Type-guided synthesis for dynamic languages

Thomas Logan

October 2022

1 Introduction

Previous work in program synthesis and type inference [] has assumed ML-style languages, possibly enriched with refinement types. ML terms are dynamically limited to those defined by some countable set of constructions, defined using ML's datatype mechanism. As such, it is always possible to infer a type with a countable bound, that is sound without rejecting too many idiomatic programs.

It is also a straightforward search to synthesize a term from a type, since a type are limited to finite set of constructor primitives.

In a dynamic language, dynamically accepted terms are not limited to those taken from a countable set. A type system that always restricts terms to countable sets, would reject many idiomatic programs that are accepted dynamically. However, having no restrictions would also allow many erroneous programs to slip through. Finding the balance of when to restrict terms statically is a key challenge.

Manually typing programs has been shown to be error-prone [].

To alleviate the error-prone task of specifying types, we infer types striking the balance between strict types and lenient types.