The computational content of Zorn's lemma

Thomas Powell

University of Innsbruck

Classical Logic and Computation (CL&C'16)

Porto
Portugal
23 June 2016

A long time ago in 2012 I wrote a paper for CL&C called

Applying Gödel's Dialectica Interpretation to Obtain a Constructive Proof of Higman's Lemma

Unfortunately the extracted program was completely incomprehensible (to me, at least). This is because the classical proof of Higman's lemma uses a non-trivial form of Zorn's lemma, and the usual interpretation via bar recursion was both indirect and highly complex. There must be a better way!

Background to talk

A long time ago in 2012 I wrote a paper for CL&C called

Applying Gödel's Dialectica Interpretation to Obtain a Constructive Proof of Higman's Lemma

Unfortunately the extracted program was completely incomprehensible (to me, at least). This is because the classical proof of Higman's lemma uses a non-trivial form of Zorn's lemma, and the usual interpretation via bar recursion was both indirect and highly complex. There must be a better way!

In January 2016 I attended a Dagstuhl Seminar on *Well-Quasi Orderings in Computer Science*, which prompted me to revisit this paper.

Purpose of short talk

1. A brief discussion on Zorn's lemma and its functional interpretation.
2. Some conjectures and open questions (the most important part!)

First some undergraduate logic...

Let $(X, \sqsubset)$ be a partially ordered set.

Zorn's lemma. If every chain in X has an upper bound in X, then X contains a maximal element.

> *"The axiom of choice is is obviously true, the well-ordering theorem is obviously false, and who can tell about Zorn's lemma?"*

First some undergraduate logic...

Let $(X, \sqsubset)$ be a partially ordered set.

Zorn's lemma. If every chain in X has an upper bound in X, then X contains a maximal element.

> *"The axiom of choice is is obviously true, the well-ordering theorem is obviously false, and who can tell about Zorn's lemma?"*

Zorn's lemma is typically required when

- we are trying to build a structure in stages
- but there is no obvious way of completing the construction.

What is the computational meaning of an instance of Zorn's lemma?

We can often replace statements which assert the existence of an ideal object with an equivalent 'finitized', or 'hard' version which asserts the existence of finite approximations to this object.

What is the computational meaning of an instance of Zorn's lemma?

We can often replace statements which assert the existence of an ideal object with an equivalent 'finitized', or 'hard' version which asserts the existence of finite approximations to this object.

FINITARY ZORN'S LEMMA (ROUGH VERSION): If X is a (approximately) chain complete poset, then X contains approximations to a maximal element of arbitrarily high quality.

Moreover, we would expect that

- we can build an approximately maximal element in stages, which reflect the non-constructive stages of the original statement,
- the construction eventually terminates.

What is the computational meaning of an instance of Zorn's lemma?

We can often replace statements which assert the existence of an ideal object with an equivalent 'finitized', or 'hard' version which asserts the existence of finite approximations to this object.

FINITARY ZORN'S LEMMA (ROUGH VERSION): If X is a (approximately) chain complete poset, then X contains approximations to a maximal element of arbitrarily high quality.

Moreover, we would expect that

- we can build an approximately maximal element in stages, which reflect the non-constructive stages of the original statement,
- the construction eventually terminates.

Gödel's functional interpretation provides an opportunity to make some of this precise.

Example: Σ^0_1 comprehension over numbers

Suppose that we have a sequence of Σ^0_1-predicates $\exists x P(n,x)$. Then there exists a choice function $f : D \to \mathbb{N}$ with $D \subseteq \mathbb{N}$ satisfying

$$(*) \quad \forall n([n \in D \to P(n, f(n))] \wedge [n \notin D \to \forall x \neg P(n,x)]).$$

Example: Σ^0_1 comprehension over numbers

Suppose that we have a sequence of Σ^0_1-predicates $\exists x P(n,x)$. Then there exists a choice function $f : D \to \mathbb{N}$ with $D \subseteq \mathbb{N}$ satisfying

$$(*) \quad \forall n([n \in D \to P(n, f(n))] \wedge [n \notin D \to \forall x \neg P(n,x)]).$$

Proof. Take X to be the poset of partial choice functions

$$\{g : D \to \mathbb{N} \mid \forall n [n \in D \to P(n, g(n))]\}$$

ordered by function extension. Then X is chain complete and so has a maximal element f, which must satisfy $(*)$.

Example: Σ^0_1 comprehension over numbers

Suppose that we have a sequence of Σ^0_1-predicates $\exists x P(n,x)$. Then there exists a choice function $f : D \to \mathbb{N}$ with $D \subseteq \mathbb{N}$ satisfying

$$(*)\quad \forall n([n \in D \to P(n, f(n))] \wedge [n \notin D \to \forall x \neg P(n,x)]).$$

Proof. Take X to be the poset of partial choice functions

$$\{g : D \to \mathbb{N} \mid \forall n[n \in D \to P(n, g(n))]\}$$

ordered by function extension. Then X is chain complete and so has a maximal element f, which must satisfy $(*)$.

If not, then there exist some $n \notin D$ and x such that $P(n,x)$. Then

$$f \cup \{n \mapsto x\}$$

belongs to X and extends f, a contradiction.

The functional interpretation of $(*)$ states that for any counterexample functionals $\varphi, \xi : (\mathbb{N} \rightharpoonup \mathbb{N}) \to \mathbb{N}$ there exists an *approximation* $f : D \to X$ to a choice function satisfying

$$(\dagger) \quad [\varphi(f) \in D \to P(\varphi(f), f(\varphi(f)))] \wedge [\varphi(f) \notin D \to \neg P(\varphi(f), \xi(f))].$$

The functional interpretation of $(*)$ states that for any counterexample functionals $\varphi, \xi : (\mathbb{N} \rightharpoonup \mathbb{N}) \to \mathbb{N}$ there exists an *approximation* $f : D \to X$ to a choice function satisfying

$$(\dagger) \quad [\underbrace{\varphi(f)}_{n} \in D \to P(\underbrace{\varphi(f)}_{n}, \underbrace{f(\varphi(f))}_{f(n)})] \wedge [\underbrace{\varphi(f)}_{n} \notin D \to \neg P(\underbrace{\varphi(f)}_{n}, \underbrace{\xi(f)}_{x})].$$

The functional interpretation of $(*)$ states that for any counterexample functionals $\varphi, \xi : (\mathbb{N} \rightharpoonup \mathbb{N}) \to \mathbb{N}$ there exists an *approximation* $f : D \to X$ to a choice function satisfying

$$(\dagger) \quad [\underbrace{\varphi(f)}_{n} \in D \to P(\underbrace{\varphi(f)}_{n}, \underbrace{f(\varphi(f))}_{f(n)})] \wedge [\underbrace{\varphi(f)}_{n} \notin D \to \neg P(\underbrace{\varphi(f)}_{n}, \underbrace{\xi(f)}_{x})].$$

Moreover we can construct such an f explicitly. Define the sequence $f_i : D_i \to \mathbb{N}$ recursively as follows:

1. f_0 is the empty partial function, and
2. if $\varphi(f_i) \notin D_i$ and $P(\varphi(f_i), \xi(f_i))$ then

$$f_{i+1} := f_i \cup \{\varphi(f_i) \mapsto \xi(f_i)\}.$$

The functional interpretation of $(*)$ states that for any counterexample functionals $\varphi, \xi : (\mathbb{N} \rightharpoonup \mathbb{N}) \to \mathbb{N}$ there exists an *approximation* $f : D \to X$ to a choice function satisfying

$$(\dagger) \quad [\underbrace{\varphi(f)}_{n} \in D \to P(\underbrace{\varphi(f)}_{n}, \underbrace{f(\varphi(f))}_{f(n)})] \wedge [\underbrace{\varphi(f)}_{n} \notin D \to \neg P(\underbrace{\varphi(f)}_{n}, \underbrace{\xi(f)}_{x})].$$

Moreover we can construct such an f explicitly. Define the sequence $f_i : D_i \to \mathbb{N}$ recursively as follows:

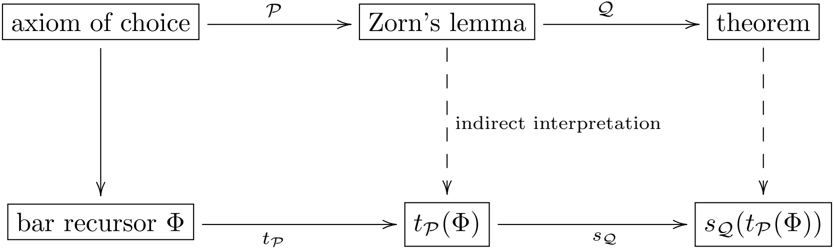
1. f_0 is the empty partial function, and
2. if $\varphi(f_i) \notin D_i$ and $P(\varphi(f_i), \xi(f_i))$ then

$$f_{i+1} := f_i \cup \{\varphi(f_i) \mapsto \xi(f_i)\}.$$

For each i, the approximation $f_{i+1} \sqsupset f_i$ is an 'improvement' of the approximation f_i. Whenever φ, ξ are continuous, we eventually reach a sufficiently good approximation f_I satisfying $(\dagger)$.

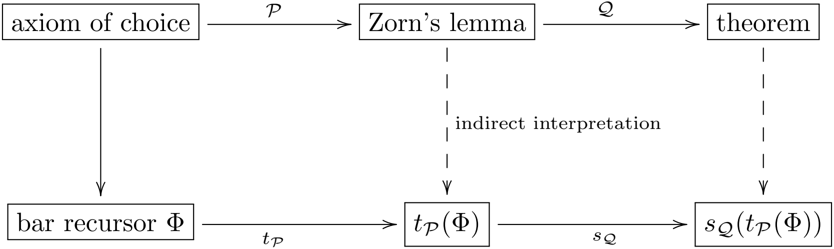
... however, traditionally bar recursion is used to realize the functional interpretation of comprehension, and the resulting program is much less natural. This is a general problem:

... however, traditionally bar recursion is used to realize the functional interpretation of comprehension, and the resulting program is much less natural. This is a general problem:



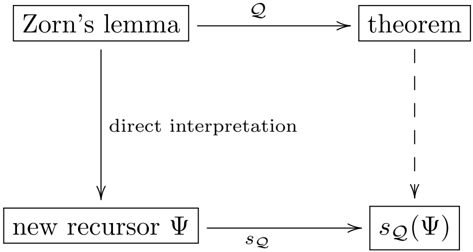
... however, traditionally bar recursion is used to realize the functional interpretation of comprehension, and the resulting program is much less natural. This is a general problem:

KEY QUESTION: Is there a useful form of Zorn's lemma to which we can give a direct computational interpretation? We want extracted programs which reflect the idea that Zorn's lemma allows us to build objects in stages.



... however, traditionally bar recursion is used to realize the functional interpretation of comprehension, and the resulting program is much less natural. This is a general problem:

KEY QUESTION: Is there a useful form of Zorn's lemma to which we can give a direct computational interpretation? We want extracted programs which reflect the idea that Zorn's lemma allows us to build objects in stages.



Let $(X, <)$ be a **well-founded** partial order, and consider the partial order $(\mathbb{N} \to X, \ <_{\text{lex}} \ )$, where

$$f <_{\text{lex}} g := \exists n(\bar{f}n = \bar{g}n \wedge f(n) < g(n))$$

where $\bar{f}n = [f(0), \ldots, f(n-1)]$.

Let $(X, <)$ be a **well-founded** partial order, and consider the partial order $(\mathbb{N} \to X, <_{\text{lex}})$, where

$$f <_{\text{lex}} g := \exists n(\bar{f}n = \bar{g}n \wedge f(n) < g(n))$$

where $\bar{f}n = [f(0), \ldots, f(n-1)]$.

A predicate on $\mathbb{N} \to X$ is piecewise if it is of the form $\forall n\, B(\bar{f}n)$. Consider

$$Y := \{f : \mathbb{N} \to X \mid \forall n\, B(\bar{f}n)\}$$

Then whenever Y is non-empty, it is chain-complete w.r.t. $<_{\text{lex}}$ (in the downward direction), and therefore by Zorn's lemma it has a maximal (in this case minimal) element.

Let $(X, <)$ be a **well-founded** partial order, and consider the partial order $(\mathbb{N} \to X, <_{\text{lex}})$, where

$$f <_{\text{lex}} g := \exists n (\bar{f}n = \bar{g}n \wedge f(n) < g(n))$$

where $\bar{f}n = [f(0), \ldots, f(n-1)]$.

A predicate on $\mathbb{N} \to X$ is piecewise if it is of the form $\forall n\, B(\bar{f}n)$. Consider

$$Y := \{f : \mathbb{N} \to X \mid \forall n\, B(\bar{f}n)\}$$

Then whenever Y is non-empty, it is chain-complete w.r.t. $<_{\text{lex}}$ (in the downward direction), and therefore by Zorn's lemma it has a maximal (in this case minimal) element.

We can write this as an axiom schema

$$\mathsf{ZL}_{\text{lex}} : \exists f A(f) \to \exists f (A(f) \wedge \forall g <_{\text{lex}} f \neg A(g)).$$

where $A(f) \equiv \forall n B(\bar{f}n)$ ranges over piecewise formulas.

Why is ZL_{lex} *so useful?*

It is a weak form of Zorn's lemma (equivalent to dependent choice) which subsumes many instances of Zorn's lemma used in everyday mathematics.

Why is ZL_{lex} *so useful?*

It is a weak form of Zorn's lemma (equivalent to dependent choice) which subsumes many instances of Zorn's lemma used in everyday mathematics.

- Let $X = Y \cup \bot$ with $x < y$ iff x defined and y undefined. Then $<_{\text{lex}}$ contains the extension relation on partial functions $\mathbb{N} \to Y$, and in particular ZL_{lex} implies arithmetical comprehension.

Why is ZL_{lex} so useful?

It is a weak form of Zorn's lemma (equivalent to dependent choice) which subsumes many instances of Zorn's lemma used in everyday mathematics.

- Let $X = Y \cup \perp$ with $x < y$ iff x defined and y undefined. Then $<_{\text{lex}}$ contains the extension relation on partial functions $\mathbb{N} \to Y$, and in particular ZL_{lex} implies arithmetical comprehension.
- If $X = \mathbb{B}$ and $x < y$ iff $x = 1$ and $x = 0$, and we view objects of type $\mathbb{N} \to \mathbb{B}$ as characteristic functions for subsets of $\mathbb{N}$, then $<_{\text{lex}}$ contains the subset relation $\subset$ and ZL_{lex} can be used to prove standard properties of countable algebraic structures e.g. every countable (nonzero) ring has a maximal ideal.

Why is ZL_{lex} *so useful?*

It is a weak form of Zorn's lemma (equivalent to dependent choice) which subsumes many instances of Zorn's lemma used in everyday mathematics.

- Let $X = Y \cup \bot$ with $x < y$ iff x defined and y undefined. Then $<_{\mathrm{lex}}$ contains the extension relation on partial functions $\mathbb{N} \to Y$, and in particular $\mathsf{ZL}_{\mathrm{lex}}$ implies arithmetical comprehension.
- If $X = \mathbb{B}$ and $x < y$ iff $x = 1$ and $x = 0$, and we view objects of type $\mathbb{N} \to \mathbb{B}$ as characteristic functions for subsets of $\mathbb{N}$, then $<_{\mathrm{lex}}$ contains the subset relation $\subset$ and $\mathsf{ZL}_{\mathrm{lex}}$ can be used to prove standard properties of countable algebraic structures e.g. every countable (nonzero) ring has a maximal ideal.
- We can also give very short proofs of results in WQO theory, such as the existence of a minimal bad sequence, used to prove Higman's lemma and Kruskal's theorem.

Because chain-completeness is encoded in the assumption that $A(f)$ is piecewise, $\mathsf{ZL}_{\mathrm{lex}}$ is essentially a minimum principle over the non well-founded order $<_{\mathrm{lex}}$. The usual minimum principle for well-founded decidable relations:

$$\exists x A(x) \to \exists x (A(x) \to \forall y < x \neg A(y))$$

has a functional interpretation which can be solved as long as we have access to a well-founded recursor of *arbitrary* finite output type:

$$\mathsf{R}^F_<(x) = F_x(\lambda y < x \,.\, \mathsf{R}_<(y))$$

Details are given in Schwichtenberg 2008.

Because chain-completeness is encoded in the assumption that $A(f)$ is piecewise, ZL_{lex} is essentially a minimum principle over the non well-founded order $<_{\text{lex}}$. The usual minimum principle for well-founded decidable relations:

$$\exists x A(x) \to \exists x (A(x) \to \forall y < x \neg A(y))$$

has a functional interpretation which can be solved as long as we have access to a well-founded recursor of *arbitrary* finite output type:

$$\mathsf{R}^F_<(x) = F_x(\lambda y < x \,.\, \mathsf{R}_<(y))$$

Details are given in Schwichtenberg 2008.

The general idea is that recursion over $<$ is used to construct an approximation to a minimal element, by guessing a possible candidates and using any mistakes to improve each guess.

Can the functional interpretation of $\mathsf{ZL}_{\mathrm{lex}}$ i.e.

$$\exists f A(f) \to \exists f (A(f) \wedge \forall g <_{\mathrm{lex}} f \neg A(g))$$

for piecewise formulas A, be solved using a lexicographic recursor:

$$\mathsf{ZR}^F_{\mathrm{lex}}(f) = F_f(\lambda n, x < f(n), g \,.\, \mathsf{ZR}_{\mathrm{lex}}(\bar{f}n * x * g))?$$

Can the functional interpretation of ZL_{lex} i.e.

$$\exists f A(f) \to \exists f (A(f) \wedge \forall g <_{\text{lex}} f \neg A(g))$$

for piecewise formulas A, be solved using a lexicographic recursor:

$$\mathsf{ZR}^F_{\text{lex}}(f) = F_f(\lambda n, x < f(n), g \,.\, \mathsf{ZR}_{\text{lex}}(\bar{f}n * x * g))?$$

This kind of problem was first considered by U. Berger, where the contrapositive of ZL_{lex} (called open induction) is given a modified realizability interpretation using ZR_{lex} restricted to base type.

Can the functional interpretation of ZL_{lex} i.e.

$$\exists f A(f) \to \exists f (A(f) \wedge \forall g <_{\text{lex}} f \neg A(g))$$

for piecewise formulas A, be solved using a lexicographic recursor:

$$\mathsf{ZR}^F_{\text{lex}}(f) = F_f(\lambda n, x < f(n), g \,.\, \mathsf{ZR}_{\text{lex}}(\bar{f}n * x * g))?$$

This kind of problem was first considered by U. Berger, where the contrapositive of ZL_{lex} (called open induction) is given a modified realizability interpretation using ZR_{lex} restricted to base type.

The functional interpretation of ZL_{lex} is unfortunately significantly more complex. In particular, we need a lexicographic recursor of *arbitrary* finite type, in which case ZR_{lex} no longer defines a total continuous functional.

I won't go into any real detail, but will sketch how one can overcome this particular difficulty.

Suppose that $G : Y^{\mathbb{N}} \to \mathbb{N}$ is continuous. Then for any $\alpha : Y^{\mathbb{N}}$ there is some point n such that

$$G(\overline{\alpha, n}) < n$$

where $\overline{\alpha, n} = \alpha(0), \ldots, \alpha(n-1), 0, 0, 0, \ldots$.

Suppose that $G : Y^{\mathbb{N}} \to \mathbb{N}$ is continuous. Then for any $\alpha : Y^{\mathbb{N}}$ there is some point n such that

$$G(\overline{\alpha, n}) < n$$

where $\overline{\alpha, n} = \alpha(0), \ldots, \alpha(n-1), 0, 0, 0, \ldots$.

Let $N_{G,\alpha}$ be the least such n. Then $\overline{\alpha, N_{G,\alpha}}$ is primitive recursively definable as

$$\overline{\alpha, N_{G,\alpha}} = \lambda m. \begin{cases} \alpha(m) & \text{if } \forall k \leq m(G(\overline{\alpha, k}) \geq k) \\ 0_Y & \text{if } \exists k \leq m(G(\overline{\alpha, k}) < k) \end{cases}$$

Suppose that $G : Y^{\mathbb{N}} \to \mathbb{N}$ is continuous. Then for any $\alpha : Y^{\mathbb{N}}$ there is some point n such that

$$G(\overline{\alpha, n}) < n$$

where $\overline{\alpha, n} = \alpha(0), \ldots, \alpha(n-1), 0, 0, 0, \ldots$.

Let $N_{G,\alpha}$ be the least such n. Then $\overline{\alpha, N_{G,\alpha}}$ is primitive recursively definable as

$$\overline{\alpha, N_{G,\alpha}} = \lambda m. \begin{cases} \alpha(m) & \text{if } \forall k \leq m(G(\overline{\alpha, k}) \geq k) \\ 0_Y & \text{if } \exists k \leq m(G(\overline{\alpha, k}) < k) \end{cases}$$

Define the lexicographic recursor $\widehat{\mathsf{ZR}}_{\mathrm{lex}}$ by

$$\widehat{\mathsf{ZR}}^{F,G}_{\mathrm{lex}}(f) = F(\overline{\underbrace{[f \mid \lambda n, x < f(n), g \mathbin{.} \widehat{\mathsf{ZR}}^{F,G}_{\mathrm{lex}}(\bar{f}n * x * g)]}_{\alpha}, N_{G,\alpha}}).$$

Suppose that $G : Y^{\mathbb{N}} \to \mathbb{N}$ is continuous. Then for any $\alpha : Y^{\mathbb{N}}$ there is some point n such that

$$G(\overline{\alpha, n}) < n$$

where $\overline{\alpha, n} = \alpha(0), \ldots, \alpha(n-1), 0, 0, 0, \ldots$.

Let $N_{G,\alpha}$ be the least such n. Then $\overline{\alpha, N_{G,\alpha}}$ is primitive recursively definable as

$$\overline{\alpha, N_{G,\alpha}} = \lambda m. \begin{cases} \alpha(m) & \text{if } \forall k \leq m (G(\overline{\alpha, k}) \geq k) \\ 0_Y & \text{if } \exists k \leq m (G(\overline{\alpha, k}) < k) \end{cases}$$

Define the lexicographic recursor $\widehat{\mathsf{ZR}}_{\text{lex}}$ by

$$\widehat{\mathsf{ZR}}_{\text{lex}}^{F,G}(f) = F(\overline{\underbrace{[f \mid \lambda n, x < f(n), g \,.\, \widehat{\mathsf{ZR}}_{\text{lex}}^{F,G}(\bar{f}n * x * g)]}_{\alpha}, N_{G,\alpha}}).$$

Here, F is forced by G to only look at a finite initial segment of its input, making totality of $\widehat{\mathsf{ZR}}_{\text{lex}}$ a piecewise property. Thus $\widehat{\mathsf{ZR}}_{\text{lex}}$ defines a total functional by Zorn's lemma!

Claims

Claims

1. There is a term Φ definable in $T + \widehat{\mathsf{ZR}}_{\text{lex}}$ which solves the functional interpretation of ZL_{lex}.

Claims

1. There is a term Φ definable in $T + \widehat{\mathsf{ZR}}_{\text{lex}}$ which solves the functional interpretation of ZL_{lex}.

2. The recursor $\widehat{\mathsf{ZR}}_{\text{lex}}$ (and hence Φ) is definable from Spector's bar recursion over system T, and thus exists (and is S1-S9 computable) in the usual model $\mathcal{C}^\omega$ of total continuous functionals, and also the strongly majorizable functionals $\mathcal{M}^\omega$.

Claims

1. There is a term Φ definable in $T + \widehat{\mathsf{ZR}}_{\text{lex}}$ which solves the functional interpretation of ZL_{lex}.

2. The recursor $\widehat{\mathsf{ZR}}_{\text{lex}}$ (and hence Φ) is definable from Spector's bar recursion over system T, and thus exists (and is S1-S9 computable) in the usual model $\mathcal{C}^\omega$ of total continuous functionals, and also the strongly majorizable functionals $\mathcal{M}^\omega$.

3. The realzing term Φ computes the limit of a learning procedure as defined in (P. 2016), which can be intuitively seen as computing an approximation to a lexicographically minimal element via trial and error.

Claims

1. There is a term Φ definable in $T + \widehat{\mathsf{ZR}}_{\text{lex}}$ which solves the functional interpretation of ZL_{lex}.

2. The recursor $\widehat{\mathsf{ZR}}_{\text{lex}}$ (and hence Φ) is definable from Spector's bar recursion over system T, and thus exists (and is S1-S9 computable) in the usual model $\mathcal{C}^\omega$ of total continuous functionals, and also the strongly majorizable functionals $\mathcal{M}^\omega$.

3. The realzing term Φ computes the limit of a learning procedure as defined in (P. 2016), which can be intuitively seen as computing an approximation to a lexicographically minimal element via trial and error.

4. In many cases, programs extracted from proofs involving ZL_{lex} will be shorter, easier to understand and potentially more efficient than those which would be normally extracted using e.g. bar recursion.

Can we back up the last claim with some concrete case studies?

Can we back up the last claim with some concrete case studies?

Constructive algebra. Plenty of results in this area make use of $\mathsf{ZL}_{\mathrm{lex}}$ in some form. The computational content of such results have been studied by e.g. T. Coquand, H. Lombardi, P. Schuster.

- The functional interpretation together with $\widehat{\mathsf{ZR}}_{\mathrm{lex}}$ could be well suited to program extraction in algebra.
- Can we establish a relationship between weak instances of $\widehat{\mathsf{ZR}}_{\mathrm{lex}}$ and bar recursion of low type, so that we can obtain e.g. primitive recursive bounds for witnesses of Π^0_2-formulas?

Can we back up the last claim with some concrete case studies?

CONSTRUCTIVE ALGEBRA. Plenty of results in this area make use of $\mathsf{ZL}_{\mathrm{lex}}$ in some form. The computational content of such results have been studied by e.g. T. Coquand, H. Lombardi, P. Schuster.

- The functional interpretation together with $\widehat{\mathsf{ZR}}_{\mathrm{lex}}$ could be well suited to program extraction in algebra.
- Can we establish a relationship between weak instances of $\widehat{\mathsf{ZR}}_{\mathrm{lex}}$ and bar recursion of low type, so that we can obtain e.g. primitive recursive bounds for witnesses of Π^0_2-formulas?

WQO THEORY. It would be good to extract a short, clean program from the usual proof of Higman's lemma (this was the original aim, after all)!

- What about Kruskal's theorem?
- In general there is a lot of interest in constructive WQO theory from computer science, where results like Higman's lemma are used to prove termination of programs...

What is going on more generally? Can we give a functional interpretation to stronger instances of Zorn's lemma?