A Zoo of Continuity Properties in Constructive Type Theory

Martin Baillon
Nantes Université, École Centrale Nantes, CNRS, Inria, LS2N, UMR 6004, F-44000 Nantes, France

Yannick Forster
Inria, Paris, France

Assia Mahboubi
Nantes Université, École Centrale Nantes, CNRS, Inria, LS2N, UMR 6004, F-44000 Nantes, France
Vrije Universiteit Amsterdam, The Netherlands

Pierre-Marie Pédrot
Nantes Université, École Centrale Nantes, CNRS, Inria, LS2N, UMR 6004, F-44000 Nantes, France

Matthieu Piquerez
Institut für Mathematik, Goethe-Universität Frankfurt, Germany

Abstract

Continuity principles stating that all functions are continuous play a central role in some schools of constructive mathematics. However, there are different ways to formalise the property of being continuous in constructive foundations. We analyse these continuity properties from the perspective of constructive reverse mathematics. We work in constructive type theory, which can be seen as a minimal foundation for constructive reverse mathematics. We treat continuity of functions $F : (Q \to A) \to R$, i.e. with question type Q, answer type A, and result type R. Concretely, we discuss continuity defined via moduli, making the relevant list $L : \mathbb{L}Q$ of questions explicit, dialogue trees, making the question-answer process explicit as inductive tree, and tree functions, making the question-answer process explicit as function. We prove equivalences where possible and isolate necessary and sufficient axioms for equivalence proofs. Many of the results we discuss are already present in the works of Hancock, Pattinson, Ghani, Kawai, Fujiwara, Brede, Herbelin, Escardó, and others. Our main contribution is their formulation over a uniform foundation, the observation that no choice axioms are necessary, the generalisation to arbitrary types from natural numbers where possible, and a mechanisation in the Coq/Rocq proof assistant.

2012 ACM Subject Classification Theory of computation → Constructive mathematics

Keywords and phrases type theory, constructive mathematics, continuity, Coq

Digital Object Identifier 10.4230/LIPIcs.FSCD.2025.9

Related Version *Full Version*: `https://inria.hal.science/hal-04908282v3/file/continuity-zoo-fscd25.pdf`

Supplementary Material *Software (Source Code)*: `https://github.com/amahboubi/continuity-zoo/tree/fscd25`, archived at `swh:1:dir:c67f6a86a5e8f4a170574a93755283429afca449`

Funding This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No.101001995).

Acknowledgements The idea to investigate connections between different continuity properties was born during fruitful conversations with Yann Leray, Meven Lennon Bertrand and Loïc Pujet. The project then greatly benefited from discussions with Andrej Bauer, Bruno da Rocha Paiva, Martín Escardó, Dominik Kirst, Henri Lombardi and Vincent Rahli. Finally, we want to thank Olivier Bournez and all the attendees from the *Continuity, Computability, Constructivity* (*CCC'2024*) workshop for their interesting remarks and thoughts.

© Martin Baillon, Yannick Forster, Assia Mahboubi, Pierre-Marie Pédrot, and Matthieu Piquerez;
licensed under Creative Commons License CC-BY 4.0

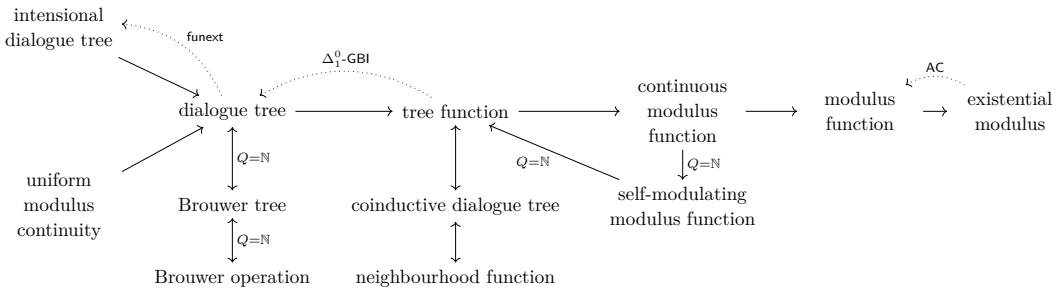

Figure 1 Overview of implications. Dotted arrows indicate that axioms are used.

1 Introduction

One of the most famous theorems of Brouwer's intuitionism is that all functions on the (Dedekind) reals, i.e. of type $\mathbb{R} \to \mathbb{R}$, are continuous [6, 7]. A similar result is due to Markov, proving that functions $\mathbb{R} \to \mathbb{R}$ are sequentially non-discontinuous in the setting of the Russian school of constructive recursive mathematics. A result due to Kreisel and Lacombe and Schoenfield [32], independently also proved by Ceitin [8] generalises this observation to mappings of a complete separable metric space into a separable metric space, and thus in particular to all computable functions $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$.

Continuity properties of functions are a central ingredient in undergraduate mathematics. A function $F : X \to Y$ between topological spaces is continuous if the pre-image $F^{-1}V$ of an open set $V \subseteq Y$ is open, and the $\epsilon\delta$ formulation from undergraduate real analysis unfolds this definition for the standard topology of the real line. The latter is classically equivalent to sequential continuity, which requires that $f(x_n) \to f(x)$ when $n \to \infty$ for any sequence $x_n \to x$. Stronger forms of continuity include uniform continuity and modulus continuity.

The equivalence or implication proofs between this rich zoo of different notions have deep intuitions and are often non-trivial. Some of them rely crucially on classical logic (i.e. the law of excluded middle LEM) or variants of the axiom of choice such as countable choice CC.

In constructive mathematics, the zoo of continuity properties becomes even richer. Troelstra and van Dalen [39] take a modulus-based definition as their central definition in the book on constructive mathematics. It is folklore that the study of the continuity of functions $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ generalises to that of functions $F : (Q \to A) \to R$ for some arbitrary types R, Q and A instead of $\mathbb{N}$. The definition by Troelstra and van Dalen expresses that the value of Ff can only depend on finitely many values of f:

$$\forall f : Q \to A.\ \exists L : \mathbb{L}Q.\ \forall g : Q \to A.(\forall q \in L.\ f\ q = g\ q) \to F\ f = G\ g$$

where $\mathbb{L}Q$ stands for lists of Q. The notion can be derived from the topological notion by instantiating with the product topology on the function space and the discrete one on Q.

As already mentioned, other definitions of continuity are also of interest in constructive mathematics. For instance, one can see functions $(Q \to A) \to R$ as computations with access to an oracle, which explains the suggestive notation Q for questions, A for answers and R for return values. One can then make the question-answer process of a computation explicit in the definition of continuity via a tree where leaves are labelled with return values of type R, internal nodes are labelled by questions of type Q, and branches are labelled by answers of type A. This view ultimately goes back to Kleene [31].

Escardó [18] proposed the first definition of continuity based on an inductive data structure of trees, coined dialogue trees. If $Q = \mathbb{N}$, one can forego labeling internal nodes and consider the depth of a node its implicit label, which was done as an inductive type first by Hancock,

Pattinson, and Ghani [25], resulting in what Escardó and Oliva call Brouwer trees [19]. Weakening the definition, one can also just mandate the existence of a function computing the label given a path in the tree, used for instance by van Oosten [40]. Lastly, one can mandate the existence of a Kleene associate [30] (also called neighbourhood function) or Brouwer operations – notions that turn out to be equivalent to the ones mentioned before, we go into more details in the main text of the paper.

Since, as mentioned, the classical equivalence proofs relying on the law of excluded middle LEM and the axiom of countable choice CC, they do not carry over to constructive mathematics, where one has no access to the law of excluded middle and potentially not even the axiom of countable choice. It is thus a natural question what exactly is needed for an equivalence proof, i.e. whether there is a principle that is equivalent to the equivalence statement. Formally, the question is what well-known independent axiom P validates

$$P \quad \leftrightarrow \quad \forall F : (Q \to A) \to R.\, C_1(F) \to C_2(F)$$

where C_1 and C_2 are different continuity properties in a setting where $C_2(F) \to C_1(F)$ holds constructively. An obvious candidate for the direction from left to right is $\mathsf{LEM} \wedge \mathsf{CC}$, but usually this is too strong an assumption and does not allow proving converse direction.

In general, the field concerned with proving *equivalences* between theorems and assumed axioms is called constructive reverse mathematics [27, 17].

Figure 1 depicts common definitions of continuity and their connection. Several lines of research have yielded insights on how they are connected, in different foundations of constructive mathematics, which we review here.

History and related work. Besides the aforementioned modulus continuity, Troelstra and van Dalen [39] discuss neighbourhood functions, due to Kleene [30] and Brouwer operations. They prove that every Brouwer operation gives rise to a neighbourhood function, but that the statement that every neighbourhood function gives rise to a Brouwer operation is equivalent to the axiom of bar induction for decidable bars (Δ_1^0-BI). Kawai [28] extends this result to more general notions of bar induction. Ishihara [26] discusses other continuity properties in constructive mathematics. He considers the statements "every mapping is sequentially non-discontinuous", "every sequentially non-discontinuous mapping is sequentially continuous", and "every sequentially continuous mapping is continuous" in informal Bishop style constructive mathematics together with the axiom of countable choice. He establishes that the first is equivalent to the negation of the weak principle of limited omniscience for both complete metric and complete separable metric domains, that the second is equivalent to weak Markov's principle for both complete metric and complete separable metric domains, and that the third is equivalent to the principle that every nonempty pseudo-bounded subset of $\mathbb{N}$ is bounded, both for metric and complete metric domains. He concludes by observing that Richman's axiom stating that partial functions are countable suffices to prove such statements in the presence of Markov's principle (MP).

Ghani, Hancock, and Pattinson [25] consider an inductive definition of Kleene's computation trees, called Brouwer trees by Escardó and Oliva [19]. They prove that modulus continuity implies Brouwer tree continuity under the assumption of bar induction (BI). Brede and Herbelin [5] consider generalised bar induction to arbitrary, non-necessarily countable types, resulting in the generalised bar induction principle (GBI). Escardo and Oliva [19] and Sterling [38] give distinct Agda formalisations of the proof that dialogue trees with $Q = \mathbb{N}$ and Brouwer trees are equivalent. Fujiwara and Kawai [22] prove that having a modulus continuous modulus of continuity function is equivalent to a self-modulating modulus of

continuity function over extensional intuitionistic arithmetic E-HA$^\omega$ with countable choice for quantifier-free formulas that are relations between functions $\mathbb{N} \to \mathbb{N}$ and numbers $\mathbb{N}$. Steinberg, Théry, and Thies [37] give a Coq/Rocq formalisation of a related result, assuming the full axiom of choice.

Regarding axioms stating the continuity of all functions, Cohen and Rahli [11, 12] show that a stateful variant TT$^\square$ of constructive type theory effectfully realizes an axiom stating that every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is modulus continuous. In a follow up work with Tosun [9] they furthermore show that a stronger variant stating that every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is dialogue continuous can also be realized. Since TT$^\square$ can be used to construct models of Martin-Löf type theory (MLTT) [10], their results possibly extend to more standard variants of type theory, although we are not aware of any such explicit transposition.

Escardó [18] proves that all functions in System T are dialogue continuous. Kawai [29] generalises this proof strategy to HA$^\omega$, while Baillon, Mahboubi, and Pédrot [1] prove the same for BTT (Baclofen type theory).

Contribution. In this paper, we work towards consolidating the current state of knowledge regarding the connections between different continuity properties for *total* functions in a unified framework for constructive mathematics.

Concretely, we present several properties (see Figure 1) defined for functions $F : (Q \to A) \to R$, whenever possible for arbitrary Q, R, A, and formalised in the Calculus of Inductive Constructions (CIC), the type theory underlying the Coq/Rocq proof assistant. This system can be seen as a minimal foundation with respect to choice axioms, since the axiom of countable choice is not provable, nor is the axiom of unique choice.

We mechanise well-known implication proofs between these properties and carry out both known and novel constructive reverse analysis.

The paper is written in abstract mathematical notation, but lemmas, and theorems are hyperlinked with an HTML version of the Coq/Rocq code available at `https://github.com/amahboubi/continuity-zoo/tree/fscd25`.

Outline. In Section 2 we recall useful statements of principles for the rest of the article. In Sections 3–5 we introduce notions of continuity and prove the implications from Figure 1. In Section 6 we show that Brede and Herbelin's generalised bar induction suffices to prove that coinductive dialogue trees give rise to dialogue trees, and furthermore that Δ^0_1-BI is equivalent to the statement that every coinductive dialogue continuous function is dialogue continuous for $Q = \mathbb{N}$. We furthermore show how a general continuity axiom stating that every function is dialogue continuous implies the law of excluded middle or Markov's principle respectively for certain choices of Q, A, R, or even outright falsity for too general choices. In Section 7 we discuss other related work that is not part of our zoo.

2 Principles in Constructive Mathematics

We work in the Calculus of Inductive Constructions CIC [14, 15, 35], the type theory underlying the Coq/Rocq proof assistant [13]. CIC is a foundational system that provides both an expressive logic and a rich programming language.

To settle the conventions, we make use of the following inductive types: Booleans $\mathbb{B}$ generated by true and false, natural numbers $\mathbb{N}$ generated by 0 and $\mathsf{S} : \mathbb{N} \to \mathbb{N}$, options $\mathbb{O}X$ generated by None and $\mathsf{Some} : X \to \mathbb{O}X$, sums $X + Y$ generated by $\mathsf{inl}\ x$ and $\mathsf{inr}\ y$ for $x : X$ and $y : Y$, and lists $\mathbb{L}X$ generated by $[\,]$ and $x :: l$ for $x : X$, $l : \mathbb{L}X$

These types enjoy the structural large elimination principle systematically derived from their constructors [35]. For a non-empty list $l : \mathbb{L}T$, l_k denotes the k-th element of l, for $0 \leq k < |l|$, where $|l|$ is the length of l.

Contrarily to the original Martin-Löf type theory, CIC segregates proofs from programs by using two distinct sorts. Types of programs are given the sort[1] $\mathbb{T}$, while propositions are given the sort $\mathbb{P}$. Unlike programs, proofs of propositions are intended to be proof-irrelevant. The theory is designed so that there is no way for a program whose type is in $\mathbb{T}$ to depend computationally on a proof of a proposition in $\mathbb{P}$. This is reflected by the standard erasure model of CIC [34] which strips terms from their propositional subcomponents, effectively keeping only the computational skeleton of the proof. This stratification allows extending CIC with classical axioms, as long as they live in $\mathbb{P}$, without disrupting the computational interpretation of the theory.

Despite this, we work in an axiom-free $\mathbb{P}$ fragment and thus in a purely intuitionistic logic. This means in particular that a propositional case analysis $P \vee \neg P$ can only be used if one has previously proven this is an intuitionistically valid disjunction.

The axiom that $\forall P : \mathbb{P}.\ P \vee \neg P$ is called the law of excluded middle (LEM). It is equivalent to the axiom of double negation elimination, i.e. $\forall P : \mathbb{P}.\ \neg\neg P \to P$. Restricting double negation elimination to Σ_1^0 formulas yields Markov's principle (MP). When Σ_1^0 predicates are defined in terms of type theoretic functions MP reads as follows:

$$\forall f : \mathbb{N} \to \mathbb{B}.\ \neg\neg(\exists n.\ f n = \mathsf{true}) \to \exists n.\ f n = \mathsf{true}.$$

Given a predicate $P : X \to \mathbb{P}$ over some type $X : \mathbb{T}$, one can form the existential quantification $(\exists x : X.\ P\ x) : \mathbb{P}$, which is a proposition. When we use the word *exists* in natural language, except if mentioned otherwise, we always refer to this notion of existence rather than the more informative Σ-type of computationally relevant dependent pairs.

Due to the computational segregation of CIC, one cannot define a projection function $(\exists x : X.\ P\ x) \to X$. More generally, most forms of the axiom of choice in $\mathbb{P}$ are not provable. Notably, countable choice is *not* provable, even when the codomain of the relation is just $\mathbb{N}$:

$$\mathsf{AC}(\mathbb{N},\mathbb{N}) := \forall R : \mathbb{N} \to \mathbb{N} \to \mathbb{P}.\ (\forall x : \mathbb{N}.\ \exists y : \mathbb{N}.\ R\ x\ y) \to \exists f : \mathbb{N} \to \mathbb{N}.\ \forall x.\ R\ x\ (f\ x)$$

The only exception where case analysis over a proposition to form a term in a proof-relevant type is allowed is called the *singleton* criterion. There are basically three cases where such a case analysis over a proposition $T : \mathbb{P}$ is allowed.

- T is an empty type, which corresponds to dead branches in the program.
- T is an equality, which corresponds to transparent type casts at runtime.
- T is a well-founded accessibility predicate, which corresponds to an *a priori* unbounded fixpoint whose termination is ensured by a metatheoretical argument.

The latter allows to prove a weak form of choice principle for computationally decidable relations over $\mathbb{N}$. We call this principle Δ_1^0-$\mathsf{AC}(X,\mathbb{N})$ or Δ_1^0-choice for short:

$$\Delta_1^0\text{-}\mathsf{AC}(X,\mathbb{N}) := \forall g : X \to \mathbb{N} \to \mathbb{B}.\ (\forall x : X.\ \exists y : \mathbb{N}.\ g\ x\ y = \mathsf{true}) \to \exists f.\ \forall x.\ g\ x\ (f\ x) = \mathsf{true}$$

We make use of this technique in this paper twice to be able to eliminate from proofs.

Note that there is more than one sensible definition of Δ_1^0 in constructive systems, ours is one of them. Δ_1^0-choice is assumed in virtually all constructive foundations, and it is provable in all foundations based on constructive type theory. We explained why it holds in

[1] Actually a denumerable hierarchy thereof, but we do not rely on it in this paper.

$$F\ f := \begin{cases} f\ 1 & \text{if } f\ 0 = 0 \\ f\ 2 & \text{otherwise.} \end{cases} \qquad \begin{aligned} &\beta\ 0\ (\lambda a_0.\ \textbf{if}\ a_0 = 0\ \textbf{then}\ \beta\ 1\ (\lambda a_1.\ \eta\ a_1) \\ &\qquad\qquad \textbf{else}\ \beta\ 2\ (\lambda a_2.\ \eta\ a_2)) \end{aligned}$$

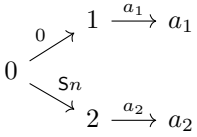

Figure 2 Running example.

CIC. In MLTT-like systems without a universe of propositions, the principle holds trivially even without restricting to Δ_1^0 because the $\forall x.\exists y$ proof can be used to construct the function by projection since $\exists$ is modeled as Σ. In univalent systems with a semantic notion of propositions it holds because unique choice holds, and $\forall x.\exists y$ can be rephrased as $\forall x.\exists! y$, such that y is the least y with $g\ x\ y = \mathsf{true}$. We refer to, e.g. the paper by Cohen et al. [10, Section 2] for a longer discussion.

3 Inductive trees

In this section, we consider various inductive definitions of trees, namely dialogue trees (with labelled leaves, nodes, and branches), Brouwer trees (without labelled nodes), and intensional dialogue trees (a predicate on functions), and define notions of continuity based on them. We explain the proofs that intensional dialogue tree continuity implies dialogue tree continuity, which in turn is equivalent to Brouwer tree continuity for $Q = \mathbb{N}$.

3.1 Dialogue trees

We define an inductive type of *dialogue trees* $\mathbb{D}$ with leaves η labelled by a result R and internal nodes β labelled by a question $q : Q$ and branching via answers in type A:

$$\frac{r : R}{\eta\ r : \mathbb{D}} \qquad\qquad \frac{q : Q \qquad k : A \to \mathbb{D}}{\beta\ q\ k : \mathbb{D}}$$

We define an evaluation function $\partial : (Q \to A) \to \mathbb{D} \to R$ as follows.

$$\partial\ f\ (\eta\ r) := r \qquad\qquad \partial\ f\ (\beta\ q\ k) := \partial\ f\ (k\ (f\ q))$$

Definition 1. *A function $F : (Q \to A) \to R$ is* dialogue continuous via $d : \mathbb{D}$ *if*

$$\forall f : Q \to A.\ F\ f = \partial\ f\ d$$

and we say that F is dialogue continuous *if such a d exists.*

The definition goes back to Ghani, Hancock, and Pattinson [24, 25]. Dialogue continuous functions are called *eloquent* by Escardó [18].

As a running example for the rest of the paper we consider $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ as defined in Figure 2. It is easy to see that F is dialogue continuous via the dialogue tree given in the figure, depicted also as a diagram on the right.

Note, however, that a single function may be represented by several dialogue trees: For instance, dialogue trees can include irrelevant questions (left) or ask the same questions multiple times (right):

$$\begin{aligned} &\beta\ 4\ (\lambda a_4.\ \beta\ 0\ (\lambda a_0. \\ &\quad \textbf{if}\ a_0 = 0\ \textbf{then}\ \beta\ 1\ (\lambda a_1.\ \eta\ a_1)\ \textbf{else}\ \beta\ 2\ (\lambda a_2.\ \eta\ a_2))) \end{aligned} \qquad\qquad \beta\ 0\ (\lambda a_0.\ \beta\ 0\ (\lambda a_0'.\ a_0)).$$

3.2 Brouwer trees

Dialogue trees with $Q = \mathbb{N}$ can ask questions in arbitrary order. By contrast, Brouwer trees $\mathbb{D}_{\mathsf{B}}$ do not have labels on their internal nodes, and thus always ask question n at depth n.

$$\frac{r : R}{\eta_{\mathsf{B}}\, r : \mathbb{D}_{\mathsf{B}}} \qquad \frac{k : A \to \mathbb{D}_{\mathsf{B}}}{\beta_{\mathsf{B}}\, k : \mathbb{D}_{\mathsf{B}}}$$

We define an evaluation function $\partial_{\mathsf{B}} : (\mathbb{N} \to A) \to \mathbb{D}_{\mathsf{B}} \to R$:

$$\partial_{\mathsf{B}}\, f\, (\eta_{\mathsf{B}}\, r) := r \qquad \partial_{\mathsf{B}}\, f\, (\beta_{\mathsf{B}}\, k) := \partial_{\mathsf{B}}\, (\lambda n.\, f\, (\mathsf{S}\, n))\, (k\, (f\, 0))$$

Definition 2. *A function $F : (\mathbb{N} \to A) \to R$ is* Brouwer continuous via $d : \mathbb{D}_{\mathsf{B}}$ *if*

$$\forall f : \mathbb{N} \to A.\ F\, f = \partial_{\mathsf{B}}\, f\, d$$

and we say that F is Brouwer continuous *if such a d exists.*

Lemma 3. *A function F of appropriate type is Brouwer continuous iff it is dialogue continuous.*

We give the full proof in the appendix, where the forward direction is easy and the backwards direction follows the proof by Escardó and Oliva [19].

3.3 Intensional dialogue trees

We define the following more intensional variant of dialogue continuity, enforcing the syntactic shape of the function. We define an (indexed) inductive type $\mathbb{D}'_{\mathsf{i}} : ((Q \to A) \to R) \to \mathbb{T}$ as

$$\frac{r : R}{\eta_{\mathsf{i}}\, r : \mathbb{D}'_{\mathsf{i}}\, (\lambda f.\, r)} \qquad \frac{q : Q \qquad k : A \to (Q \to A) \to R \qquad H : \forall a : A.\ \mathbb{D}'_{\mathsf{i}}\, (k\, a)}{\beta_{\mathsf{i}}\, q\, k\, H : \mathbb{D}'_{\mathsf{i}}\, (\lambda f.\, k\, (f\, q)\, f)}$$

Definition 4. *We use $\mathbb{D}'_i$ to define a predicate $\mathbb{D}_i : ((Q \to A) \to R) \to \mathbb{T}$ as the inhabitation of this type, i.e. $\mathbb{D}_i F := \exists d : \mathbb{D}'_i F.\ \top$. We say that F is* intensionally dialogue continuous.

Lemma 5. *If $\mathbb{D}'_i\, F$, then one can construct $d : \mathbb{D}$ such that F is dialogue continuous via d. Consequently, intensional dialogue continuity implies dialogue continuity.*

Proof. We define recursively the function $\delta : \forall (F : (Q \to A) \to R).\ \mathbb{D}'_{\mathsf{i}}\, F \to \mathbb{D}$ as

$$\delta\, F\, (\eta_{\mathsf{i}}\, r) := \eta\, r \qquad \delta\, F\, (\beta_{\mathsf{i}}\, q\, k\, H) := \beta\, q\, (\lambda a.\, \delta\, (k\, a)\, (H\, a))$$

Then $\forall F\, d\, f.\, \partial\, f\, (\delta\, F\, d) = F\, f$ follows by induction on d. ◀

Lemma 6. *Assuming functional extensionality, the converse implication holds.*

It seems obvious that functional extensionality is necessary here, we do not go into details.

4 Tree functions

In this section, we consider various non-inductive definitions of trees, namely tree functions (with labelled leaves, nodes, and branches), coinductive dialogue trees, neighbourhood functions (essentially tree functions without node labels), and Brouwer operations (inductive predicates on neighbourhood functions), and define notions of continuity based on them.

We explain the proofs that tree function continuity follows from dialogue tree continuity and is equivalent to coinductive dialogue tree continuity as well as neighbourhood function continuity. Lastly, we show that Brouwer operation continuity is equivalent to Brouwer tree continuity, making use of an elimination technique for inductive predicates that becomes crucial again in Section 6.

4.1 Trees as functions

A standard modeling of trees in areas of mathematics and theoretical computer science that traditionally avoid inductive definitions is to define a (possibly infinite) tree with internal labels Q, branch labels A, and leaf labels R as a function $\tau : \mathbb{L}A \to (Q + R)$. Usually, one requires that whenever $\tau\, l = \mathsf{inr}\, r$, then for all l', $\tau\, (l +\!\!+ l') = \mathsf{inr}\, r$, but we can omit such a condition for our continuity definition because we stop evaluating once a result is reached.

We write $\mathsf{ask}\ q$ for $\mathsf{inl}\ q$ and $\mathsf{ret}\ r$ for $\mathsf{inr}\ r$. We define once again an evaluation operation $\partial_{\mathcal{T}} : (Q \to A) \to (\mathbb{L}A \to (Q + R)) \to \mathbb{N} \to Q + R$ evaluating a tree τ until a depth as follows:

$$\partial_{\mathcal{T}}\ f\ \tau\ (\mathsf{S}\ n) := \partial_{\mathcal{T}}\ f\ (\lambda l.\ \tau\ (l +\!\!+ [f\ q]))\ n \quad \text{if } \tau\ [\,] = \mathsf{ask}\ q \qquad \partial_{\mathcal{T}}\ f\ \tau\ n := \tau\ [\,] \quad \text{otherwise}$$

This function is computed by recursion on the depth. Note however that the actual evaluation never depends on the depth.

Lemma 7. *If* $\partial_{\mathcal{T}}\ f\ \tau\ n_1 = \mathsf{ret}\ r_1$ *and* $\partial_{\mathcal{T}}\ f\ \tau\ n_2 = \mathsf{ret}\ r_2$ *then* $r_1 = r_2$.

We call a tree τ *well-founded* if for any $f : Q \to A$ there exist n and r s.t. $\partial_{\mathcal{T}}\ f\ \tau\ n = \mathsf{ret}\ r$. Thus, by Δ^0_1-choice every well-founded tree τ gives rise to a total function $\hat{\tau} : (Q \to A) \to R$.

Definition 8. *F is called* tree function continuous *if there exists a tree $\tau : \mathbb{L}A \to Q + R$ s.t.*

$$\forall f.\ \exists n.\ \partial_{\mathcal{T}}\ f\ \tau\ n = \mathsf{ret}\ (F\ f).$$

Regarding $F : (\mathbb{N} \to A) \to A$ from Figure 2, it has the following underlying tree function:

$$\tau\ [] := \mathsf{ask}\ 0 \quad \tau\ [0] := \mathsf{ask}\ 1 \quad \tau\ [\mathsf{S}\ n] := \mathsf{ask}\ 2 \quad \tau\ (n :: a :: l) := \mathsf{ret}\ a$$

It is straightforward to turn inductive dialogue trees into tree functions.

Lemma 9. *If F is dialogue continuous, F is tree function continuous.*

Proof. We recursively define a function $\mathcal{T} : \mathbb{D} \to \mathbb{L}A \to Q + R$ as

$$\mathcal{T}\ (\eta\ o)\ l := \mathsf{ret}\ o \qquad \mathcal{T}\ (\beta\ q\ k)\ [\,] := \mathsf{ask}\ q \qquad \mathcal{T}\ (\beta\ q\ k)\ (a :: l) := \mathcal{T}\ (k\ a)\ l$$

By induction over $d : \mathbb{D}$ it is easy to check that $\mathcal{T}\ d$ is always well-founded and that in particular, the property $\exists n : \mathbb{N}.\ \partial\ f\ d = \partial_{\mathcal{T}}\ f\ (\mathcal{T}\ d)\ n$ holds for any $d : \mathbb{D}$. ◀

Section 6 discusses that the converse is unprovable and equivalent to bar induction.

4.2 Coinductive dialogue trees

We define the following coinductive analogue to dialogue trees, using double lines to emphasize the coinductive definition of the type.

$$\frac{r : R}{\eta_\infty\, o : \mathbb{D}_\infty} \qquad \frac{q : Q \qquad k : A \to \mathbb{D}_\infty}{\beta_\infty\, q\, k : \mathbb{D}_\infty}$$

Note that coinductive dialogue trees are *interaction trees* without silent transitions [41].

We define an evaluation function $\partial_\infty : (Q \to A) \to \mathbb{D}_\infty \to \mathbb{N} \to Q + R$ on coinductive dialogue trees, similar to that on dialogue trees. As for tree functions, we define it by recursion on a natural number fuel, returning either a question to ask or an output.

$$\partial_\infty\, f\, (\eta_\infty\, r)\, n := \mathsf{inr}\, r \qquad \partial_\infty\, f\, (\beta_\infty\, q\, k)\, 0 := \mathsf{inl}\, q \qquad \partial_\infty\, f\, (\beta_\infty\, q\, k)\, (\mathsf{S}\, n) := \partial_\infty\, f\, (k\, (f\, q))\, n$$

Definition 10. *A function $F : (Q \to A) \to R$ is* coinductively dialogue continuous via $d : \mathbb{D}_\infty$ *whenever*

$$\forall f : Q \to A. \exists n.\ \mathsf{inr}\, F\, f = \partial_\infty\, f\, d\, n$$

and we say that F is coinductively dialogue continuous *if such a d exists.*

Lemma 11. *A function is tree function continuous if and only if it is coinductively dialogue continuous.*

Proof. To convert a tree function to a coinductive dialogue tree, we define a co-fixpoint.

$$C'\ \tau\ l := \begin{cases} \beta_\infty\ q\ (\lambda a.\ C'\ \tau\ (l +\!\!+ [a])) & \text{if } \tau\ l = \mathsf{ask}\ q \\ \eta_\infty\ r & \text{if } \tau\ l = \mathsf{ret}\ r \end{cases} \qquad C\ \tau := C'\ \tau\ [\,]$$

Note that this is a valid co-fixpoint because it returns a constructor application in every case. For the converse direction, we define a function by recursion on the answer list:

$$T\ (\eta_\infty\ q\ k)\ l := \mathsf{ret}\ q \qquad T\ (\beta_\infty\ q\ k)\ [\,] := \mathsf{ask}\ q \qquad T\ (\beta_\infty\ q\ k)\ := T\ (k\ a)\ l$$ ◀

4.3 Neighbourhood functions

Kleene [30] defines *neighbourhood functions* (originally called *associates*) for $F : (\mathbb{N} \to A) \to R$ as tree functions associated to Brouwer trees. Formally, they are defined as functions $\nu : \mathbb{L}\ A \to \mathbb{O}\ R$. which are required to be well-founded and monotonic:

$$\forall \alpha : \mathbb{N} \to A.\ \exists r\ n.\ \nu[\alpha\ 0, ..., \alpha\ n] = \mathsf{Some}\, r \quad \forall l\ l'.\ \forall a : A.\ \nu l = \mathsf{Some}\, a \to \nu\ (l +\!\!+ l') = \mathsf{Some}\, a.$$

Actually, similarly to tree functions, the latter does not matter for the definition of continuity.

Definition 12. *A function $F : (\mathbb{N} \to A) \to R$ has an underlying* neighbourhood function $\nu : \mathbb{L}\ A \to \mathbb{O}\ R$ *if*

$$\forall \alpha : \mathbb{N} \to A.\ \exists n : \mathbb{N}.\ \nu\ [\alpha\ 0, \ldots, \alpha\ (n-1)] = \mathsf{Some}\, (f\ \alpha),$$

and is neighbourhood continuous *if such a ν exists.*

Continuing the running example from Figure 2, the function $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ has the following underlying neighbourhood function:

$$\nu\ [0; a_1] := \mathsf{Some}\, a_1 \quad \nu\ [\mathsf{S}\, n; a_1; a_2] := \mathsf{Some}\, a_2 \quad \nu\ l := \mathsf{None}\ \text{otherwise}$$

Lemma 13. *A function $F : (\mathbb{N} \to A) \to R$ has an underlying neighbourhood function if and only if it has an underlying tree function.*

Proof. Turning a neighbourhood function ν into a tree function is trivial by

$$\tau\, l := \begin{cases} \mathsf{ask}\ |l| & \text{if } \nu\, l = \mathsf{None} \\ \mathsf{ret}\ r & \text{if } \nu\, l = \mathsf{Some}\, r \end{cases}$$

The converse direction can be done in three steps, all similar to previous proofs. First, we define a coinductive analogon to Brouwer trees as follows:

$$\frac{r : R}{\eta^B_\infty\, r : {\mathbb{D}_\infty}^B} \qquad\qquad \frac{k : A \to \mathbb{D}^B_\infty}{\beta_\infty\, q\, k : {\mathbb{D}_\infty}^B}$$

Then, with a proof similar to Lemma 11, we show that neighbourhood functions are equivalent to coinductive Brouwer trees. Using Lemma 11, all that is left is to show that coinductive Brouwer trees and coinductive dialogue trees are equivalent. This is done by adapting Escardó and Oliva's proof to the coinductive case.
In Coq/Rocq we work with a direct proof, free of coinductive concepts. ◀

4.4 Brouwer operations

Brouwer operations are defined as inductive predicates on neighbourhood functions. The usual definition is the following: $\mathcal{B} : (\mathbb{L}A \to \mathbb{O}R) \to \mathbb{P}$.

$$\frac{\forall l.\ \nu\, l = \mathsf{Some}\, r}{\mathcal{B}(\nu)} \qquad\qquad \frac{\nu\, [\,] = \mathsf{None} \qquad \forall a.\ \mathcal{B}(\lambda l.\ \nu\, (a :: l))}{\mathcal{B}(\nu)}$$

However, this definition is mostly used in a setting with function extensionality, which is not provable in Coq/Rocq. We use the following definition instead, indexed by a list of answers: $\mathcal{B}' : \mathbb{L}A \to (\mathbb{L}A \to \mathbb{O}R) \to \mathbb{P}$.

$$\frac{\forall l'.\ \nu\, (l +\!\!+ l') = \mathsf{Some}\, r}{\mathcal{B}'_l\, \nu} \qquad\qquad \frac{\nu\, l = \mathsf{None} \qquad \forall a.\ \mathcal{B}'_{l +\!\!+ [a]}\, \nu}{\mathcal{B}'_l\, \nu}$$

Assuming function extensionality, $\mathcal{B}$ is equivalent to $\mathcal{B}'\, [\,]$.

Definition 14. *A function $F : (\mathbb{N} \to A) \to R$ is* Brouwer operation continuous *if there is a Brouwer operation ν which is a neighbourhood for F.*

Troelstra and van Dalen [39] call K the set of Brouwer operation continuous functions.
Note that the F from our running example in Figure 2 is Brouwer operation continuous.

Lemma 15. *Any Brouwer operation continuous function is neighbourhood continuous.*

Proof. Well-foundedness and monotonicity of ν can be proved by induction on $\mathcal{B}(\nu)$. ◀

Lemma 16. *Any Brouwer continuous function is Brouwer operation continuous.*

Proof. Define ν by recursion on the tree, and prove $\mathcal{B}(\nu)$ by induction. ◀

The converse direction is more intricate than it might seem. The reason is that Brouwer operations are defined as inductive *predicates* on neighbourhoods. Since inductive predicates with more than one constructor cannot be directly used to construct elements of a computational inductive type such as Brouwer trees, we first have to show that the Brouwer operation predicate can equivalently be presented as single-constructor inductive type, so as to resort to the singleton elimination rule and its elimination to an informative sort (cf Section 2).

Lemma 17. *Any Brouwer operation continuous function is Brouwer continuous.*

Proof. We define the predicate $\mathcal{B}'' : \mathbb{L}A \to (\mathbb{L}A \to \mathbb{O}R) \to \mathbb{P}$ with one rule:

$$\frac{\nu\, l = \mathsf{None} \to \forall a.\, \mathcal{B}''_{l +\!\!+ [a]}(\nu)}{\mathcal{B}''_l(\nu)}$$

It is indexed by the list of answers already given. We have $\mathcal{B}'_l\, \nu \leftrightarrow \mathcal{B}''_l\, \nu$ by two inductions. Now $\mathcal{B}''$ entails a computational elimination principle for $\mathcal{B}'$, yielding a Brouwer tree. ◀

Note that it is integral in the proof that one can computationally decide whether $\nu l = \mathsf{None}$.

5 Continuity properties based on modulus functions

In this section, we consider modulus functions, uniform modulus continuity, and the existence of moduli, and define notions of continuity based on them.

We explain how uniform continuity implies dialogue tree continuity, and is in fact equivalent for $Q = \mathbb{N}$ and $A = \mathbb{B}$, that tree function continuity implies the existence of a modulus function continuous modulus function, which implies the existence of a self-modulating modulus function for $Q = \mathbb{N}$ which implies tree function continuity for $Q = \mathbb{N}$. Furthermore, modulus function continuity implies existential modulus continuity, while the reverse direction can be proved with the axiom of countable choice.

5.1 Modulus functions

We say that $F : (Q \to A) \to R$ has modulus $L : \mathbb{L}Q$ at $f : Q \to A$ if F is determined solely by the questions to f in L, i.e. for any g such that $\forall x \in L.\, f\, x = g\, x$ we have $F\, f = F\, g$.

Definition 18. *A function $F : (Q \to A) \to R$ is* existentially modulus continuous *if*

$$\forall f : Q \to A.\, \exists L : \mathbb{L}Q.\, \forall g : Q \to A.\, (\forall x \in L.\, f\, x = g\, x) \to F\, f = F\, g.$$

A function $F : (Q \to A) \to R$ has a modulus of continuity function $M : (Q \to A) \to \mathbb{L}Q$ *if for all f, $M\, f$ is a modulus of continuity for F at f. Formally:*

$$\forall f g : Q \to A.\, (\forall x \in M f.\, f\, x = g\, x) \to F\, f = F\, g$$

If a modulus of continuity function exists, we say that F is modulus function continuous.

Continuing the running example from Figure 2, the function $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ can be given several modulus of continuity functions. For instance the constant one $M\ f := [0, 1, 2]$ or $M\ f := [0, \textbf{if}\ f\ 0 = 0\ \textbf{then}\ 1\ \textbf{else}\ 2]$ returning a shorter list.

Lemma 19. *If F is tree function continuous, it is modulus function continuous.*

Proof. By assumption we have a tree function τ such that $\forall f.\, \exists n.\, \partial_\mathcal{T}\, \tau\, f\, n = \mathsf{ret}\ (F\, f)$. Using Δ^0_1-choice, we get a function $N : (Q \to A) \to \mathbb{N}$ s.t. $\forall f.\, \partial_\mathcal{T}\, \tau\, f\, (N\, f) = \mathsf{ret}\ (F\, f)$.

We define a trace function $\mathfrak{t} : (Q \to A) \to (\mathbb{L}A \to (Q + R)) \to \mathbb{N} \to \mathbb{L}Q$ similar to $\partial_\mathcal{T}$ as $\mathfrak{t}\, f\, \tau\, (\mathsf{S}\, n) := q :: \mathfrak{t}\, f\, (\lambda l.\, \tau\, (l +\!\!+ [f\, q]))\, n$ if $\tau\, [\,] = \mathsf{ask}\ q$ and $\mathfrak{t}\, f\, \tau\, n := [\,]$ otherwise

Now define $M\, f := \mathfrak{t}\, f\, \tau\, (N\, f)$. ◀

Lemma 20. *If F is modulus function continuous, it is existentially modulus function continuous, and the converse holds when assuming the axiom of choice on the relevant types, i.e. $\mathsf{AC}(Q \to A, \mathbb{L}Q)$.*

5.2 Uniform continuity

Definition 21. *A function $F : (Q \to A) \to R$ is* uniformly continuous *if the list of questions L does not depend on the input, i.e. if*

$$\exists L : \mathbb{L}Q.\ \forall f g : Q \to A.\ (\forall x \in L.\ f\ x = g\ x) \to F\ f = F\ g.$$

Lemma 22. *If Q is finite, i.e. $\exists l : \mathbb{L}Q.\forall q.\ q \in l$, every $F : (Q \to A) \to R$ that is existentially modulus continuous is uniformly continuous.*

Proof. Take the list $L : \mathbb{L}Q$ containing all elements of Q. ◀

Lemma 23. *If A is inhabited, Q has decidable equality and F is uniformly continuous then it is dialogue continuous.*

Proof. We define a dialogue tree which one after the other asks all questions in the list L and then calls F on the function returning the given answers to these questions and a default value otherwise. Formally, take F and $a_0 : A$ and define $d : \mathbb{L}Q \to \mathbb{L}(Q \times A) \to \mathbb{D}$ by recursion over its first argument as

$$d\ [\,]\ \ell := \eta\ (F\ (\varrho\ \ell)) \qquad\qquad d\ (q :: L)\ \ell = \beta\ q\ (\lambda a.\ d\ l\ ((q, a) :: \ell))$$

where $\varrho : \mathbb{L}(Q \times A) \to Q \to A$ associates to $q : Q$ its first value in the list and returns a_0 otherwise, relying on decidable equality of Q. Now F is dialogue continuous via $d\ L\ [\,]$. ◀

Uniform continuity and dialogue continuity are not equivalent in general.

Lemma 24. *When A is finite, dialogue continuity implies uniform continuity.*

Proof. We just need to prove the direction from right to left. Let $as : \mathbb{L}A$ contain all elements of A. Define $L\ (\eta\ r) := [\,]$ and $L\ (\beta\ q\ k) := q :: [q' \mid q' \in L\ (k\ a), a \in as]$.

The claim then follows by induction on the dialogue tree. ◀

Note that the running example is uniformly continuous with modulus $[0; 1; 2]$.

Lemma 25. *The function $F(f : \mathbb{N} \to \mathbb{N}) := f\ (f\ 0)$ is dialogue continuous but not uniformly continuous.*

Proof. It is easy to check that it is dialogue continuous via $\beta\ 0\ (\lambda a.\ \beta\ a\ (\lambda r.\ \eta\ r))$.

Assume there is a list $L : \mathbb{L}\mathbb{N}$ which is a modulus of continuity for F at any f. Let n be a number not in L and greater than 0. Define

$$f\ q := \begin{cases} 0 & \text{if } q = n \\ n & \text{otherwise} \end{cases} \qquad\qquad g\ q := \begin{cases} 1 & \text{if } q = n \\ n & \text{otherwise} \end{cases}$$

Now $F\ f = f\ (f\ 0) = f\ n = 0 \neq 1 = g\ n = g\ (g\ 0) = F\ g$. But because n is not in L and f and g agree on all other elements they in particular agree on L and thus $F\ f = F\ g$ since L is a modulus of continuity for both. ◀

Note that this implies that the axiom stating that every function $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is uniformly continuous is inconsistent.

5.3 Self modulation

Fujiwara and Kawai [22] prove that tree function continuity is equivalent to having a modulus of continuity function which is also a modulus function for itself. The backwards direction of the equivalence crucially requires that $Q = \mathbb{N}$, and on Δ_1^0-$\mathsf{AC}_{X,\mathbb{N}}$ for any type X.

Lemma 26. *Tree function continuous F have a self-modulating modulus of continuity function.*

Proof. Let τ be a tree function for F. It turns out that the modulus function $M f := \mathfrak{t} f \tau (N f)$ that we defined in the proof of Lemma 19 is a modulus of continuity for F and for itself. ◀

We give the full proof that for the special case when $Q = \mathbb{N}$ one can turn any continuous modulus of continuity function M for a given F into a self-modulating modulus of continuity function N for F, following Fujiwara and Kawai, in the appendix.

Lemma 27. *If $Q = \mathbb{N}$, and F has a continuous modulus of continuity function, then F has a self-modulating modulus of continuity function.*

Lemma 28. *If $Q = \mathbb{N}$, and F has a self-modulating modulus of continuity function, then F is tree function continuous.*

Proof. Let M be a modulus of continuity function for F. In order to construct a tree function for f, we reverse the central construction to the proof of Lemma 27 and introduce for $l : \mathbb{L}A$, $a : A$, and $m : \mathbb{N}$ the function $\restriction_a l\ m := \textbf{if}\ m \leq |l|\ \textbf{then}\ l_m\ \textbf{else}\ a$.

Note that for any l and any a, when the largest natural number $\max\,(M\,(\restriction_a l))$ of this modulus is strictly smaller than the size $|l|$ of its argument, then $F\ \restriction_a l = F\ f$ as soon as $f\ k = l_k$ for $k < |l|$. This observation is used in the definition of the desired tree function to decide when the list argument is long enough for this tree function to return a value. We thus define the transformation $T : ((\mathbb{N} \to A) \to \mathbb{L}\mathbb{N}) \to \mathbb{L}A \to (\mathbb{N} + R)$ as follows:

$$T\ M\ [\,] := \mathsf{ask}\ 0 \quad T\ M\ (a :: l) := \textbf{let}\ g := \restriction_a (a :: l)\ \textbf{in}\ \textbf{if}\ \max\,(M\ g) < |a :: l|\ \textbf{then}\ \mathsf{ret}\ (F\ g)\ \textbf{else}\ \mathsf{ask}\ |a :: l|.$$

Now for $f : \mathbb{N} \to A$, set $m_f := \max\,(M\ f)$. We have that $M\ f = M\ \restriction_{f\,0} [f\,0; \ldots; f\,m_f]$ when M is self-modulating. Then $T\,M$ is a tree function for F when M is self-modulating modulus continuity function because the evaluation of $T\,M$ on f at step index $m_f + 1$ returns $F\,f$. ◀

Note that generalising beyond $Q = \mathbb{N}$ seems infeasible. One route to go would be to generalise to any retract of $\mathbb{N}$, but in practice this only extends the result to finite types, where any function with finite question type is already uniformly continuous.

Other generalisations, at least of the present proof, seem to be impossible. First, turning a list of questions and answers into a function requires decidable equality on Q. Second, the required choice function has to return minimal elements, meaning we would either have to require a size function into natural numbers and a minimizing choice principle w.r.t. this size, or a decidable, antisymmetric, linear order on Q with a minimizing choice principle. Both properties seem rather strong and are possibly equivalent to Q being a retract of $\mathbb{N}$ again.

6 Reverse analysis: (Generalised) Bar Induction

In this section, we discuss the impossibility to prove that coinductive dialogue continuous functions (or equivalently tree continuous functions) $F : (Q \to A) \to R$ are dialogue

continuous. Concretely, we prove that if one takes this statement as a principle, it implies the principle of bar induction, which is independent in CIC. Analogous to bar induction, we call this principle $\mathsf{CI}(Q, A, R)$, for continuity induction, since it allows turning a coinductive tree into an inductive tree. It can be seen as a generalisation of Fujiwara and Kawai's $\mathsf{BC_c}$ principle [23], which states that any function $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ with a continuous modulus of continuity function is induced by a Brouwer operation. Fujiwara and Kawai prove that BC_c is equivalent to bar induction for decidable bars for $Q = A = R = \mathbb{N}$. Note that variants of bar induction are equivalent to the so-called fan theorem [39].

Brede and Herbelin generalise bar induction to arbitrary Q, A and obtain the principle $\mathsf{GBI}(Q, A)$ [5], which states that branching trees over a type A with internal nodes labelled over Q are inductively barred as soon as they are barred. In their setting, a bar T is a predicate on $\mathbb{L}A$ which eventually holds for a certain prefix of any path from the root, and an inductive bar is a predicate on $\mathbb{L}A$ whose hereditary closure contains the empty list. They prove that $\mathsf{GBI}(\mathbb{N}, A)$ is equivalent to regular bar induction $\mathsf{BI}(A)$, where the nodes of the bars are unlabelled and they still branch via A. We were unable to reconstruct the proof, but show that this equivalence holds for decidable bars, which is sufficient.

We then prove that $\Delta^0_1\text{-}\mathsf{GBI}(Q, A)$ implies $\mathsf{CI}(Q, A, R)$ when Q has decidable equality, and that $\mathsf{CI}(\mathbb{N}, A, \mathbb{L}A)$ implies $\Delta^0_1\text{-}\mathsf{BI}(A)$.

We introduce both bar induction $\mathsf{BI}(A)$ and generalised bar induction $\mathsf{GBI}(Q, A)$. Let P be a predicate on lists $\mathbb{L}A$. P is *barred* if the property on the left holds. Brede and Herbelin generalise this to predicates P' on lists $\mathbb{L}(Q \times A)$ as (Q, A)-*barred*, on the right:

$$\forall f : \mathbb{N} \to A.\ \exists n : \mathbb{N}.\ P\,[f\,0, \ldots, f\,n] \qquad \forall f : Q \to A.\ \exists l : \mathbb{L}Q.\ P'\,[(x, f\,x) \mid x \in l.]$$

We now define the *hereditary closure* $\mathcal{H}\ P\ l$ of $P : \mathbb{L}A \to \mathbb{P}$ at $l : \mathbb{L}A$ and the (Q, A)-*hereditary closure* $\mathcal{H}^{\mathsf{G}}_{Q,A}\ P'\ l$ of $P' : \mathbb{L}(Q \times A) \to \mathbb{P}$ at l:

$$\frac{P\ l}{\mathcal{H}\ P\ l} \qquad \frac{\forall (a : A).\ \mathcal{H}\ P\ (l +\!\!+ [a])}{\mathcal{H}\ P\ l} \qquad \frac{P'\ l \quad l \subseteq l'}{\mathcal{H}^{\mathsf{G}}_{Q,A}\ P'\ l'} \qquad \frac{q \notin l \quad \forall (a : A).\ \mathcal{H}^{\mathsf{G}}_{Q,A}\ P'\ (l +\!\!+ [(q, a)])}{\mathcal{H}^{\mathsf{G}}_{Q,A}\ P'\ l}$$

where $l \subseteq l'$ is inclusion for lists seen as finite sets. Observe that this latter condition makes l an arbitrary permutation of some possibly duplicated elements of l'. A predicate is *inductively barred* if $\mathcal{H}\ P\ [\,]$ holds and (Q, A)-*inductively barred* if $\mathcal{H}^{\mathsf{G}}_{Q,A}\ P'\ [\,]$ holds.

Definition 29. $\mathsf{BI}(A)$ *states that any barred predicate is inductively barred.* $\Delta^0_1\text{-}\mathsf{BI}(A)$ *restricts* $\mathsf{BI}(A)$ *to decidable predicates, i.e. predicates reflected by a function* $f : \mathbb{L}A \to \mathbb{B}$.

Definition 30. $\mathsf{GBI}(Q, A)$ *states that any* (Q, A)*-barred predicate is* (Q, A)*-inductively barred.* $\Delta^0_1\text{-}\mathsf{GBI}(Q, A)$ *restricts* $\mathsf{GBI}(Q, A)$ *to decidable predicates, i.e. predicates reflected by a function* $f : \mathbb{L}(Q \times A) \to \mathbb{B}$.

We give the proof of the following in the appendix, and then turn towards CI.

Lemma 31. $\Delta^0_1\text{-}\mathsf{GBI}(\mathbb{N}, A) \leftrightarrow \Delta^0_1\text{-}\mathsf{BI}(A)$.

Lemma 32. *If* Q *has decidable equality, then* $\Delta^0_1\text{-}\mathsf{GBI}(Q, A) \to \mathsf{CI}(Q, A, R)$.

Proof. Let $F : (Q \to A) \to R$ be coinductively dialogue continuous with witness $d : \mathbb{D}_\infty$. We first define the evaluation of coinductive dialogue trees $\partial^\perp_\infty : \mathbb{D}_\infty \to \mathbb{L}(Q \times A) \to \mathbb{N} \to Q + R$ using finite functions represented as lists:

$$\begin{array}{l} \partial^\perp_\infty\ (\eta_\infty\ r)\ l\ n \quad := \mathsf{ret}\ r \\ \partial^\perp_\infty\ (\beta_\infty\ q\ k)\ l\ 0 := \mathsf{ask}\ q \end{array} \qquad \partial^\perp_\infty\ (\beta_\infty\ q\ k)\ l\ (\mathsf{S}\ n) := \begin{cases} \partial^\perp_\infty\ (k\ a)\ l\ n \text{ if } (q, a) \in l \\ \mathsf{ask}\ q \text{ otherwise} \end{cases}$$

We now define the decidable predicate $T\,(l : \mathbb{L}(Q \times A)) := \exists(r : R).\, \partial_\infty^\perp\, d\, l\, |l| = \mathsf{ret}\ r$. The fact that F is coinductively dialogue continuous means that T is (Q, A)-barred. We can thus apply Δ_1^0-$\mathsf{GBI}(Q, A)$ and get a proof that T is inductively (Q, A)-barred. However, this is not enough to conclude: inductive (Q, A)-barredness is an inductive predicate on T, it is not possible in general to extract a dialogue tree out of it. Fortunately, all properties under scrutiny are decidable, and we can tailor inductive predicates to use singleton elimination. The proof is technical, we refer the interested reader to the mechanization. ◀

Lemma 33. $\mathsf{CI}(\mathbb{N}, A, \mathbb{L}A) \to \Delta_1^0$-$\mathsf{BI}(A)$.

Proof. Let a decidable bar $P : \mathbb{L}A \to \mathbb{P}$ be given. Assume that P is barred, i.e. that $\forall f : (\mathbb{N} \to A).\, \exists n.\, P\,[f\,0, \ldots, f\,n]$. Since P is decidable we can use Δ_1^0-$\mathrm{AC}_{\mathbb{N}\to A,\mathbb{N}}$ and define a function $F : (\mathbb{N} \to A) \to \mathbb{L}A$ as $F\ f := [f\,0, \ldots, f\,n]$ for the smallest $n : \mathbb{N}$ such that $P\,[f\,0, \ldots, f\,n] = \mathsf{true}$. Moreover, similarly to Lemma 11, we can use P to define a coinductive Brouwer tree with leaves in $\mathbb{L}A$, as follows:

$$C'\ P\ l := \begin{cases} \eta_\infty^B\ l & \text{if } P\ l = \mathsf{true} \\ \beta_\infty^B\ (\lambda a.\ C'\ P\ (l +\!\!+ [a])) & \text{if } P\ l = \mathsf{false} \end{cases} \qquad C\ P := C'\ P\ [\,]$$

Then F is coinductively Brouwer continuous via $C\ P\ [\,]$. By assumption, it is Brouwer continuous via some $d : \mathbb{D}_\mathsf{B}$. We show that P is inductively barred by induction on d. ◀

Thus, via a bijection between $\mathbb{L}\mathbb{N}$ and $\mathbb{N}$, the statement that coinductive dialogue continuous functions $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ are dialogue continuous is independent in CIC.

Furthermore, Brede and Herbelin [5] prove that too general forms of GBI are inconsistent. We were unable to reconstruct the proof without classical logic, but sketch the proof assuming double negation elimination in the appendix. It crucially relies on the fact that one can prove that there is no injection from $\mathbb{N} \to \mathbb{P}$ to $\mathbb{N}$.

Theorem 34. $\mathsf{GBI}(\mathbb{N} \to \mathbb{P}, \mathbb{N})$ *is inconsistent assuming classical logic.*

Note that the proof does not go through with $\mathsf{GBI}(\mathbb{N} \to \mathbb{B}, \mathbb{N})$ because an injection $\mathbb{N} \to \mathbb{B}$ to $\mathbb{N}$ is consistent [2]. Furthermore, we remark that the statement that *every* $F : (Q \to A) \to R$ is dialogue continuous implies classical logic for some Q, A, R or even is outright inconsistent. We call this statement $\mathsf{Cont}(Q, A, R)$.

Theorem 35. *We have the following:*

1. *If for a proposition P,* $\mathsf{Cont}(P, \perp, \perp)$*, then P is stable under double negation.*
2. *If for all retracts Q of $\mathbb{N}$,* $\mathsf{Cont}(Q, \perp, \perp)$*, then Markov's principle holds.*
3. $\mathsf{Cont}(\mathbb{N}, \mathbb{B}, \mathbb{P})$ *is inconsistent.*

Proof. (1) is by instantiating the continuity assumption with $\neg\neg P$ and doing a case analysis on the resulting dialogue. For $\eta\,(f : \perp)$ the statement is vacuously true, and for $\beta\,(H : P)\,(k : \perp \to \mathbb{D})$ by assumption. (2) is similar, observing that the dependent pair type $\Sigma n.\ f\,n = \mathsf{true}$ is a retract of $\mathbb{N}$. (3) follows because $Ff := \forall n.fn = \mathsf{true}$ is not continuous: If it would be, in particular it would have a modulus $M : \mathbb{L}\mathbb{N}$ for $fn := \mathsf{true}$. But then we would have $\forall n \in M.\ fn = gn$ for gn reflecting $n \in M$, but $\forall n.\ fn = \mathsf{true}$ and not $\forall n.\ gn = \mathsf{true}$. ◀

7 Related work and Future Work

Since the main contribution of this paper is a discussion of a mechanized zoo of continuity properties in a unified framework, all related work is *de facto* future work w.r.t. an integration into the zoo.

Other notions of continuity for total functions. We are aware of at least two definitions that seem related to continuity in the literature of constructive mathematics that we would like to integrate into the zoo in the future. The first is the notion of the dynamical method from a constructive treatment of algebra by Coste and Lombardi [16]. The second is the notion of predicates with cofinite character due to Misselbeck-Wessel and Schuster [33].

In a setting of *classical* reverse mathematics, where one has access to the law of excluded middle, but restricts comprehension axioms, i.e. which predicates one can form, Sanders [36] analyses the strength of different notions of continuity. It would be interesting to discuss these notions in a constructive setting.

Furthermore, it would be interesting to study continuity for dependent total functions, i.e. of type $(\forall q : Q, Aq) \to R$ where $A : Q \to \mathbb{T}$.

Continuity for partial functions. In this paper, we consider total functions $F : (Q \to A) \to R$. It would be interesting to also consider continuity of partial functions $F : (Q \rightharpoonup A) \rightharpoonup R$, as used to model Turing reducibility by Forster, Kirst, and Mück [20, 21].

For partial functions, one can define continuity for instance via the preservation of least upper bounds w.r.t. the order relating two partial functions f and g if g has the same values as f on all points where f is defined. For total functions, this order is just the identity, thus preservation of least upper bounds does not yield an interesting notion of continuity.

It could even be interesting to generalise the partiality monads to arbitrary monads in such as study.

Continuity axioms. Continuity principles stating that all (total) functions $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ were the explicit motivation of our work. They appear in the literature under different names, e.g. KLS [4], KLST [26] after the Kreisel-Schoenfield-Lacombe theorem in computability theory [32], independently also proved by Ceitin (Tseitin) [8], or Brouwer's continuity principle (BC) [3]. Extensionality issues aside, for the definitions of continuity covered here we end up with three variants of such a principle, which we call Cont here.

Cont for dialogue tree continuity implies Cont for tree function / modulus function continuity, which in turn implies Cont for existential modulus continuity. Our results yield that Δ^0_1-BI and c-BI are respectively enough to prove the implications. However, that the implications of Cont principles are respectively equivalent to Δ^0_1-BI and c-BI does not follow. In future work, we plan to analyse whether the principles can be separated as well as the exact strength of the implications.

Kleene's second algebra. While we discuss what continuity properties are equivalent under which circumstances, we do not discuss which equivalent properties are more convenient to use in formal proofs or other use cases. One benchmark could be to construct Kleene's second algebra in terms of continuous functions $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ using different continuity properties to see which ones allow the most elegant proofs.

References

1 Martin Baillon, Assia Mahboubi, and Pierre-Marie Pédrot. Gardening with the Pythia A Model of Continuity in a Dependent Setting. In *CSL*, volume 216 of *LIPIcs*, pages 5:1–5:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.CSL.2022.5.

2 Andrej Bauer. An injection from the Baire space to natural numbers. *Mathematical Structures in Computer Science*, 25(7):1484–1489, 2015. doi:10.1017/S0960129513000406.

3 Andrej Bauer. Continuity principles and the KLST theorem. Personal blog post, 2023. URL: https://math.andrej.com/2023/07/19/continuity-principles-and-the-klst-theorem/.

4 Michael J. Beeson. The nonderivability in intuitionistic formal systems of theorems on the continuity of effective operations. *Journal of Symbolic Logic*, 40(3):321–346, September 1975. doi:10.2307/2272158.

5 Nuria Brede and Hugo Herbelin. On the logical structure of choice and bar induction principles. In *LICS*, pages 1–13. IEEE, 2021. doi:10.1109/LICS52264.2021.9470523.

6 Luitzen E. J. Brouwer. On the domains of definition of functions. In *From Frege to Gödel: A source book in mathematical logic, 1879–1931*, volume 9, page 446. Harvard University Press, 1927.

7 Luitzen E. J. Brouwer. Über Definitionsbereiche von-Funktionen. *Mathematische Annalen*, 97(1):60–75, 1927. doi:10.1007/BF01447860.

8 Grigori Samuilovitsch Ceitin. Algorithmic operators in constructive complete separable metric spaces. *Doklady Akademii Nauk SSSR*, 128:49–52, 1959.

9 Liron Cohen, Bruno da Rocha Paiva, Vincent Rahli, and Ayberk Tosun. Inductive Continuity via Brouwer Trees. In *MFCS*, volume 272 of *LIPIcs*, pages 37:1–37:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.MFCS.2023.37.

10 Liron Cohen, Yannick Forster, Dominik Kirst, Bruno da Rocha Paiva, and Vincent Rahli. Separating Markov's Principles. In *LICS*, pages 28:1–28:14. ACM, 2024. doi:10.1145/3661814.3662104.

11 Liron Cohen and Vincent Rahli. Realizing Continuity Using Stateful Computations. In *CSL*, volume 252 of *LIPIcs*, pages 15:1–15:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CSL.2023.15.

12 Liron Cohen and Vincent Rahli. $\mathrm{TT}^{\Box}_{\mathcal{C}}$: a Family of Extensional Type Theories with Effectful Realizers of Continuity. *Log. Methods Comput. Sci.*, 20(2), 2024. doi:10.46298/LMCS-20(2:18)2024.

13 The Coq Development Team. The Coq proof assistant, 2024. doi:10.5281/ZENODO.11551307.

14 Thierry Coquand. Metamathematical investigations of a calculus of constructions. Technical Report RR-1088, INRIA, 1989. URL: https://hal.inria.fr/inria-00075471.

15 Thierry Coquand and Gérard P Huet. The calculus of constructions. *Information and Computation*, 76(2/3):95–120, 1988. doi:10.1016/0890-5401(88)90005-3.

16 Michel Coste, Henri Lombardi, and Marie-Françoise Roy. Dynamical method in algebra: effective Nullstellensätze. *Annal of Pure and Applied Logic*, 111(3):203–256, 2001. doi:10.1016/S0168-0072(01)00026-4.

17 Hannes Diener. Constructive reverse mathematics: Habilitationsschrift, 2018. Revised as https://arxiv.org/abs/1804.05495. URL: https://dspace.ub.uni-siegen.de/handle/ubsi/1306.

18 Martín Hötzel Escardó. Continuity of Gödel's System T Definable Functionals via Effectful Forcing. In *MFPS*, volume 298 of *Electronic Notes in Theoretical Computer Science*, pages 119–141. Elsevier, 2013. doi:10.1016/j.entcs.2013.09.010.

19 Martin Escardó and Paulo Oliva. Conversion of dialogue trees to Brouwer trees. https://www.cs.bham.ac.uk//~mhe/dialogue/dialogue-to-brouwer.agda, 2017.

20 Yannick Forster, Dominik Kirst, and Niklas Mück. Oracle computability and turing reducibility in the calculus of inductive constructions. In *APLAS*, volume 14405 of *Lecture Notes in Computer Science*, pages 155–181. Springer, 2023. doi:10.1007/978-981-99-8311-7_8.

21 Yannick Forster, Dominik Kirst, and Niklas Mück. The kleene-post and post's theorem in the calculus of inductive constructions. In *CSL*, volume 288 of *LIPIcs*, pages 29:1–29:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CSL.2024.29.

22 Makoto Fujiwara and Tatsuji Kawai. Equivalence of bar induction and bar recursion for continuous functions with continuous moduli. *Annals of Pure and Applied Logic*, 170(8):867–890, 2019. doi:10.1016/j.apal.2019.04.001.

23 Makoto Fujiwara and Tatsuji Kawai. Characterising Brouwer's continuity by bar recursion on moduli of continuity. *Archive for Mathematical Logic*, 60(1-2):241–263, 2021. doi:10.1007/S00153-020-00740-9.

24 Neil Ghani, Peter Hancock, and Dirk Pattinson. Continuous Functions on Final Coalgebras. *Electronic Notes in Theoretical Computer Science*, 249:3–18, August 2009. doi:10.1016/j.entcs.2009.07.081.

25 Peter Hancock, Dirk Pattinson, and Neil Ghani. Representations of Stream Processors Using Nested Fixed Points. *Logical Methods in Computer Science*, Volume 5, Issue 3, September 2009. doi:10.2168/LMCS-5(3:9)2009.

26 Hajime Ishihara. Continuity properties in constructive mathematics. *Journal of Symbolic Logic*, 57(2):557–565, June 1992. doi:10.2307/2275292.

27 Hajime Ishihara. Reverse mathematics in Bishop's constructive mathematics. *Philosophia Scientae*, (CS 6):43–59, September 2006. doi:10.4000/philosophiascientiae.406.

28 Tatsuji Kawai. Principles of bar induction and continuity on Baire space. *Journal of Logic and Analysis*, 2019. doi:10.4115/jla.2019.11.ft3.

29 Tatsuji Kawai. Representing definable functions of ha^{ω} by neighbourhood functions. *Ann. Pure Appl. Log.*, 170(8):891–909, 2019. doi:10.1016/J.APAL.2019.04.011.

30 Stephen Cole Kleene. Countable functionals. In *Constructivity in mathematics: Proceedings of the colloquium held at Amsterdam, 1957 (edited by A. Heyting)*, Studies in Logic and the Foundations of Mathematics, pages 81–100. North-Holland, Amsterdam, 1959.

31 Stephen Cole Kleene. Recursive Functionals and Quantifiers of Finite Types I. *Transactions of the American Mathematical Society*, 91(1):1, April 1959. doi:10.2307/1993145.

32 Georg Kreisel, Daniel Lacombe, and Joseph R. Shoenfield. Partial recursive functionals and effective operations. In *Constructivity in mathematics: Proceedings of the colloquium held at Amsterdam, 1957 (edited by A. Heyting)*, Studies in Logic and the Foundations of Mathematics, pages 290–297. North-Holland, Amsterdam, 1959.

33 Daniel Misselbeck-Wessel and Peter Schuster. Radical theory of Scott-open filters. *Theoretical Computer Science*, 945:113677, February 2023. doi:10.1016/j.tcs.2022.12.027.

34 Christine Paulin-Mohring. *Extraction de programmes dans le Calcul des Constructions. (Program Extraction in the Calculus of Constructions)*. PhD thesis, Paris Diderot University, France, 1989. URL: https://tel.archives-ouvertes.fr/tel-00431825.

35 Christine Paulin-Mohring. Inductive definitions in the system Coq rules and properties. In *International Conference on Typed Lambda Calculi and Applications*, pages 328–345. Springer, 1993. doi:10.1007/BFb0037116.

36 Sam Sanders. On the computational properties of weak continuity notions. In *Twenty Years of Theoretical and Practical Synergies - 20th Conference on Computability in Europe, CiE 2024, Amsterdam, The Netherlands, July 8-12, 2024, Proceedings*, pages 113–125, 2024. doi:10.1007/978-3-031-64309-5_10.

37 Florian Steinberg, Laurent Thery, and Holger Thies. Computable analysis and notions of continuity in Coq. *Logical Methods in Computer Science*, Volume 17, Issue 2, May 2021. doi:10.23638/LMCS-17(2:16)2021.

38 Jonathan Sterling. Higher order functions and Brouwer's thesis. *Journal of Functional Programming*, 31:e11, 2021. doi:10.1017/s0956796821000095.

39 Anne Sjerp Troelstra and Dirk van Dalen. Constructivism in mathematics. Vol. I. *Studies in Logic and the Foundations of Mathematics*, 26, 1988.

40 Jaap van Oosten. Partial Combinatory Algebras of Functions. *Notre Dame Journal of Formal Logic*, 52(4):431–448, 2011. doi:10.1215/00294527-1499381.

41 Li-yao Xia, Yannick Zakowski, Paul He, Chung-Kil Hur, Gregory Malecha, Benjamin C. Pierce, and Steve Zdancewic. Interaction trees: representing recursive and impure programs in Coq. *Proc. ACM Program. Lang.*, 4(POPL):51:1–51:32, 2020. doi:10.1145/3371119.

A Proofs

We here give proofs left out in the main text.

Lemma 3. *A function F of appropriate type is Brouwer continuous iff it is dialogue continuous.*

Proof. It is quite clear that a Brouwer tree can be turned into a dialogue tree. The converse is a bit more tricky. We follow the Agda proof from Escardó and Oliva [19]. We first define a function $\texttt{follow} : A \to \mathbb{D}_\mathsf{B} \to \mathbb{D}_\mathsf{B}$ that skips a node when we already know the answer.

$$\texttt{follow}\ \_\ (\eta_\mathsf{B}\ r) := \eta_\mathsf{B}\ r \qquad\qquad \texttt{follow}\ a\ (\beta_\mathsf{B}\ k) := k\ a$$

Then, given a node $\beta\ q\ k$, we build a Brouwer tree of height q, then prune the branches of the tree using `follow` to get rid of questions $q' < q$. To do this, we implement an alternative node function $\beta'_\mathsf{B} : \mathbb{N} \to (A \to \mathbb{D}_\mathsf{B}) \to \mathbb{D}_\mathsf{B}$ as follows:

$$\begin{aligned}\beta'_\mathsf{B}\ 0\ k &:= \beta_\mathsf{B}\ (\lambda a : A.\ \texttt{follow}\ a\ (k\ a))\\ \beta'_\mathsf{B}\ (\mathsf{S}\ i)\ k &:= \beta_\mathsf{B}\ (\lambda a : A.\ \beta'_\mathsf{B}\ i\ (\lambda a' : A.\ \texttt{follow}\ a\ (k\ a'))).\end{aligned}$$

The conversion function $\texttt{DtoB} : \mathbb{D} \to \mathbb{D}_\mathsf{B}$ is then the following:

$$\texttt{DtoB}\ (\eta\ r) := \eta_\mathsf{B}\ r \qquad\qquad \texttt{DtoB}\ (\beta\ q\ k) := \beta'_\mathsf{B}\ q\ (\lambda a : A.\ \texttt{DtoB}\ (k\ a))$$

Finally, we prove by induction on d that `DtoB` d computes the same function as d. ◀

Lemma 27. *If $Q = \mathbb{N}$, and F has a continuous modulus of continuity function, then F has a self-modulating modulus of continuity function.*

Proof. For this proof, we use an equivalent definition of modulus continuity, where the modulus is just a natural number rather than a list of questions. We call $n : \mathbb{N}$ a number modulus for $F : (\mathbb{N} \to A) \to R$ at $f : \mathbb{N} \to A$ if the list $[f\ 0; \ldots; f\ n]$, is a modulus for F at f. F has a number modulus at f (respectively a number modulus of continuity function) if and only if it has a modulus of continuity at f (resp. a modulus of continuity function). Note that if k is a number modulus for F at f, then so is any natural number $l \geq k$.

Let M be a continuous number modulus of continuity function for F, i.e.:

$$\forall f g.\ (\forall m \leq M\ f, f\ m = g\ m) \to F\ f = F\ g$$

It suffices to construct a self-modulating number modulus function N for F. Following Fujiwara and Kawai, we introduce for $f : \mathbb{N} \to A$ and $n : \mathbb{N}$ the function $f \lfloor n : \mathbb{N} \to A$

$$f \lfloor n := \lambda m.\ \textbf{if}\ m \leq n\ \textbf{then}\ f\ m\ \textbf{else}\ f\ 0.$$

Observe that a natural number k such that $M\ (f \lfloor k) \leq k$ is actually a number modulus at f itself. In this case indeed, the corresponding modulus is short enough to only concern values that are common to f and $f \lfloor k$. We thus define:

$$N\ f := \mu k.\ M\ (f \lfloor k) \leq k$$

This candidate N is computable. Indeed, $M\ (f \lfloor k_f) \leq k_f$ for $k_f = (M\ f) + k + 1$, and we can thus look up for the minimal such k. Now by the previous observation, N is a number modulus function for f. Finally, for any f and g such that $\forall m < N\ f, f\ m = g\ m$, note that $f \lfloor m$ and $g \lfloor m$ *coincide* for any $m \leq Nf$, and therefore so do $M\ (f \lfloor m)$ and $M(g \lfloor m)$. In particular, $M\ (g \lfloor N\ f) \leq N\ f$ thus $N\ g \leq N\ f$. But by minimality, for any $k < N\ f$, we have $M\ (g \lfloor k) = M\ (f \lfloor k) > k$ and thus $N\ g \geq N\ f$. Therefore N is self-modulating. ◀

Lemma 31. $\Delta_1^0\text{-}\mathsf{GBI}(\mathbb{N}, A) \leftrightarrow \Delta_1^0\text{-}\mathsf{BI}(A)$.

Proof. We define the position-annotating function $\mathtt{ord} : \mathbb{L}A \to \mathbb{L}(\mathbb{N} \times A)$:

$$\mathtt{ord}\, l := [(k, l_k) \mid 0 \leq k < |l|].$$

Given a predicate P on $\mathbb{L}(\mathbb{N} \times A)$, we can turn it into a predicate on $\mathbb{L}A$: $\|P\|\, l := P\,(\mathtt{ord}\, l)$. Conversely, given $U : \mathbb{L}A \to \mathbb{P}$, we can turn it into a predicate $\widehat{U} : \mathbb{L}(\mathbb{N} \times A) \to \mathbb{P}$ as follows:

$$\widehat{U}\, l := \exists l_0 : \mathbb{L}A.\ l = \mathtt{ord}\, l_0 \wedge U\, l_0.$$

If P is barred then $\widehat{P}$ is $(\mathbb{N}, A)$-barred. However, the converse is not true. Let us take a predicate T such that T is true on lists of the form $[(2, \_)]$ and false anywhere else. Then T is $(\mathbb{N}, A)$-barred but $\|T\|$ is not barred. This would only be the case if T was monotone, i.e. $l \subseteq l' \to T\, l \to T\, l'$.

Likewise, because of the condition $l \subseteq l'$ in the first rule of $\mathcal{H}^{\mathsf{G}}_{Q,A}$, the fact that $\widehat{U}$ is $(\mathbb{N}, A)$-inductively barred only implies that the *monotonization* of U, which is the smallest monotone predicate U_m such that $U\, l \to U_m\, l$, is inductively barred. This is where the proof in [5] fails.

Fortunately, when P is decidable, then so is $(\widehat{P^m})_m$, for P^m the smallest predicate such that $P\, l \to P^m\, l'$ as soon as l is a prefix of l'. This suffices to recover inductive barredness of P. We refer the interested reader to the formalisation for the rest of the proof. ◀

Theorem 34. $\mathsf{GBI}(\mathbb{N} \to \mathbb{P}, \mathbb{N})$ *is inconsistent assuming classical logic.*

Proof. We define the following predicate on $\mathbb{L}((\mathbb{N} \to \mathbb{P}) \times \mathbb{N})$.

$$T\, l := \exists (f f' : \mathbb{N} \to \mathbb{P})(n : \mathbb{N}).f \neq f' \wedge (f, n) \in l \wedge (f', n) \in l$$

We have to prove that T is $(\mathbb{N} \to \mathbb{P}, \mathbb{N})$-barred, i.e. $\forall F : (\mathbb{N} \to \mathbb{P}) \to \mathbb{N}.\ \exists l : \mathbb{L}(\mathbb{N} \to \mathbb{P}).\ T\,[(f, Ff) \mid f \in l]$. Applying double-negation elimination, it is enough to prove falsity of the following statement:

$$\begin{aligned}&\exists F : (\mathbb{N} \to \mathbb{P}) \to \mathbb{N}.\ \forall u : \mathbb{L}(\mathbb{N} \to \mathbb{P}).\ \forall (f f' : \mathbb{N} \to \mathbb{P}).\\ &\quad \forall n : \mathbb{N}.\ (f, n) \in [(x,\, F\, x) \mid x \in u] \to (f', n) \in [(x,\, F\, x) \mid x \in u] \to f = f'\end{aligned}$$

This statement asserts the existence of an injection from $\mathbb{N} \to \mathbb{P}$ into $\mathbb{N}$, which is known to be disprovable. Thus T is $(\mathbb{N} \to \mathbb{P}, \mathbb{N})$-barred. By $\mathsf{GBI}(\mathbb{N} \to \mathbb{P}, \mathbb{N})$, it is $(\mathbb{N} \to \mathbb{P}, \mathbb{N})$-inductively barred. By definition, this means that $\mathcal{H}^{\mathsf{G}}_{\mathbb{N}\to\mathbb{P},\mathbb{N}} T\,[\,]$ holds. However, let us notice that $T\,[\,]$ does not hold. Generalising a bit, it is thus enough to prove $\forall (l : \mathbb{L}(\mathbb{N} \to \mathbb{P} \times \mathbb{N})).\, \mathcal{H}^{\mathsf{G}}_{\mathbb{N}\to\mathbb{P},\mathbb{N}}\, T\, l \to \neg(T\, l) \to \bot$. We go by induction on the proof of $\mathcal{H}^{\mathsf{G}}_{\mathbb{N}\to\mathbb{P},\mathbb{N}}\, T\, l$.

In the leaf case, $T\, l$ for some $l \subseteq l'$, then $T\, l'$ which is inconsistent with our hypothesis.

In the branching case, by induction hypothesis we get a function $f : \mathbb{N} \to \mathbb{P}$ and we have to produce a natural number n such that $\neg(T(l +\!\!+ [(f, n)]))$. It is sufficient to pick any natural number not already present in the list l. ◀