Gradual Types as Error Suppression

A Constructive View of Type Warnings

LILY BROWN, ANDY FRIESEN, and ALAN JEFFREY, Roblox, USA

This paper presents the view of gradual typing adopted by the Luau programming language. Prior work on gradual typing has been based on *type compatibility*, that is a relation on types $T \sim U$ given by contextually closing $T \sim$ any $\sim U$. Type systems based on type compatibility use $\sim$ rather than type equivalence (or a similar presentation for languages with subtyping). We take a different tack, which is to view type warnings *constructively* as a proof object $\mathrm{Warn}(\Gamma \vdash M : T)$ saying that the type derivation $\Gamma \vdash M : T$ should generate a warning. Viewing type warnings constructively allows us to talk about *error suppression*, for example the any type is error suppressing, and so this type system is gradual in the sense that developers can explicitly annotate terms with the any type to switch off type warnings. This system has the usual "well-typed programs don't go wrong" result for program which do not have explicit type annotations with error-suppressing types, except this property can now be stated as the presence of $\mathrm{Warn}(\Gamma \vdash M : T)$ rather than the absence of a run-time error. This system has been deployed as part of the Luau programming language, used by millions of users of Roblox Studio.

CCS Concepts: • **Software and its engineering** → **Semantics**.

1 INTRODUCTION

TODO

2 FURTHER WORK

TODO

A PRAGMATIC SEMANTIC SUBTYPING

TODO

REFERENCES

Authors' address: Lily Brown; Andy Friesen; Alan Jeffrey, Roblox, San Mateo, CA, USA.