  $\ell\colon\ I_j := 1$

  $\ell\colon\ I_j := I_j + 1$

  $\ell\colon$ if $I_j = I_k$ then goto ℓ_1 else goto ℓ_2



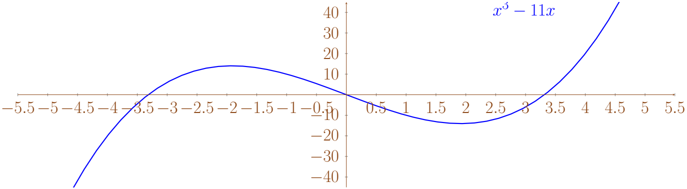
Let f be an $\mathcal{A}$-computable function. Then $\nu[f](x_1, \ldots, x_n)$ is defined as the set

$$\{y \in \mathcal{U}_{\mathcal{A}} \mid \exists (y_2, \ldots, y_m) \in {\mathcal{U}_{\mathcal{A}}}^{\infty} \, [f(x_1, \ldots, x_n, y, y_2, \ldots, y_m) = a]\}$$

A ν-operator instruction in a program takes the form

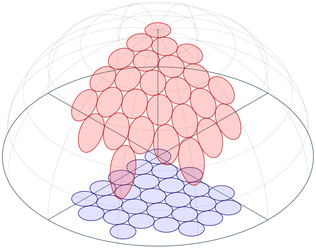
$$Z_j := y \in \nu[f](Z_1, \ldots, Z_{I_1})$$

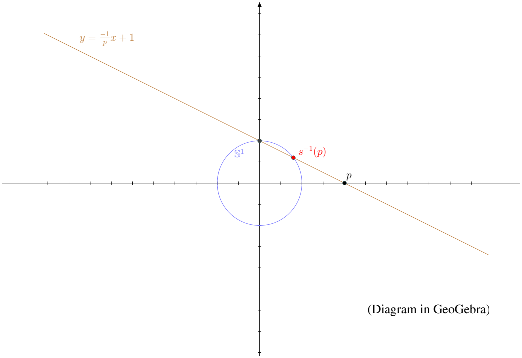
for some (non-deterministically obtained) value of y (see **[?]**).



$$Z_j := y \in \{-2\sqrt{11}, 0, 2\sqrt{11}\}$$

Relation between $\nu_{\sup}$ and $\nu_{\max}$





We note that in $\mathbb{R}_{\text{field}}$ the operators $\nu_{\sup}$ and ν_{smub} coincide: they both just give us the usual supremum of the zero-set.

Lemma. *The stereographic projection's inverse is $\nu_{\sup}$-computable in the algebraic structure $\mathbb{R}_{\text{field}}$.*

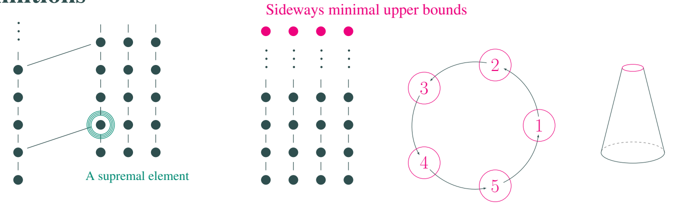
Lemma

From the stereographic projection's inverse we can construct a function from $\mathbb{R}$ to a bounded ν_{sup}-computable set.

Theorem

Given a ν_{sup}-computable bijection between the real line and a ν_{sup}-computable bounded set, every $\nu_{\max}$-computable function is ν_{sup}-computable.

Definitions



If neither $x \leq y$ nor $y \leq x$ we write $x \perp y$.

Let $Y \subseteq \mathcal{U}_{\mathcal{A}}$. A *supremal element in $\mathcal{U}_{\mathcal{A}}$ for Y* is an element $p \in \mathcal{U}_{\mathcal{A}}$ such that

1. $Y \neq \varnothing \to \exists c \in Y (c \leq p)$
2. $\forall c \in Y [(c \leq p) \vee (c \perp p)]$
3. $\forall p'$ satisfying 1 and 2 above $[(p \leq p') \vee (p \perp p')]$.

Let $Y \subseteq \mathcal{U}_{\mathcal{A}}$. A *sideways minimal upper bound (smub) element in $\mathcal{U}_{\mathcal{A}}$ for Y* is an element $p \in P$ such that

1. $Y \neq \varnothing \to \exists c \in Y (c \leq p)$
2. $\forall c \in Y [(c \leq p) \vee (c \perp p)]$
3. $\forall d_0 \in Y [(d_0 < p) \to \exists d_1 \in Y (d_0 < d_1 \leq p)]$
4. $\forall p'$ satisfying 1 to 3 above $[(p \leq p') \vee (p \perp p')]$.

ν_{smub} is an operator which for an $(\mathcal{U}_{\mathcal{A}}, \leq)$-computable function f returns non-deterministically some sideways minimal upper bound element in $\mathcal{U}_{\mathcal{A}}$ of $f^{-1}[\{a\}]$. Analogously ν_{sup} returns non-deterministically some supremal element of $f^{-1}[\{a\}]$ and $\nu_{\max}$ returns non-deterministically some maximal element of $f^{-1}[\{a\}]$.

Lemma

One could utilise the ν_{smub} operator to perform any of the following tasks:

1) *Check whether for a computable function f, and a vector $\vec{x}$, the following equality holds*

$$f(\vec{x}) = a.$$

2) *Check whether for a computable function f and an arbitrary prefix vector $\vec{x}$, there exists a vector $\vec{y}$ such that*

$$f(\vec{x} \star \vec{y}) = a.$$

3) *Construct a function $g_{\vec{x},\vec{c}}$ such that [1.] $g_{\vec{x},\vec{c}}(\vec{x}) = \vec{c}$.*
$\forall \vec{y} \in \mathcal{U_A}^{\infty} \, [\vec{y} \neq \vec{x} \supset (g_{\vec{x},\vec{c}}(\vec{y}) = \mathscr{P}(\vec{y}))]$.

Theorem

Let minimality be decidable in $\mathcal{U}_{\mathcal{A}}$. Then one could search for zeroes without non-halting issues with a ν_{sup}-machine.

Theorem

Let f be an $\mathcal{A}$-computable function such that $f^{-1}[\{a\}]$ is finite. Then all the elements of $f^{-1}[\{a\}]$ can be given algorithmically with ν_{smub}-machines over $\mathcal{A}$.

Proposition

Let $\mathbb{N} \subseteq D$ *and* f *be an* $\mathcal{A}$*-computable function. Suppose further that* $\forall i \in \mathbb{N}(f(i) = a \vee f(i) = b)$*. A* $\nu_{\min}$*-oracle machine can determine precisely whether* $\forall i \in \mathbb{N}(f(i) = a)$ *or whether* $\exists j \in \mathbb{N}$ *such that* $f(j) = b$*. Moreover it can produce such a* j *in one step.*

Proposition

Let $\mathbb{N} \subseteq D$ and f be an $\mathcal{A}$-computable function. Suppose further that $\forall i \in \mathbb{N}(f(i) = a \vee f(i) = b)$. A $\nu_{\min}$-oracle machine can determine precisely whether $\forall i \in \mathbb{N}(f(i) = a)$ or whether $\exists j \in \mathbb{N}$ such that $f(j) = b$. Moreover it can produce such a j in one step.

Proposition

Let φ be any sentence of the language of **PA**. *Then a $\nu_{\max}$-oracle machine can decide in one step whether φ is a theorem of* **PA**.

Blum, L., Shub, M., and Smale, S.: *On a theory of computation and complexity over the real numbers: NP-completeness, recursive funcitons and universal machine*; Bulletin of the Amer. Math. Soc. 21 (1989), 1–46.

Moschovakis, Y. N.: *Abstract first order computability*; Transactions of the American Mathematical Society 138 (1969), 427–464. http://dx.doi.org/10.2307/1994926

Gaßner, C.: *Computation over Algebraic Structures and a Classification of Undecidable Problems*; (2015), accepted in MSCS.

Gentzen, Gerhard, *Untersuchungen über das logische Schließen*, Mathematische Zeitschrift. 39 (2): 176-210, 1934.

Girard, Jean-Yves, *Proofs and Types*, Cambridge University Press, 1989.

Negri, Sara, von Plato, Jan, *Structural Proof Theory*, Cambridge University Press, 2001.

Takeuti, Gaisi, *Proof Theory*, North Holland Publishing Company, 1975.

Troelstra Anne S., Schwichtenberg Helmut, *Basic Proof Theory*, Cambridge University Press, 2000.

Valencia Vizcaíno Pedro F., *Some Uses of Cut Elimination*, PhD Thesis, Leeds, 2014.