Decidability in Dependently Constrained Universal Existential Paths

by Sven Nilsen, 2018

The universal existential path `∃` is undecidable. However, is the dependently constrained universal existential path decidable? In this paper I show that decidability depends on whether the dependently typed variable is finite or infinite. This has consequences for when safe AI boxing is decidable.

A dependently constrained universal existential path is the following:

$$\forall\ x : X, A, B : X \to \text{type}\ \{\ \exists\{A(x) \to B(x)\} : (A(x) \to B(x)) \to (B(x) \to \text{bool})\ \}$$

It takes a function of type `A(x) → B(x)` and returns a new function that tells what the input function returns. The returned function returns `true` if a value is returned for some input and `false` otherwise.

The question is whether it is possible to construct a such function. If it is undecidable, then it is impossible to write down the source code that executes on a Turing machine.

Assume a type of `x` that has only one member. This reduces to:

$$\forall\ A, B : \text{type}\ \{\ \exists\{A \to B\} : (A \to B) \to (B \to \text{bool})\ \}$$

If both `A` and `B` are finite, then this becomes decidable.

Likewise, if the type of `x` had two members, one could construct a constrained universal existential path from the reduced problem above by using `$2^2 = 4$` sub-constrained universal existential paths, if both `A` and `B` are finite. By proof by induction, this means that if the type of `x` is finite, then constructing a constrained universal existential path is decidable, if both `A` and `B` are finite.

However, if `x` infinite, then even if `A(x)` is finite, there are infinitely many types to constrain the universal existential path over, which means the problem becomes undecidable. Notice that one must not assume additional constraints, e.g. `B(x) : (= false)` or something. The problem is only undecidable when considering all constrained types.

Therefore, assuming `A(x)` and `B(x)` are finite, the problem is decidable if and only if the type of `x` is finite. This has consequences for the interpretation of proof of decidability for safe AI boxing.

In safe AI boxing, the Safety Interruption Oracle (SIO) uses the world as a dependently typed variable. Since the world might be thought of as consisting of infinite states (a simplified assumption), the problem becomes undecidable. However, if the world is assumed to consist of finite states, then the problem becomes decidable. This happens because `Input <=> A` and `Behavior <=> B` are both finite, so decidability only depends on whether the world is finite or not. It draws the “line in the sand”, figuratively speaking, when safe AI boxing is decidable.