Gödel's functional interpretation and the concept of learning

Thomas Powell

University of Innsbruck

Workshop on
Efficient and Natural Proof Systems

University of Bath, UK
16 December 2015

Notation. Σ_2^0 formulas written as $P \equiv \exists x \forall y |P|_y^x$, where $|P|_y^x$ is always decidable.

Notation. Σ_2^0 formulas written as $P \equiv \exists x \forall y |P|_y^x$, where $|P|_y^x$ is always decidable.

A SIMPLE NON-CONSTRUCTIVE THEOREM: $\exists n \forall m (|P|^m \to |P|^n)$

"In a pub there is always a person such that if anyone else is drinking, then that person in drinking"

Notation. Σ_2^0 formulas written as $P \equiv \exists x \forall y |P|_y^x$, where $|P|_y^x$ is always decidable.

A simple non-constructive theorem: $\exists n \forall m (|P|^m \to |P|^n)$

"In a pub there is always a person such that if anyone else is drinking, then that person in drinking"

$$\dfrac{\exists k |P|^k \vee \forall k \neg |P|^k \qquad \dfrac{|P|^k \to \forall m (|P|^m \to |P|^k)}{|P|^k \to \exists n \forall m (|P|^m \to |P|^n)}}{\exists k |P|^k \to \exists n \forall m (|P|^m \to |P|^n)} \qquad \dfrac{\forall k \neg |P|^k \to |P|^m \to |P|^0}{\forall k \neg |P|^k \to \forall m (|P|^m \to |P|^0)}}{\forall k \neg |P|^k \to \exists n \forall m (|P|^m \to |P|^n)}$$

$$\exists n \forall m (|P|^m \to |P|^n)$$

Notation. Σ^0_2 formulas written as $P \equiv \exists x \forall y |P|^x_y$, where $|P|^x_y$ is always decidable.

A simple non-constructive theorem: $\exists n \forall m (|P|^m \to |P|^n)$

"In a pub there is always a person such that if anyone else is drinking, then that person in drinking"

$$\dfrac{\exists k |P|^k \vee \forall k \neg |P|^k \qquad \dfrac{|P|^k \to \dfrac{|P|^k \to \forall m(|P|^m \to |P|^k)}{|P|^k \to \exists n \forall m(|P|^m \to |P|^n)}}{\exists k |P|^k \to \exists n \forall m(|P|^m \to |P|^n)} \qquad \dfrac{\forall k \neg |P|^k \to |P|^m \to |P|^0}{\forall k \neg |P|^k \to \forall m(|P|^m \to |P|^0)}}{\forall k \neg |P|^k \to \exists n \forall m(|P|^m \to |P|^n)}$$

$$\exists n \forall m (|P|^m \to |P|^n)$$

There is no *effective* way of realizing $\exists n$. So what is the constructive interpretation of the drinkers paradox?

Method I: Hilbert's ϵ-calculus

Idea: Replace quantifiers by 'magic' ϵ-terms:

$$\exists k A(k) \rightsquigarrow A(\epsilon_k A),$$

and quantifier axioms by critical formulas:

$$A(t) \to A(\epsilon_k A).$$

1. *Translation.* Convert proofs in predicate logic to proofs in epsilon calculus. Instances of quantifier axioms replaced by critical formulas.

2. *Epsilon elimination.* Suppose we only use a finite set of critical formulas. Interpret all ϵ-terms by 0. If we find a mistake i.e. $A(t) \wedge \neg A(0)$, 'learn' from this mistake and update $\epsilon_k A \mapsto t$.

Interpreted proof:

$$\dfrac{\exists k|P|^k \vee \forall k\neg|P|^k \qquad \dfrac{\dfrac{|P|^k \to \forall m(|P|^m \to |P|^k)}{|P|^k \to \exists n\forall m(|P|^m \to |P|^n)}}{\exists k|P|^k \to \exists n\forall m(|P|^m \to |P|^n)} \qquad \dfrac{\dfrac{\dfrac{\neg|P|^m \to |P|^m \to |P|^0}{\forall k\neg|P|^k \to |P|^m \to |P|^0}}{\forall k\neg|P|^k \to \forall m(|P|^m \to |P|^0)}}{\forall k\neg|P|^k \to \exists n\forall m(|P|^m \to |P|^n)}}{\exists n\forall m(|P|^m \to |P|^n)}$$

Critical formulas:

ϵ-elimination:

Interpreted proof:

$$\dfrac{|P|^{\epsilon_k} \vee \neg|P|^{\epsilon_k} \qquad \dfrac{|P|^{\epsilon_k} \to \forall m(|P|^m \to |P|^{\epsilon_k})}{|P|^{\epsilon_k} \to \exists n \forall m(|P|^m \to |P|^n)} \qquad \dfrac{\dfrac{\dfrac{\neg|P|^m \to |P|^m \to |P|^0}{\neg|P|^{\epsilon_k} \to |P|^m \to |P|^0}}{\neg|P|^{\epsilon_k} \to \forall m(|P|^m \to |P|^0)}}{\neg|P|^{\epsilon_k} \to \exists n \forall m(|P|^m \to |P|^n)}}{\exists n \forall m(|P|^m \to |P|^n)}$$

Critical formulas:

$$|P|^m \to |P|^{\epsilon_k}$$

ϵ-elimination:

Interpreted proof:

$$\dfrac{|P|^{\epsilon_k} \vee \neg|P|^{\epsilon_k} \qquad \dfrac{|P|^{\epsilon_k} \to |P|^{\epsilon_m\,\epsilon_k} \to |P|^{\epsilon_k}}{|P|^{\epsilon_k} \to \exists n(|P|^{\epsilon_m\,n} \to |P|^{n})} \qquad \dfrac{\dfrac{\neg|P|^{\varepsilon_m\,0} \to |P|^{\varepsilon_m\,0} \to |P|^{0}}{\neg|P|^{\epsilon_k} \to |P|^{\epsilon_m\,0} \to |P|^{0}}}{\neg|P|^{\epsilon_k} \to \exists n(|P|^{\epsilon_m\,n} \to |P|^{n})}}{\exists n(|P|^{\epsilon_m\,n} \to |P|^{n})}$$

Critical formulas:

$$|P|^{\epsilon_m\,0} \to |P|^{\epsilon_k}$$

ϵ-elimination:

Interpreted proof:

$$\dfrac{|P|^{\epsilon_k} \vee \neg|P|^{\epsilon_k} \qquad \dfrac{|P|^{\epsilon_k} \to |P|^{\epsilon_m\,\epsilon_k} \to |P|^{\epsilon_k}}{|P|^{\epsilon_k} \to (|P|^{\epsilon_m\,\epsilon_n} \to |P|^{\epsilon_n})} \qquad \dfrac{\dfrac{\neg|P|^{\varepsilon_m\,0} \to |P|^{\varepsilon_m\,0} \to |P|^{0}}{\neg|P|^{\epsilon_k} \to |P|^{\epsilon_m\,0} \to |P|^{0}}}{\neg|P|^{\epsilon_k} \to (|P|^{\epsilon_m\,\epsilon_n} \to |P|^{\epsilon_n})}}{|P|^{\epsilon_m\,\epsilon_n} \to |P|^{\epsilon_n}}$$

Critical formulas:

$$|P|^{\epsilon_m\,0} \to |P|^{\epsilon_k}$$

$$(|P|^{\epsilon_m\,\epsilon_k} \to |P|^{\epsilon_k}) \to (|P|^{\epsilon_m\,\epsilon_n} \to |P|^{\epsilon_n})$$

$$(|P|^{\epsilon_m\,0} \to |P|^{0}) \to (|P|^{\epsilon_m\,\epsilon_n} \to |P|^{\epsilon_n})$$

ϵ-elimination:

Interpreted proof:

$$\dfrac{|P|^{\epsilon_k} \vee \neg|P|^{\epsilon_k} \qquad \dfrac{|P|^{\epsilon_k} \to |P|^{\epsilon_m\,\epsilon_k} \to |P|^{\epsilon_k}}{|P|^{\epsilon_k} \to (|P|^{\epsilon_m\,\epsilon_n} \to |P|^{\epsilon_n})} \qquad \dfrac{\dfrac{\neg|P|^{\varepsilon_m\,0} \to |P|^{\varepsilon_m\,0} \to |P|^{0}}{\neg|P|^{\epsilon_k} \to |P|^{\epsilon_m\,0} \to |P|^{0}}}{\neg|P|^{\epsilon_k} \to (|P|^{\epsilon_m\,\epsilon_n} \to |P|^{\epsilon_n})}}{|P|^{\epsilon_m\,\epsilon_n} \to |P|^{\epsilon_n}}$$

Critical formulas:

$$\begin{aligned} |P|^{\epsilon_m\,0} \to |P|^{0} \qquad & ? \\ (|P|^{\epsilon_m\,0} \to |P|^{0}) \to (|P|^{\epsilon_m\,0} \to |P|^{0}) \qquad & \checkmark \\ (|P|^{\epsilon_m\,0} \to |P|^{0}) \to (|P|^{\epsilon_m\,0} \to |P|^{0}) \qquad & \checkmark \end{aligned}$$

ϵ-elimination:

- Try $\epsilon_k = \epsilon_n = 0$. Works unless $|P|^{\epsilon_m\,0} \wedge \neg|P|^{0}$.

Interpreted proof:

$$\dfrac{|P|^{\epsilon_k} \vee \neg|P|^{\epsilon_k} \qquad \dfrac{|P|^{\epsilon_k} \to |P|^{\epsilon_m \epsilon_k} \to |P|^{\epsilon_k}}{|P|^{\epsilon_k} \to (|P|^{\epsilon_m \epsilon_n} \to |P|^{\epsilon_n})} \qquad \dfrac{\dfrac{\neg|P|^{\varepsilon_m 0} \to |P|^{\varepsilon_m 0} \to |P|^0}{\neg|P|^{\epsilon_k} \to |P|^{\epsilon_m 0} \to |P|^0}}{\neg|P|^{\epsilon_k} \to (|P|^{\epsilon_m \epsilon_n} \to |P|^{\epsilon_n})}}{|P|^{\epsilon_m \epsilon_n} \to |P|^{\epsilon_n}}$$

Critical formulas:

$$|P|^{\epsilon_m 0} \to |P|^{\epsilon_m 0} \qquad \checkmark$$

$$(|P|^{\epsilon_m \epsilon_m 0} \to |P|^{\epsilon_m 0}) \to (|P|^{\epsilon_m \epsilon_m 0} \to |P|^{\epsilon_m 0}) \qquad \checkmark$$

$$(|P|^{\epsilon_m 0} \to |P|^0) \to (|P|^{\epsilon_m \epsilon_m 0} \to |P|^{\epsilon_m 0}) \qquad \checkmark$$

ϵ-elimination:

- Try $\epsilon_k = \epsilon_n = 0$. Works unless $|P|^{\epsilon_m 0} \wedge \neg|P|^0$.
- But now we have a witness for $\exists k|P|^k$, so set $\epsilon_k = \epsilon_m = \epsilon_m 0$.

Finitary drinker's paradox I: For an arbitrary ϵ-term $\epsilon_m(\cdot)$ there exists some ϵ_n satisfying

$$|P|^{\epsilon_m \epsilon_n} \to |P|^{\epsilon_n}.$$

This can be computed by the algorithm

- `Set` $\epsilon_n := 0$.
- `Check` $|P|^{\epsilon_m 0} \to |P|^0$. `If true, END.`
- `Else` $\epsilon_n := \epsilon_m 0$.

Finitary drinker's paradox I: For an arbitrary ϵ-term $\epsilon_m(\cdot)$ there exists some ϵ_n satisfying

$$|P|^{\epsilon_m \epsilon_n} \to |P|^{\epsilon_n}.$$

This can be computed by the algorithm

- `Set` $\epsilon_n := 0$.
- `Check` $|P|^{\epsilon_m 0} \to |P|^0$. `If true, END.`
- `Else` $\epsilon_n := \epsilon_m 0$.

The term $\epsilon_m(\cdot)$ represents the *proof theoretic environment*, a measure of how we might *use* the drinkers paradox as a lemma. More specifically, exactly when we need the $\forall$-axiom

$$\exists n \forall m(|P|^m \to |P|^n) \to \exists n(|P|^t \to |P|^n).$$

Method II: Gödel's functional (Dialectica) interpretation

Idea: A two stage translation: *Negative translation + Dialectica interpretation.*

1. Eliminate classical reasoning by applying negative translation (can be more flexible here e.g. ignore atomic formulas).

2. Extract realizing terms for *Dialectica* interpretation of this formula. More complex than realizability - need to fully Skolemize implication:

$$\begin{aligned}(A \to B) &\rightsquigarrow (\exists x \forall y |A|^x_y \to \exists u \forall v |B|^u_v) \\ &\rightsquigarrow \forall x \exists u \forall v \exists y (|A|^x_y \to |B|^u_v) \\ &\rightsquigarrow \exists U, Y \forall x, v (|A|^x_{Yxv} \to |B|^{Ux}_v)\end{aligned}$$

Contraction problem: Interpretation of classical reasoning requires us to test atomic formulas and use case definitions.

$$\frac{\exists k|P|^k \vee \forall k\neg|P|^k \qquad \begin{array}{c}[\exists k|P|^k] \\ \vdots \\ \exists n\forall m(|P|^m \to |P|^n)\end{array} \qquad \begin{array}{c}[\forall k\neg|P|^k] \\ \vdots \\ \exists n\forall m(|P|^m \to |P|^n)\end{array}}{\exists n\forall m(|P|^m \to |P|^n)}$$

$$\dfrac{\exists k|P|^k \vee \forall k\neg|P|^k \qquad \begin{array}{c}[\exists k|P|^k] \\ \vdots \\ \neg\neg\exists n\forall m(|P|^m \to |P|^n)\end{array} \qquad \begin{array}{c}[\forall k\neg|P|^k] \\ \vdots \\ \neg\neg\exists n\forall m(|P|^m \to |P|^n)\end{array}}{\neg\neg\exists n\forall m(|P|^m \to |P|^n)}$$

$$\dfrac{\exists k|P|^k \vee \forall k \neg |P|^k \qquad \begin{array}{c}[|P|^k] \\ \vdots \\ |P|^{gk} \to |P|^k\end{array} \qquad \begin{array}{c}[\forall k \neg |P|^k] \\ \vdots \\ \neg\neg\exists n \forall m(|P|^m \to |P|^n)\end{array}}{\neg\neg\exists n \forall m(|P|^m \to |P|^n)}$$

First branch:

$$\begin{aligned}
&\exists k|P|^k \to \neg\neg\exists n\forall m(|P|^m \to |P|^n)\\
\rightsquigarrow &\exists k|P|^k \to \forall g^{\mathbb{N}\to\mathbb{N}}\exists n(|P|^{gn} \to |P|^n)\\
\rightsquigarrow &\forall g, k\exists n(|P|^k \to |P|^{gn} \to |P|^n)\\
\rightsquigarrow &\forall g, k(|P|^k \to |P|^{gk} \to |P|^k)
\end{aligned}$$

$$\dfrac{\exists k|P|^k \vee \forall k \neg |P|^k \qquad \begin{array}{c}[|P|^k] \\ \vdots \\ |P|^{gk} \to |P|^k\end{array} \qquad \begin{array}{c}[\neg|P|^{g0}] \\ \vdots \\ |P|^{g0} \to |P|^0\end{array}}{\neg\neg\exists n \forall m(|P|^m \to |P|^n)}$$

First branch:

$$\begin{aligned}
&\exists k|P|^k \to \neg\neg\exists n\forall m(|P|^m \to |P|^n)\\
\rightsquigarrow&\exists k|P|^k \to \forall g^{\mathbb{N}\to\mathbb{N}}\exists n(|P|^{gn} \to |P|^n)\\
\rightsquigarrow&\forall g, k\exists n(|P|^k \to |P|^{gn} \to |P|^n)\\
\rightsquigarrow&\forall g, k(|P|^k \to |P|^{gk} \to |P|^k)
\end{aligned}$$

Second branch:

$$\begin{aligned}
&\forall k\neg|P|^k \to \neg\neg\exists n\forall m(|P|^m \to |P|^n)\\
\rightsquigarrow&\forall k\neg|P|^k \to \forall g\exists n(|P|^{gn} \to |P|^n)\\
\rightsquigarrow&\forall g\exists k, n(\neg|P|^k \to |P|^{gn} \to |P|^n)\\
\rightsquigarrow&\forall g(\neg|P|^{g0} \to |P|^{g0} \to |P|^0)
\end{aligned}$$

$$\dfrac{|P|^{g0} \vee \neg|P|^{g0} \qquad \begin{array}{c}[|P|^{g0}]\\ \vdots \\ |P|^{g(g0)} \to |P|^{g0}\end{array} \qquad \begin{array}{c}[\neg|P|^{g0}]\\ \vdots \\ |P|^{g0} \to |P|^{0}\end{array}}{\neg\neg\exists n \forall m(|P|^m \to |P|^n)}$$

First branch:

$$\begin{aligned}
&\exists k|P|^k \to \neg\neg\exists n\forall m(|P|^m \to |P|^n)\\
\rightsquigarrow&\exists k|P|^k \to \forall g^{\mathbb{N}\to\mathbb{N}}\exists n(|P|^{gn} \to |P|^n)\\
\rightsquigarrow&\forall g, k\exists n(|P|^k \to |P|^{gn} \to |P|^n)\\
\rightsquigarrow&\forall g, k(|P|^k \to |P|^{gk} \to |P|^k)
\end{aligned}$$

Second branch:

$$\begin{aligned}
&\forall k\neg|P|^k \to \neg\neg\exists n\forall m(|P|^m \to |P|^n)\\
\rightsquigarrow&\forall k\neg|P|^k \to \forall g\exists n(|P|^{gn} \to |P|^n)\\
\rightsquigarrow&\forall g\exists k, n(\neg|P|^k \to |P|^{gn} \to |P|^n)\\
\rightsquigarrow&\forall g(\neg|P|^{g0} \to |P|^{g0} \to |P|^0)
\end{aligned}$$

$$\dfrac{|P|^{g0} \vee \neg|P|^{g0} \qquad \begin{array}{c}[|P|^{g0}] \\ \vdots \\ |P|^{g(g0)} \to |P|^{g0}\end{array} \qquad \begin{array}{c}[\neg|P|^{g0}] \\ \vdots \\ |P|^{g0} \to |P|^{0}\end{array}}{|P|^{g(Ng)} \to |P|^{Ng}}$$

Solved by

$$Ng := \begin{cases} 0 & \text{if } \neg|P|^{g0} \\ g0 & \text{if } |P|^{g0} \end{cases}$$

Finitary drinker's paradox II: For an arbitrary function $g\colon \mathbb{N} \to \mathbb{N}$ there exists some $N\colon (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ satisfying

$$|P|^{g(Ng)} \to |P|^{Ng}.$$

This can be defined as

$$Ng := \begin{cases} 0 & \text{if } \neg|P|^{g0} \\ g0 & \text{otherwise.} \end{cases}$$

Finitary drinker's paradox II: For an arbitrary function $g\colon \mathbb{N} \to \mathbb{N}$ there exists some $N\colon (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ satisfying

$$|P|^{g(Ng)} \to |P|^{Ng}.$$

This can be defined as

$$Ng := \begin{cases} 0 & \text{if } \neg|P|^{g0} \\ g0 & \text{otherwise.} \end{cases}$$

The proof theoretic environment is represented by an explicit 'counterexample function' g. Any instance of the drinkers paradox in a bigger proof will involve a concrete instantiation g_v of g:

$$\exists n \forall m(|P|^m \to |P|^n) \to B$$
$$\exists U, g \forall n, v((|P|^{gn} \to |P|^n) \to |B|_v^{Un})$$

and hence $\forall v|B|_v^{U(Ng)v}$ holds.

General finitary drinker's paradox: There exists an approximate witness $\mathcal{N}$ to $\exists n \forall m(|P|^n \to |P|^m)$, that works relative to any environment $\mathcal{M}$ (representing $\forall m$).

Technique	$\mathcal{N}$	$\mathcal{M}$
ϵ-calculus	$\epsilon_n := 0$ `Check` $\|\mathtt{P}\|^{\epsilon_{\mathtt{m}}\mathtt{0}} \to \|\mathtt{P}\|^{\mathtt{0}}$. `If true, END.` `Else` $\epsilon_n := \epsilon_m 0$	$\epsilon_m(\cdot)$
Dialectica	$Ng := \begin{cases} 0 & \text{if } \neg\|P\|^{g0} \\ g0 & \text{otherwise} \end{cases}$	$g\colon \mathbb{N} \to \mathbb{N}$

They both carry out learning, but in completely different frameworks:

ϵ-calculus $\rightsquigarrow$ Explicitly via ϵ-elimination procedure

Dialectica $\rightsquigarrow$ Implicitly via contractions in negative-translated proofs

A **learning algorithm** $\mathcal{L} := (\mathsf{Good}, q, p, \oplus)$ of type X, Y, Z consists of the following components:

- A decidable predicate Good on $X \times Y$ with the meaning

$$\mathsf{Good}(x, y) \Leftrightarrow x \text{ is good relative to a counterexample object } y;$$

- A function $q\colon X \to Z$ which assigns a final value $q(x) \in Z$ to a good element $x \in X$;
- A function $p\colon X \to Y$ that takes each candidate $x \in X$ and returns a counterexample object $p(x) \in Y$;
- An operation $\oplus\colon X \times Y \to X$ which updates $x \in X$ with new information obtained from a counterexample object y: the update

$$x \mapsto x \oplus y$$

is an instance of *learning*.

The **learning procedure** starting from $x_0 \in X$ is the sequence defined by

$$\neg \mathsf{Good}(x_i, p(x_i)) \Rightarrow x_{i+1} := x_i \oplus p(x_i).$$

This sequence is finite iff it eventually arrives at some good element x_k i.e. $Q(x_k, p(x_k))$. In this case we say that $q(x_k)$ is the **limit** of $\mathcal{L}_{x_0}$ and write

$$\lim_{x_0} \mathcal{L} := q(x_k).$$

REMARK. If $x \oplus p(x) \prec x$ for some well-founded ordering $\prec$ then $\lim_{x_0} \mathcal{L}$ is definable by well-founded recursion on $\prec$.

What are learning procedures designed to compute? Suppose that

- A is an initial predicate over X,
- B is a target predicate over Z, and

$$\forall x \begin{cases} \mathsf{Good}(x, p(x)) \to A(x) \to B(q(x)) \\ \neg\mathsf{Good}(x, p(x)) \to A(x) \to A(x \oplus p(x)) \end{cases}$$

then

$$\forall x_0 (A(x_0) \to B(\lim_{x_0} \mathcal{L}))$$

whenever the limit exists.

The next two slides are inspired by (Schwichtenberg 2004). Consider the least element principle for Σ_1-formulas relative to some arbitrary well-founded ordering $\prec$:

$$\exists x|C|^x \to \exists y(|C|^y \wedge \forall z \prec y \neg|C|^z).$$

The functional interpretation of this is given by

$$\begin{aligned}
&\rightsquigarrow \exists x|C|^x \to \neg\neg\exists y(|C|^y \wedge \forall z \prec y \neg|C|^z) \\
&\rightsquigarrow \exists x|C|^x \to \forall p\exists y(|C|^y \wedge (p(y) \prec y \to \neg|C|^{p(y)})) \\
&\rightsquigarrow \forall x, p\exists y(|C|^x \to |C|^y \wedge (p(y) \prec y \to \neg|C|^{p(y)})) \quad (*)
\end{aligned}$$

Theorem. Define $\mathcal{L}_p := (\mathsf{Good}_C, \mathrm{id}, p, \pi_1)$ for

$$\mathsf{Good}_C(y, z) :\equiv z \prec y \to \neg|C|^z.$$

Then $(*)$ is realized by

$$\lambda x, p \,.\, \lim_x \mathcal{L}_p$$

For any $a \geq b > 0$ there exist some n, m such that $am + bn \mid a, b$

Proof. Define the measure $\mu\colon \mathbb{N}^2 \to \mathbb{N}$ by $\mu\underline{x} := \underline{x} \cdot (a, b)$ and the ordering $\succ$ on $(\mathbb{N}^2)^*$ by

$$[] \succ [\underline{x}] \text{ and } s * \underline{x} \succ s * \underline{x} * \underline{y} \text{ iff } \mu\underline{x} > \mu\underline{y}.$$

Define $|C|^s :\equiv \forall i < |s|(\mu\tilde{s}_{i+2} > 0 \wedge \mu\tilde{s}_{i+2} = \mathrm{rem}(\mu\tilde{s}_i, \mu\tilde{s}_{i+1}))$, where $\tilde{s} := [e_0, e_1] * s$.

Then $|C|^{[]}$ trivially holds, and thus there is some minimal s satisfying $|C|^s$.

Suppose that $\mu\tilde{s}_l = q\mu\tilde{s}_{l+1} + r$ and define $\underline{r} = \tilde{s}_l - q\tilde{s}_{l+1}$. Then $s \succ s * \underline{r}$, but $\neg|C|^{s*\underline{r}}$ implies that $\mu\underline{r} = 0$ and thus $\mu\tilde{s}_{l+1} \mid \mu\tilde{s}_l$.

By induction we must have $\mu\tilde{s}_{l+1} \mid \mu\tilde{s}_i$ for all $i \leq l$ and thus $\mu\tilde{s}_{l+1} \mid \mu e_0, \mu e_1$ i.e. $\mu\tilde{s}_{l+1} \mid a, b$.

What is the computational content of this proof? The main non-constructive component is a single instance of the least element principle, so should essentially be a simple learning procedure modulo some details.

The formula $\forall a, b \exists m, n(am + bn \mid a, b)$ is realized by the program

$$\lambda a, b \,.\, \lim_{[]} \mathcal{L}_{a,b}$$

where $\mathcal{L}_{a,b} := (\mathsf{Good}, q, p_{a,b}, *)$ for

- $p_{a,b}(s) = \tilde{s}_l - \mathrm{quot}(\mu_{a,b}\tilde{s}_l, \mu_{a,b}\tilde{s}_{l+1})$ and $q = \tilde{s}_{l+1}$, and
- $\mathsf{Good}_{a,b}(s, t) :\equiv t \prec s \to \neg|C|^t$

Claim: $\mathcal{L}_{a,b}$ is just the Euclidean algorithm!

A core part of most learning-based computational interpretations of classical logic involves understanding how to deal with the so called Σ_1-LEM^{-}:

$$\forall n(\exists x|P_n|^x \vee \forall y \neg |P_n|^y).$$

Much work has been done recently on extracting programs from proofs relative to non-computable Skolem functions f_P for Σ_1-LEM^{-} satisfying

$$\exists f \forall n, y(|P_n|^{fn} \vee \neg |P_n|^y).$$

General idea behind ϵ-calculus and more modern approaches (e.g. 'interactive learning-based realizability' of Aschieri and Berardi'):

1. If $\mathsf{HA} + \Sigma_1$-$\mathsf{LEM}^{-} \vdash A$ then there is a realizing term t for A that is primitive recursive in Skolem functions f_P.

2. When A is e.g. Π_2 the term t only requires *approximations* to f_P. Therefore an approximation $t_0 \sqsubset t_1 \sqsubset \ldots \sqsubset t$ to t can be built by 'learning' a finite amount of information about f_P.

An equivalent problem involves giving a computational interpretation to the axiom of countable choice for Π^0_1-formulas:

$$\forall n \exists x \forall y |A_n|^x_y \to \exists f \forall n, y |A_n|^{f(n)}_y$$

This has a functional interpretation given by

$$\begin{aligned}
&\rightsquigarrow \forall n \neg\neg \exists x \forall y |A_n|^x_y \to \neg\neg \exists f \forall n, y |A_n|^{f(n)}_y \\
&\rightsquigarrow \forall n, p \exists x |A_n|^x_{p(x)} \to \forall \varphi, \phi \exists f |A_{\varphi f}|^{f(\varphi f)}_{\phi f} \\
&\rightsquigarrow \forall L, \varphi, \phi \exists f (\forall n, p |A_n|^{Lnp}_{p(Lnp)} \to |A_{\varphi f}|^{f(\varphi f)}_{\phi f})
\end{aligned}$$

Now suppose that the premise is realized by a countable sequence $(\mathcal{L}[n])_{n\in\mathbb{N}}$ of learning procedures i.e.

$$Lnp := \lim_{x_n} \mathcal{L}[n]_p$$

for $\mathcal{L}[n]_p := (\mathsf{Good}_n, q_n, p \circ q_n, \oplus_n)$ and $(x_n)_{n\in\mathbb{N}}$ some sequence of initial objects.

We define a 'product' $\mathcal{L}[\infty]$ of the $(\mathcal{L}[n])_{n\in\mathbb{N}}$, namely

$$\mathcal{L}[\infty]_{\varphi,\phi} := (\mathsf{Good}_\infty, q_\infty, [\varphi,\phi]\circ q_\infty, \oplus_\infty)$$

has type $X^{\mathbb{N}}, \mathbb{N}\times X$ and

- $f \oplus_\infty (n,x) := f[n \mapsto f(n)\oplus_n x]$;
- $\mathsf{Good}_\infty(f,[n,x]) :\equiv T_n(f(n),x)$; and
- $q_\infty(f) := \lambda n.q_n(f(n))$.

THEOREM. The formula

$$\forall\varphi,\phi\exists f(\forall n,p|A_n|^{\lim_{x_n}\mathcal{L}[n]_p}_{p(\lim_{x_n}\mathcal{L}[n]_p)} \to |A_{\varphi f}|^{f(\varphi f)}_{\phi f})$$

is realised by

$$\lambda\varphi,\phi\,.\,\lim_{f_0}\mathcal{L}[\infty]_{\varphi,\phi}$$

for $f_0 := \lambda n.x_n$.

Functional interpretation of $\neg\neg(\exists x|P_n|^x \vee \forall y \neg|P_n|^y)$ given by

$$\forall p \exists b, x(|P_n|^x \vee_b \neg|P_n|^{p(x)})$$

and is realised by $\lambda p.\, \lim_{(\bot,0)} \mathcal{L}[n]_p$ for $\mathcal{L}[n]_p := (\mathsf{Good}_n, \mathrm{id}, p, \pi_1)$ with

$$\mathsf{Good}_n((b,x),(b',y)) :\equiv b = \bot \to \neg|P_n|^y.$$

Therefore the functional interpretation of $\neg\neg\exists f \forall n(|P_n|^{f(n)} \vee \forall y \neg|P_n|^y)$

$$\forall \varphi, \phi \exists f(|P_{\varphi f}|^{f_1(\varphi f)} \vee_{f_0(\varphi f)} \neg|P_{\varphi f}|^{\phi f})$$

is realised by $\lambda \varphi, \phi.\, \lim \mathcal{L}[\infty]_{\varphi,\phi}$ for $\mathcal{L}[\infty]_{\varphi,\phi} := (\mathsf{Good}_\infty, \mathrm{id}, [\varphi,\phi], \oplus_\infty)$ where

- $f \oplus_\infty (n,x) := f[n \mapsto x]$ is a standard function update operation and
- $\mathsf{Good}_\infty(f,[n,(b,x)]) \Leftrightarrow n \notin \mathrm{dom}(f) \to \neg|P_n|^x$

Directions for research:

1. How does this compare to traditional interpretation of choice principles via recursors? Hopefully (P. 2016)!
2. Program extraction from proofs in mathematics: many theorems in subsystems of analysis use only Π^0_1-comprehension e.g. Ramsey's theorem (Kreuzer, Kohlenbach 2009). Can we give a learning based interpretation of these? What about more complex things like Kruskal's theorem?
3. Can we obtain complexity results analogous to those for epsilon calculus and interactive learning realizability? Potentially show that limits are computable using some kind of bar recursion of low type and adapt (Schwichtenberg 1979)...

Thank you!