On the Computational content of Brouwer's Thesis

Ulrich Berger
Swansea University

Workshop "Constructive Mathematics"
Hausdorff Trimester "Types, Sets and Constructions"
Bonn, May 2 - August 24, 2018

Overview

1. Brouwer's thesis
2. Abstract bar induction
3. Proving uniform continuity
4. Extracting the fan functional

Brouwer's thesis

Brouwer's thesis (**BT**) Every bar is inductive.

A predicate P on natural numbers is a *bar* if $\forall\alpha\exists n\, P(\bar{\alpha}\, n)$

P is an *inductive bar* if $\mathbf{IB}_P(\langle\rangle)$ holds where, inductively,

(i) If $P(s)$, then $\mathbf{IB}_P(s)$.

(ii) If $\mathbf{IB}_P(s * n)$ for all $n \in \mathbf{N}$, then $\mathbf{IB}_P(s)$.

More compactly,

$$\mathbf{IB}_P(s) \stackrel{\mu}{=} P(s) \vee \forall n\, \mathbf{IB}_P(s * n) \qquad (\mu \text{ means 'least'})$$

Hence **BT** can be written as the schema

$$\forall\alpha\exists n\, P(\bar{\alpha}\, n) \to \mathbf{IB}_P(\langle\rangle)$$

Bar induction for decidable bars (**BI**)

If

(1) P is a bar,

(2) P decidable and $P \subseteq Q$,

(3) $\forall s\,(\forall n\, Q(s * n) \to Q(s))$,

then $Q(\langle\rangle)$.

Where 'P decidable' means $\forall n\,(P(n) \vee \neg P(n))$.

Issues with **BT** and **BI** (regarding applicability)

BT $\forall\alpha\exists n\, P(\bar{\alpha}\, n) \to \mathbf{IB}_P(\langle\rangle)$

- restricted to natural numbers
- talks about infinite sequences
- the premise has computational content which is often not available
- the conclusion has unwanted computational content
- decidability of the bar P (required in **BI**) is too restrictive

Therefore, we *weaken* and *generalize* premise and conclusion.

Paths and accessibility

Let $\prec$ be an arbitrary binary relation.

$$\mathbf{Path}_{\prec}(x) \stackrel{\nu}{=} \exists y \prec x\, \mathbf{Path}_{\prec}(y) \qquad (\nu \text{ means 'greatest'})$$
$$\mathbf{Acc}_{\prec}(x) \stackrel{\mu}{=} \forall y \prec x\, \mathbf{Acc}_{\prec}(y)$$

Classically, $\mathbf{Path}_{\prec}$ and $\mathbf{Acc}_{\prec}$ are complements of each other.

$\mathbf{Path}_{\prec}(x)$ means (with dependent choice) that there is an infinite $\prec$-descending sequence starting with x.

$\mathbf{Acc}_{\prec}(x)$ means that $\prec$-induction is valid at x.

Setting $s \prec_P t \stackrel{\mathrm{Def}}{=} \exists n\, s = t * n \wedge \neg P(t)$:

$\neg\mathbf{Path}_{\prec_P}(\langle\rangle)$ means that P is a bar,

$\mathbf{Acc}_{\prec_P}(\langle\rangle)$ means that P is an inductive bar, provided P is decidable.

Brouwer's thesis without computational content

The implication $\mathbf{Acc}_{\prec}(x) \to \neg\mathbf{Path}_{\prec}(x)$ is intuitionistically valid (easy $\prec$-induction).

The converse is can be viewed as a version of Brouwer's thesis:

$\mathbf{BT}_0$ $\forall x\,(\neg\mathbf{Path}_{\prec}(x) \to \mathbf{Acc}_{\prec}(x))$

Both, the premise and conclusion of $\mathbf{BT}_0$, are Harrop formulas (do not contain $\vee$ or $\exists$ at a strictly positive position).

Therefore, $\mathbf{BT}_0$ has no computational content and hence does not spoil program extraction.

The premise, $\neg\mathbf{Path}_{\prec}(x)$, can be replaced by any equivalent Harrop property, for example,

$$\neg\exists f\,(f\,0 = x \wedge \forall n \in \mathbf{N}\, f(n+1) \prec f\,n)$$

Abstract bar induction (**ABI**)

$$y \prec^* x \quad \stackrel{\mu}{=} \quad y = x \vee \exists z\, (y \prec^* z \wedge z \prec x) \qquad \text{(refl. trans. closure)}$$
$$y \prec_P x \quad \stackrel{\text{Def}}{=} \quad y \prec x \wedge \neg P(x)$$

Let x_0 be arbitrary (playing the role of the empty sequence).

ABI If

(1) $\neg\mathbf{Path}_{\prec_P}(x_0)$
(2) $\forall x \prec^* x_0\, (\neg P(x) \vee Q(x))$,
(3) $\forall x \prec^* x_0\, (\forall y \prec x\, Q(y) \to Q(x))$,

then $Q(x_0)$.

Lemma. $\mathbf{BT}_0$ implies **ABI**.
Proof. Assume (1), (2), (3). By $\mathbf{BT}_0$, $\mathbf{Acc}_{\prec_P}(x_0)$. We prove $\mathbf{Acc}_{\prec_P} \subseteq Q$ by wellfounded induction. By i.h., $\forall y \prec_P^* x\, Q(y)$. We have to show $Q(x)$. We do a case analysis according to (2). If $Q(x)$, we are done. If $\neg P(x)$ then the i.h. is equivalent to the premise of (3), hence, again $Q(x)$.

Bang!

If A is a formula, then $!A$ is a Harrop formula with $a\,\mathbf{r}\,!A \overset{\mathrm{Def}}{=} a = \mathbf{Nil} \land \forall a\,(a\,\mathbf{r}\,A)$.

For example, $\mathbf{Nil}\,\mathbf{r}\,!(\bot \to A)$ since, $a\,\mathbf{r}\,(\bot \to A) \equiv \bot \to a\,\mathbf{r}\,A$.

Intuitively, $!A$ expresses that A is true (realizable) for trivial reasons.

Valid (realizable) rules we will use in the following:

$$\frac{A}{!A}\ !\mathbf{H} \qquad (A \text{ Harrop})$$

$$\frac{A \to\ !B}{!(A \to B)}\ !{\to} \qquad \frac{!A \land !B}{!(B \land A)}\ !{\land}$$

$$\frac{\forall x\, !A(x)}{!\forall x\, A(x)}\ !\forall \qquad \frac{\exists x\, !A(x)}{!\exists x\, A(x)}\ !\exists$$

!**LEM**

$$\frac{\neg A \to B \qquad A \to !B}{B}\ !\mathbf{LEM}$$

Lemma

The rules for bang are realizable.

Proof.

We only look at !**LEM**.

Assume $a\,\mathbf{r}\,(\neg A \to B)$ and $\mathbf{Nil}\,\mathbf{r}\,(A \to !B)$, that is,
$\neg\exists c\, c\,\mathbf{r}\,A \to a\,\mathbf{r}\,B$ and $\exists c\, c\,\mathbf{r}\,A \to \forall b\, b\,\mathbf{r}\,B$.

Using the law of excluded middle, we conclude $a\,\mathbf{r}\,B$. □

Banged bar induction

!BI If

(1) $\neg\mathbf{Path}_{\prec_P}(x_0)$,
(2) $\forall x \prec^* x_0 \, (P(x) \to !Q(x))$,
(3) $\forall x \prec^* x_0 \, (\forall y \prec x \, Q(y) \to Q(x))$,

then $Q(x_0)$.

Lemma

$\mathbf{BT}_0$ *implies* !**BI**.

Proof.

The proof is almost identical to the proof for **ABI**. The only difference is that we use !**LEM** to do a case analysis, on whether $P(x)$ holds, using (2). □

The extracted program takes as input a realizer g of (3) (note that (2) is Harrop) and returns $h \, \langle\rangle$ where

$$h \, s = g \, s \, (\lambda a \, (h \, (s * a))).$$

Proving uniform continuity

We aim to prove that every total continuous functional F on Cantor space is uniformly continuous and extract from the proof the *fan functional* that computes the minimal modulus of uniform continuity of F.

Language:

Constants: $0, 1, \bot$, where $0, 1$ represent at the same time the first two natural numbers and the Booleans, and $\bot$ represents 'undefined' (not to be confused with the formula $\bot$).

Function symbols: $+, -$, application operation (written by juxtaposition), common (primitive recursive) operations to define finite and infinite sequences.

Relation symbol: $<$ (ordinary ordering of numbers).

Natural numbers: $\mathbf{N}(x) \stackrel{\mu}{=} x = 0 \vee \mathbf{N}(x-1)$.

Partial functionals

We define the partial Booleans and partial Boolean-valued functionals of type 1 and 2:

$$
\begin{aligned}
\mathbb{B}(x) &\overset{\mathrm{Def}}{=} x = 0 \vee x = 1 \\
\mathbb{B}_\perp(x) &\overset{\mathrm{Def}}{=} x \neq \perp \to \mathbb{B}(x) \\
\mathbb{B}^1_\perp(\alpha) &\overset{\mathrm{Def}}{=} \forall n\, (\mathbf{N}(n) \to \mathbb{B}_\perp(\alpha\, n)) \\
\mathbb{B}^2_\perp(F) &\overset{\mathrm{Def}}{=} \forall \alpha\, (\mathbb{B}^1_\perp(\alpha) \to \mathbb{B}_\perp(F\alpha))
\end{aligned}
$$

For the following it wouldn't make much difference if the result predicate of F were $\mathbf{N}_\perp$ (instead of $\mathbb{B}_\perp$).

Continuity

Specialization order:

$$
\begin{aligned}
x \sqsubseteq y &\overset{\text{Def}}{=} x \neq \bot \to x = y \\
\alpha \sqsubseteq \beta &\overset{\text{Def}}{=} \forall n \in \mathbf{N}\, (\alpha\, n \sqsubseteq \beta\, n)
\end{aligned}
$$

Monotonicity, *finitarity*, *continuity*:

$$
\begin{aligned}
\mathbf{Mon}(F) &\overset{\text{Def}}{=} \forall \alpha, \beta \in \mathbb{B}^1_\bot\, (\alpha \sqsubseteq \beta \to F\,\alpha \sqsubseteq F\,\beta) \\
\mathbf{Fin}(F) &\overset{\text{Def}}{=} \forall \alpha \in \mathbb{B}^1_\bot\, (\forall n \in \mathbf{N}\, F\,(\alpha \uparrow n) = \bot \to F\,\alpha = \bot) \\
\mathbf{Cont}(F) &\overset{\text{Def}}{=} \mathbf{Mon}(F) \wedge \mathbf{Fin}(F)
\end{aligned}
$$

where $(\alpha \uparrow n)\, k = \textbf{if}\ k < n\ \textbf{then}\ \alpha\, k\ \textbf{else}\ \bot$.

Totality

$$\textbf{Total}^1(\alpha) \stackrel{\mathrm{Def}}{=} \forall n\,(\mathbf{N}(n) \to \alpha\, n \neq \bot)$$

$$\textbf{Total}^2(F) \stackrel{\mathrm{Def}}{=} \forall \alpha\,(\textbf{Total}^1(\alpha) \to F\alpha \neq \bot))$$

$$\mathbb{B}^1(\alpha) \stackrel{\mathrm{Def}}{=} \mathbb{B}^1_\bot(\alpha) \wedge \textbf{Total}^1(\alpha)$$

$$\mathbb{B}^2(F) \stackrel{\mathrm{Def}}{=} \mathbb{B}^2_\bot(F) \wedge \textbf{Total}^1(F)$$

Uniform continuity

A type 2 functional F is *uniformly continuous* if there is (a least) $n \in \mathbf{N}$ such that $F\,\alpha = F\,\beta$ for all total α, β agreeing below n.

$$\begin{aligned} \mathbf{UCont}(F, n) &\stackrel{\text{Def}}{=} \forall \alpha, \beta \in \mathbb{B}^1 (\alpha =_n \beta \to F\,\alpha = F\,\beta) \\ \mathbf{UCont}(F) &\stackrel{\text{Def}}{=} \exists n \in \mathbf{N}\, \mathbf{UCont}(F, n) \end{aligned}$$

where $\alpha =_n \beta \stackrel{\text{Def}}{=} \forall k \in \mathbf{N}\,(k < n \to \alpha\, k = \beta\, k)$.

We aim to prove that every $F \in \mathbb{B}^2_\perp$ which is total and continuous is uniformly continuous.

Deciding constancy

Let $\mathbb{B}^*$ be the set of finite sequences of Booleans, that is,

$$\mathbb{B}^*(s) \overset{\mu}{=} s = \langle\rangle \vee \exists t \in \mathbb{B}^* \, \exists b \in \mathbb{B} \, s = t * b,$$

and set

$$\mathbf{Const}(F, s) \overset{\text{Def}}{=} \exists b \in \mathbb{B} \, \forall \alpha \in \mathbb{B}^1 \, F(s * \alpha) = b$$

where $(s * \alpha)\, n = s_n$ if $n < |s|$ and $(s * \alpha)\, n = \alpha(n - |s|)$ if $n \geq |s|$.

Theorem (Decidability of constancy)

Let F be a total continuous functional, that is, $F \in \mathbb{B}^2$ and $\mathbf{Cont}(F)$. Then for every $s \in \mathbb{B}^s$ it is decidable whether F is constant on total extensions of s, that is, $\mathbf{Const}(F, s) \vee \neg\mathbf{Const}(F, s)$.

Deciding constancy

Proof.

We fix a total continuous functional F and define

$$\mathbf{sec}(s) \stackrel{\text{Def}}{=} F(s * \bot^\omega) \neq \bot \quad (\text{'}s\text{ is secured'})$$
$$s \prec t \stackrel{\text{Def}}{=} \exists b \in \mathbb{B}\, s = t * b$$

Hence $\mathbb{B}^*(s)$ iff $s \prec^* \langle\rangle$.

We define a version of the drinker formula:

$$\mathbf{Dr}(s, b, \alpha) \stackrel{\text{Def}}{=} F(s{*}\alpha) \neq \bot \wedge (\exists \beta \in \mathbb{B}^1_\bot\, F(s{*}\beta) = b \to F(s{*}\alpha) = b)$$

and set $Q_b(s) \stackrel{\text{Def}}{=} \exists \alpha \in \mathbb{B}^1_\bot\, \mathbf{Dr}(s, b, \alpha)$.

Claim: $\forall s \in \mathbb{B}^*\, Q_b(s)$ holds for every $b \in \mathbb{B}$.
Fix $b \in \mathbb{B}$. We prove the claim by banged bar induction on $\prec_{\mathbf{sec}}$.

Applying !**BI**

We have to show

(1) $\forall s \in \mathbb{B}^* \neg \mathbf{Path}_{\prec_{\mathbf{sec}}}(s)$,

(2) $\forall s \in \mathbb{B}^* (\mathbf{sec}(s) \to {!}Q_b(s))$,

(3) $\forall s \in \mathbb{B}^* \, (\forall a \in \mathbb{B} \, Q_b(s * a) \to Q_b(s))$,

(1) holds F since is total and continuous.

(2): By efq, $!{\to}$, and $!\forall$, $!\mathbb{B}^1_\perp(\perp^\omega)$. If $s \in \mathbb{B}^*$ is secured, then clearly $\mathbf{Dr}(s, b, \perp^\omega)$. Since this a Harrop formula, it follows $!\mathbf{Dr}(s, b, \perp^\omega)$, by $!\mathbf{H}$. With $!\wedge$ and $!\exists$ it follows $!Q_b(s)$.

(3): Let $s \in \mathbb{B}^*$ such that $\forall a \in \mathbb{B} \, Q_b(s * a)$, that is, we have $\alpha_0, \alpha_1 \in \mathbb{B}^1_\perp$ such that $\mathbf{Dr}(s * 0, b, \alpha_0)$ and $\mathbf{Dr}(s * 1, b, \alpha_1)$. We have to find $\alpha \in \mathbb{B}^1_\perp$ such that $\mathbf{Dr}(s, b, \alpha)$. Since $F \in \mathbb{B}^2_\perp$, we have $F(s * 0 * \alpha_0) \in \mathbb{B}$. If $F(s * 0 * \alpha_0) = b$, set $\alpha = 0 * \alpha_0$. Otherwise, set $\alpha = 1 * \alpha_1$. This completes the proof of the Claim.

To complete the proof of the theorem, let $\alpha_0, \alpha_1 \in \mathbb{B}^1_\perp$ with $\mathbf{Dr}(s, 0, \alpha_0)$ and $\mathbf{Dr}(s, 1, \alpha_1)$, according to the Claim. Let $a = F \, \alpha_0 \in \mathbb{B}$ and $b = F \, \alpha_1 \in \mathbb{B}$. Clearly, $\mathbf{Const}(F)$ iff $a = b$.

The proof of uniform continuity

Theorem

Every functional $F \in \mathbb{B}^2_\perp$ which is total and continuous is uniformly continuous.

Proof.

Let $F \in \mathbb{B}^2_\perp$ be total and continuous. We set

$$\begin{aligned} \mathbf{UCont}(s,n) &\stackrel{\mathrm{Def}}{=} \forall \alpha, \beta \in \mathbb{B}^1 (\alpha =_n \beta \to F(s * \alpha) = F(s * \beta)) \\ \mathbf{UCont}(s) &\stackrel{\mathrm{Def}}{=} \exists n \in \mathbf{N}\, \mathbf{UCont}(s,n) \end{aligned}$$

and show $\forall s \in \mathbb{B}^* \, \mathbf{UCont}(s)$ by abstract bar induction, **ABI**, on $\prec_{\mathbf{Const}}$ where $\prec$ is as in the proof of the Constancy Theorem and $\mathbf{Const}(s) \stackrel{\mathrm{Def}}{=} \mathbf{Const}(F, s)$.

Applying **ABI**

We have to show:

(1) $\mathbf{Wf}_{\prec_{\mathbf{Const}}}(\langle\rangle)$,

(2) $\forall s \in \mathbb{B}^* \, (\neg\mathbf{Const}(s) \vee \mathbf{UCont}(s))$,

(3) $\forall s \in \mathbb{B}^* \, (\forall a \in \mathbb{B} \, \mathbf{UCont}(s * a) \to \mathbf{UCont}(s))$.

(1) holds again by continuity.

(2): By the Constancy Theorem, we may assume $\mathbf{Const}(s)$. Then clearly $\mathbf{UCont}(s, 0)$.

(3): Assume $\mathbf{UCont}(s * 0, n)$ and $\mathbf{UCont}(s * 1, m)$. Then, clearly, $\mathbf{UCont}(s, 1 + \max(n, m))$.

Program extraction

Declarations:

```
type N = Int
type B = Int
type B1 = N -> B
type B2 = B1 -> B

(***) :: [B] -> B1 -> B1
s *** alpha = \n-> if n < length s
                   then s !! n
                   else alpha (n - length s)
```

Testing constancy

Testing whether a type 2 functional is constant on extensions of s:

```
thm1 :: B2 -> [B] -> Bool
thm1 f s = f (s *** (claim 0 s)) == f (s *** (claim 1 s))

  where

--  Computing the drinker
--  claim :: B -> [B] -> B1
    claim b s = let {
                      alpha0 = claim b (s++[0]) ;
                      alpha1 = claim b (s++[1])
                     }
                  in if f ((s++[0]) *** alpha0) == b
                     then [0] *** alpha0
                     else [1] *** alpha1
```

Computing the mod. of unif. cont. (fan functional)

```
thm2 :: B2 -> N
thm2 f = aux []

  where

--  aux :: [B] -> N
    aux s = if thm1 f s
            then 0
            else 1 + max (aux (s++[0])) (aux (s++[1]))
```

In-class test

```
*Main> thm2 (\alpha-> 0)
0
*Main> thm2 (\alpha-> 1)
0
*Main> thm2 (\alpha-> alpha 1)
2
*Main> thm2 (\alpha-> alpha (sum [alpha n | n <- [0..5]]) )
7
*Main> thm2 (\alpha-> alpha (sum [2 * alpha n | n <- [0..7]]) )
17
*Main> thm2 (\alpha-> alpha (sum [alpha (2*n) | n <- [0..7]]) )
15
```

Conclusion

- The fine grained control of computational content not only optimizes extracted programs but also provides access to new kinds of algorithms by program extraction.
- Limited use of classical logic seems to be required to verify the correctness of these new algorithms.
- The Harrop version of Brouwer's thesis and banged bar induction might open ways to extract programs such as the Berard-Bezem-Coquand realizer of dependent choice from a proof.

References

L. E. J. Brouwer, Beweis dass jede volle Funktion gleichmässig stetig ist.
Nederlandse Akademie van Wetenschappen Verslagen 27, 189193, 1924.

L. E. J. Brouwer, Über Definitionsbereiche von Funktionen,
Math. Annalen 97, 6075, 1927*.*

B. Totale Objekte und Mengen in der Bereichstheorie,
PhD thesis, LMU Munich, *1990.*

V. Veldman. Brouwer's Real Thesis on Bars.
Philosophia Scientiæ, CS 6, *Constructivism: Mathematics, Logic, Philosophy and Linguistics, 2006.*

H. Schwichtenberg. Minlog.
The Seventeen Provers of the World, Lecture Notes in Artificial Intell., *3600, 151–157, 2006.*
`http://www.mathematik.uni-muenchen.de/~logik/minlog/`

M. Escardó. Exhaustible sets in higher-type computation,
Logical Methods in Comput. Sci. *4 (3), 2008.*

References

B. From coinductive proofs to exact real arithmetic: theory and applications.
Logical Methods in Comput. Sci., *7(1):1–24, 2011.*

M. Escardo, P. Oliva. Bar recursion and products of selection functions, *JSL, 80(1):1-28, 2015.*

B, O. Petrovska Optimized program extraction for induction and coinduction
CiE 2018, LNCS 10936, 70-80, 2018.

H. Tsuiki. Gray code and extractions of concurrent executions of partial programs.
Talk, today, 15:00-15:30.

T. Takayama. Program Extraction in Intuitionistic Fixed Point Logic.
Talk, today, 15:30-16:00.

Thank you