Consistent Ultrafinitist Logic

Michał J. Gajda

Migamake Pte Ltd
mjgajda@migamake.com

Ultrafinitism (Kornai 2003; Podnieks 2005; Yessenin-Volpin 1970; Gefter 2013; Lenchner 2020) postulates that we can only reason and compute relatively short objects (Seth Lloyd 2000; Krauss and Starkman 2004; Sazonov 1995; S. Lloyd 2002; Gorelik 2010), and numbers beyond certain value are not available. Some philosophers also question the physical existence of real numbers beyond a certain level of accuracy (Gisin 2019). This approach would also forbid many forms of infinitary reasoning and allow removing many from paradoxes stemming from a countable enumeration.

However, philosophers still disagree on whether such a finitist logic could be consistent (Magidor 2007), while constructivist mathematicians claim that “no satisfactory developments exist” (Troelstra 1988). We present preliminary work on a proof system based on Curry-Howard isomorphism (Howard 1980) and explicit bounds for computational complexity.

This approach invalidates logical paradoxes that stem from a profligate use of transfinite reasoning (Benardete 1964; Nolan forthcoming; Schirn and Niebergall 2005), and assures that we only state problems that are decidable by the limit on input size, proof size, or the number of steps (Tarski, Mostowski, and Robinson 1955).

Consideration of complexity also solves other paradoxes, in particular the “paradox of inference” existing in classical theory of semantic information (Bar-Hillel and Carnap 1953; Duzi 2010). Using a bound on cost and depth of the term for each inference, we independently developed a very similar approach to that used for cost bounding in higher-order rewriting (Vale and Kop 2021).

By finitism we understand the mathematical logic that tries to absolve us from transfinite inductions (Kornai 2003). Ultrafinitism goes even further by postulating a definite limit for the complexity of objects that we can compute with (Seth Lloyd 2000; Krauss and Starkman 2004; Sazonov 1995; S. Lloyd 2002; Gorelik 2010). We assume these without committing to a particular limit.

In order to permit only ultrafinitist\(^1\) inferences, we postulate ultraconstructivism: we permit proofs, or constructions that are not just strictly computable, but for which there is a bound on amount of computation that is needed to resolve them. That means that we forbid proofs that go for an arbitrarily long time and require a deadline for any proof or computation.

For the sake of generality, we will attach this deadline in the form of bounding function that takes as arguments depths of input terms, and outputs the upper bound on the number of steps that the proof is permitted to make. Depths of input terms are a convenient upper bound on the complexity of normalized proof terms (those without the cut.)

Please note that notation \(\forall x_v : A \rightarrow \alpha(v), B\) has a size variable \(v\) declared as a depth of term variable \(x\), and then bound in polynomials \(\alpha(v)\) and \(\beta(v)\). The notation \(\alpha(1)\) is a shortcut for \(\alpha(1/v)\) in the rules abs and app.

\(^1\)Also called strict finitist by (Magidor 2007).
Size variables: \( v \in V \)
Term variables: \( x \in X \)
Positive naturals: \( i \in \mathbb{N} \setminus \{0\} \)
Polynomials: \( \rho ::= v | i | \rho * \rho | \rho^\rho \ | \text{iter}(\rho, \rho, v) | \rho^{\mathbb{N}/\rho} \)
Data size bounds: \( \alpha ::= \beta \)
Computation bounds: \( \beta ::= \rho \)
Types: \( \tau ::= v | \tau \wedge \tau | \tau \vee \tau | \forall x_v : \tau \to^\beta \tau | \bot \ | \circ \)
Terms: \( E ::= v | \lambda v.E | \text{in}_{1}(E) | \text{in}_{i}(E) | (E, E) | () \)

Environments: \( \Gamma ::= v_1 : \tau^1_{\beta_1}, \ldots, \tau^n_{\beta_n} \)
Judgements: \( J ::= \Gamma \vdash_\beta E : \tau \)

<table>
<thead>
<tr>
<th>( \Gamma \vdash_\beta \gamma \beta : A )</th>
<th>( \Gamma \vdash_\beta \gamma \beta : A )</th>
<th>( \Gamma \vdash_\beta \gamma \beta : A )</th>
<th>( \Gamma \vdash_\beta \gamma \beta : A )</th>
<th>( \Gamma \vdash_\beta \gamma \beta : A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Gamma, x : A \vdash_\beta \gamma \beta : x : A )</td>
<td>( \Gamma \vdash_\beta \gamma \beta : A )</td>
<td>( \Gamma \vdash_\beta \gamma \beta : A )</td>
<td>( \Gamma \vdash_\beta \gamma \beta : A )</td>
<td>( \Gamma \vdash_\beta \gamma \beta : A )</td>
</tr>
</tbody>
</table>

After elision of bounds and rule _subsume_ we see the rules for intuitionistic logic. Thus consistency can be proved by the consistency of intuitionistic logic(Brouwer 1981; Van Dalen 1986; Serensen and Urzyczyn 1998). Every valid proposition with a fixed bound on input \( n \) can be checked by enumerating inputs, and is thus decidable. It is easy to show that our logic can emulate bounded loop programs(Meyer and Ritchie 1967) which have power equivalent to primitive recursive functions( Robinson 1947). Expressing statements about undecidability implicitly requires unbounded computational effort. Since all our proofs and arguments are explicitly bounded, there is no room to state undecidability. We thus define statements that are both true, and computable a given limit(Gorelik 2010).
References


