AGI programming language

Tom Rochette <tom.rochette@coreteks.org>

July 24, 2025 — daae079c

0.1 Context

0.2 Learned in this study

0.3 Things to explore

- What are the differences between a programming language for humans and for AGI?
- Should an AGI programming language consider the typed/untyped dichotomy?

1 Overview

The programming language of an AGI does not need to be similar to the one we use. It is convenient if it is similar to those we are used to, but it is not necessary. It may be possible for the AGI to translate its programs into human readable programs on request.

2 Features of an AGI language

- Variables do not need names. They are simply placeholders (e.g., a memory address). The goal is to reduce the number of functions with the exact same syntax, structure and function calls from an infinite set to a finite one (albeit potentially enormous)
- Functions do not need names. They are simply placeholders
- Functions are indexed by their parameters type and return type. This allows for the retrieval of potential functions to use given a set of argument types and expected return type
- Every construct (conditions, loops, function calls, operations, variables, etc.) are each identified by a unique number
  - What the AGI is attempting to do is to take these building blocks and put the pieces that fit together (two integer variables in a function call that expects two integer as parameters)

3 See also

4 References