Pontifícia Universidade Católica
do Rio de Janeiro

André Murbach Maidl

Typed Lua: An Optional Type System for Lua

TESE DE DOUTORADO

Thesis presented to the Programa de Pós Graduação em Informática of the Departamento de Informática, PUC–Rio as partial fulfillment of the requirements for the degree of Doutor em Informática

Advisor : Prof. Roberto Ierusalimschy
Co–Advisor: Prof. Fabio Mascarenhas de Queiroz

Rio de Janeiro
April 2015

Pontifícia Universidade Católica
do Rio de Janeiro

André Murbach Maidl

Typed Lua: An Optional Type System for Lua

Thesis presented to the Programa de Pós Graduação em Informática, of the Departamento de Informática do Centro Técnico Científico da PUC–Rio, as partial fulfillment of the requirements for the degree of Doutor.

Prof. Roberto Ierusalimschy
Advisor
Departamento de Informática — PUC–Rio

Prof. Fabio Mascarenhas de Queiroz
Co–Advisor
UFRJ

Prof. Ana Lúcia de Moura
Departamento de Informática — PUC-Rio

Prof. Edward Hermann Haeusler
Departamento de Informática — PUC-Rio

Prof. Anamaria Martins Moreira
UFRJ

Prof. Roberto da Silva Bigonha
UFMG

Prof. José Eugênio Leal
Coordinator of the Centro Técnico Científico da PUC–Rio

Rio de Janeiro, April 10th, 2015

André Murbach Maidl

The author graduated in Computer Science from Pontifícia Universidade Católica do Paraná — PUCPR in 2004, and obtained the degree of Mestre at Universidade Federal do Paraná — UFPR in 2007. He obtained the degree of Doutor at PUC-Rio in 2015, where he worked in the field of programming languages.

Bibliographic data

Maidl, André Murbach

Typed Lua: An Optional Type System for Lua / André Murbach Maidl; advisor: Roberto Ierusalimschy; co–advisor: Fabio Mascarenhas de Queiroz. — 2015.

149 f. : il. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2015.

Inclui bibliografia.

1. Informática – Teses. 2. Linguagens de script. 3. Lua. 4. Sistemas de tipos. 5. Sistemas de tipos opcionais. 6. Tipagem gradual. I. Ierusalimschy, Roberto. II. Queiroz, Fabio Mascarenhas de. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Acknowledgments

First of all I would like to thank three indispensable people in my life: my wife, my mother, and my father. Izabella is my inexhaustible source of inspiration, and she is always pushing me forward. My mother is a very brave person, and she inspires me to never give up. My father is a very kind person, and he inspires me to try being more sensitive. I am thankful to them not only because the support they gave me while writing this thesis, but also because they are the reason of my life.

This work would have not happened without all the help from my advisors. Roberto has a singular view of computer science that pushed me to pursuit the best I could do, while Fabio gave me the confidence I needed about the work we were doing. I am glad that both of them had patience to guide me through this work, and for all the nice meetings we had that led me to learn a lot of new concepts and ideas.

It is hard to complete a thesis without having some leisure moments, so I thank my friend and roommate André Ramiro for all the crazy and fun moments that we had, which helped me to keep focused during working hours.

I also thank my friends at LabLua for the nice work environment, specially Hisham and Pablo for all the support during the hardest moments.

It may look easy moving to Rio, as it is a sunny place with lots of things to do, but it may not be as easy as it looks for someone that grew up in a cloudy and rainy town. Thus, I thank my cousin Rebecca and my aunt Nazira for all the help and affection they gave me during my adaptation in Rio, which were fundamental to keep my goals in mind.

I thank the guys from Rio de Janeiro Fixed Gear (RJFG) for all the fun we had during the night rides, and for helping me to discover Rio.

I am also most grateful to the professors Ana Lúcia de Moura, Anamaria Martins Moreira, Edward Hermann Haeusler, and Roberto da Silva Bigonha for their careful review that helped to improve this work.

Finally, I would like to thank CAPES, Google Summer of Code, and PUC-Rio for partially funding this work.

Abstract

Maidl, André Murbach; Ierusalimschy, Roberto (Advisor); Queiroz, Fabio Mascarenhas de (Co-Advisor). **Typed Lua: An Optional Type System for Lua**. Rio de Janeiro, 2015. 149p. Doctoral Thesis — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Dynamically typed languages such as Lua avoid static types in favor of simplicity and flexibility, because the absence of static types means that programmers do not need to bother with abstracting types that should be validated by a type checker. In contrast, statically typed languages provide the early detection of many bugs, and a better framework for structuring large programs. These are two advantages of static typing that may lead programmers to migrate from a dynamically typed to a statically typed language, when their simple scripts evolve into complex programs. Optional type systems allow combining dynamic and static typing in the same language, without affecting its original semantics, making easier this code evolution from dynamic to static typing. Designing an optional type system for a dynamically typed language is challenging, as it should feel natural to programmers that are already familiar with this language. In this work we present and formalize the design of Typed Lua, an optional type system for Lua that introduces novel features to statically type check some Lua idioms and features. Even though Lua shares several characteristics with other dynamically typed languages such as JavaScript, Lua also has several unusual features that are not present in the type system of these languages. These features include functions with flexible arity, multiple assignment, functions that are overloaded on the number of return values, and the incremental evolution of record and object types. We discuss how Typed Lua handles these features and our design decisions. Finally, we present the evaluation results that we achieved while using Typed Lua to type existing Lua code.

Keywords

Scripting languages; Lua; Type systems; Optional type systems; Gradual typing

Resumo

Maidl, André Murbach; Ierusalimschy, Roberto; Queiroz, Fabio Mascarenhas de. **Typed Lua: um sistema de tipos opcional para Lua**. Rio de Janeiro, 2015. 149p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Linguagens dinamicamente tipadas, tais como Lua, não usam tipos estáticos em favor de simplicidade e flexibilidade, porque a ausência de tipos estáticos significa que programadores não precisam se preocupar em abstrair tipos que devem ser validados por um verificador de tipos. Por outro lado, linguagens estaticamente tipadas ajudam na detecção prévia de diversos *bugs* e também ajudam na estruturação de programas grandes. Tais pontos geralmente são vistos como duas vantagens que levam programadores a migrar de uma linguagem dinamicamente tipada para uma linguagem estaticamente tipada, quando os pequenos *scripts* deles evoluem para programas complexos. Sistemas de tipos opcionais nos permitem combinar tipagem dinâmica e estática na mesma linguagem, sem afetar a semântica original da linguagem, tornando mais fácil a evolução de código tipado dinamicamente para código tipado estaticamente. Desenvolver um sistema de tipos opcional para uma linguagem dinamicamente tipada é uma tarefa desafiadora, pois ele deve ser o mais natural possível para os programadores que já estão familiarizados com essa linguagem. Neste trabalho nós apresentamos e formalizamos Typed Lua, um sistema de tipos opcional para Lua, o qual introduz novas características para tipar estaticamente alguns idiomas e características de Lua. Embora Lua compartilhe várias características com outras linguagens dinamicamente tipadas, em particular JavaScript, Lua também possui várias características não usuais, as quais não estão presentes nos sistemas de tipos dessas linguagens. Essas características incluem funções com aridade flexível, atribuições múltiplas, funções que são sobrecarregadas no número de valores de retorno e a evolução incremental de registros e objetos. Nós discutimos como Typed Lua tipa estaticamente essas características e também discutimos nossas decisões de projeto. Finalmente, apresentamos uma avaliação de resultados, a qual obtivemos ao usar Typed Lua para tipar código Lua existente.

Palavras–chave

Linguagens de script; Lua; Sistemas de tipos; Sistemas de tipos opcionais; Tipagem gradual

Contents

List of Figures

List of Tables