

Implementing a type system in Python (BSc/MSc)

An implementation of NLC[†], a type system for natural language compositionality, has been done in Coq (<https://gitlab.com/eluuk/nlc>) but in order to be usable in a real-world scenario, it should be implemented in a language well integrated w/ NLP, i.e. Python. The obvious problem is that Coq is an interactive theorem prover and a functional programming language with a complex static type system while Python is an imperative scripting language w/ only dynamic (duck) typing. The lack of static types in Python has to be leveraged somehow (e.g. w/ custom classes and/or set-theoretic operations) to accommodate the types of NLC. There are some ideas and code snippets on how this might work but you should contribute w/ more ideas, testing, analysis, and (eventually) developing. Ideally, you would feel comfortable w/ Python before choosing this topic. Depending on the projected result, this could be a BSc or MSc thesis topic. A knowledge of Estonian is a plus (but not required), as the system may be also tested on Estonian examples.

[†] http://ut.ee/~el/pubs/dev_lopstr

Erkki Luuk
erkkil@gmail.com
#Python #natural_language #type_system