

# Dispozicija doktorske disertacije

Žiga Lukšič

Mentor: dr. Matija Pretnar

## **KAKO SOSEDU UKRASTI SENDVIČ BREZ DA OPAZI (HOW TO STEAL YOUR NEIGHBOURS SANDWICH WITHOUT BEING NOTICED)**

### **Pregled znanstvenega področja in dosedanjih raziskav**

Uporaba algebraskih učinkov in prestreznikov je del področja teorije tipov, čigar vpliv na razvoj programskih jezikov hitro narašča. Algebraski učinki nam omogočajo preprosto in generično razširitev funkcijskih programskih jezikov z računskimi učinki, kot na primer izpis rezultatov v datoteko ali spreminjanje podatkovne baze. Eden od pristopov k temu problemu je uporaba tako imenovanih monad, vendar moramo pri tem pristopu primerno prilagoditi celotno kodo programa. Uporaba algebraskih učinkov nam omogoča, da namesto tega računske učinke obravnavamo zgolj v zunanjem sloju kode, s tako imenovanimi algebraskimi prestrezniki.

Raziskave področja so se začele zgolj z uvedbo algebraskih učinkov [3, 4], temu pa je kasneje sledil razvoj algebraskih prestreznikov [5], ki posplošijo princip delovanja prestreznikov napak z možnostjo nadaljevanja izvajanja. Izkaže se, da lahko z algebraskimi prestrezniki modeliramo večino zelenih računskih učinkov, prav tako pa je ekvivalenten monadičnemu pristopu.

Uporaba algebraskih učinkov prav tako omogoča sledenje računskim učinkom skozi sistem tipov. Pri tem želimo ohraniti lastnosti kot so polimorfizmi in varnost tipov. Razvoj sistema tipov z učinki je šel v smeri t.i. 'row-types' [2] in pa 'subtyping' [6].

Ključnega pomena za razumevanje in dokazovanje v programskih jezikih so pristopi, kjer programom podamo primeren matematičen pomen. Funkcijski programski jeziki uporabljajo strukturno denotacijsko semantiko, ki sloni na uporabi teorije domen in teorije kategorij.

tako se lepo sklopijo z obstoječo denotacijsko semantiko funkcijskih programskih jezikov [1], kar nam ponuja vpogled v matematični pomen programov.

### **Opis raziskovalne vsebine**

Razvoj zanesljivih in ekspresivnih sistemov tipov skrbi tako za varnost programov kot tudi za dokazovanje lastnosti in pravilnosti programske kode. Z uporabo algebraskih

učinkov lahko sistem tipov razširimo na način, ki nam omogoča natančno sledenje računskih učinkov. Hkrati pa nam teorija algebraskih učinkov nudi tudi enačbe, ki veljajo med učinki. Sistem tipov učinkov tako razširimo na sistem tipov teorij, ki poleg možnih učinkov hranijo tudi enačbe teorije. Te enačbe nam omogočajo lažje dokazovanje lastnosti programske kode, dodatno varnost s pomočjo iskanja napak v kodi, in odpirajo vrata novim pristopom k optimizaciji kode s pomočjo enačb.

## Raziskovalna vprašanja

Želim pokazati, da implementacija enačb v programski jezik z algebraskimi učinki ohrani vsaj del strukture teorije učinkov. Nadalje želim raziskati lastnosti, katerim mora zadoščati programski jezik, da v pripadajoči denotacijski semantiki ohranimo čim več koristnih lastnosti teorije. Poleg razvoja matematičnega ogrodja za teorijo učinkov in konstrukcije primernih logik za dokazovanje, nameravam sistem tudi uporabiti na primernih problemih, kot recimo analiza statističnih modelov.

## Pričakovani rezultati in prispevek k znanosti

Pričakovani rezultati so kul teorija, ki se bo v dejanskih programskih jezikih pojavila šele po tem, ko Elon pristane na Marsu.

## Literatura

- [1] A. Bauer and M. Pretnar. An effect system for algebraic effects and handlers. *Logical Methods in Computer Science*, 10(4), 2014.
- [2] D. Hillerström and S. Lindley. Liberating effects with rows and handlers. In J. Chapman and W. Swierstra, editors, *Proceedings of the 1st International Workshop on Type-Driven Development, TyDe@ICFP 2016, Nara, Japan, September 18, 2016*, pages 15–27. ACM, 2016.
- [3] G. D. Plotkin and J. Power. Adequacy for algebraic effects. In F. Honsell and M. Miculan, editors, *Foundations of Software Science and Computation Structures, 4th International Conference, FOSSACS 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2-6, 2001, Proceedings*, volume 2030 of *Lecture Notes in Computer Science*, pages 1–24. Springer, 2001.
- [4] G. D. Plotkin and J. Power. Algebraic operations and generic effects. *Applied Categorical Structures*, 11(1):69–94, 2003.
- [5] G. D. Plotkin and M. Pretnar. Handlers of algebraic effects. In G. Castagna, editor, *Programming Languages and Systems, 18th European Symposium on Programming, ESOP 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, volume 5502 of *Lecture Notes in Computer Science*, pages 80–94. Springer, 2009.

- [6] A. H. Saleh, G. Karachalias, M. Pretnar, and T. Schrijvers. Explicit effect subtyping. In A. Ahmed, editor, *Programming Languages and Systems - 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10801 of *Lecture Notes in Computer Science*, pages 327–354. Springer, 2018.