Evaluation of a Cal Poly MS Thesis in the Area of Programming Languages

Title of Thesis Localized Type Inference of Atomic Types in Python

Author: Brett Cannon

Date of Publication: June 2005

Institution: Cal Poly Department of Computer Science

Type of thesis: project, experimental

Area of work: programming languages

1. Problem definition

Score: 5

The thesis specifies a very specific goal -- to improve the runtime performance of the Python interpreter by 5%. The specific technical problem is how to achieve this goal by adding localized type inference to the interpreter.

In addition to stating the general problem, Chapter 1 of the thesis does a very good job of laying out the subsidiary technical problems. These are the problems of adding typing to a language such as Python that has no explicit type declarations.

The thesis delivers a very good solution to the problem, in terms of its design, implementation, and testing. Unfortunately, the performance-improvement goal is not achieved. This in no way detracts from the quality of the thesis itself. In fact, achieving the negative result provides valuable insight into the problem, and provides fodder for future work.

2. Writing Quality

Score:3.8

The author uses the "royal we" in a number of places, and some colloquial language. In terms of clarity, there a number of long-winded passages, that include sentences that are too long. As is often observed in technical writing textbooks, using short crisp sentences tends to improve clarity.

The following is a specific sentence from Chapter 1 that exemplifies the writing issues:

> *"Specifically, we explore if introducing type inference into Python's compiler along with type-specific bytecodes for Python's interpreter will lead to at least a 5% performance increase[1] across various benchmarks without any semantic changes to the compiler or language.*
>
> ---
>
> *[1]The 5% goal has been chosen since it is an informal rule of thumb used by Python's development team as a measurement of whether something is worth the added code complexity."*

This would be better in multiple sentences, without the gratuitous use of footnoting.

3. Contribution to the state-of-the-art in the field

Score: 4

The thesis contributes a valuable piece of information, though it is not extremely ground-breaking. The reason that the contribution is not more highly rated is that the thesis focuses on a specific language, and a narrow aspect of typing, i.e., atomic types. In terms of presentation, the concluding chapter does a good job of summarizing the specific contributions of the thesis.

There are 26 references, to very appropriate places. Chapter 4 specifically compares and contrasts the approach of the thesis to others.

4. Originality and innovativeness

Score: 4

The thesis explores an interesting specific idea, though, again, not ground breaking. The work done here with Python has been done by others for a number of other languages.

5. Technical depth

Score: 5

The review of type inference and related issues is suitably deep. Specifically, chapters 2 through 6 provide a very thorough discussion of type theory, type checking, design and implementation.

The design and implementation of the project itself are non-trivial. The subject matter is well presented in Chapters 5 and 6.

6. Implementation

Score: 5

The implementation is fully functional, and meets the requirements set out in the introduction. Chapters 5 and 6 cover the implementation in depth.

7. Validation of the work, as appropriate to the subject matter

Score: 5

The experimentation is well set up, it uses appropriate benchmarks, and gathers real data. A solid chapter 7 presents the experimental results.

As noted above, the performance-improvement goal was not achieved. Again, this does not detract from the worthiness of the work as an MS thesis.

8. Potential for publication

Score: 3.5

A partially-refereed version of the results has appeared in the PyCon conference. This is a conference dedicated to work related to the Python language.

Due to the negative results, and similarity to work in other languages, it is doubtful that the work could be published in a top-flight conference, such as the ACM SIGPLAN compilers conference. Without additional work and better results, it is not journal publishable, e.g., in ACM TOPLAS.

9. Potential for future research

Score: 5

There is definitely potential for future research, given the negative results. A full chapter (9) does a good job of laying out what specifically could be done.

10. Overall quality of the thesis

Score: 4.8

Despite the negative performance results, it's a very solid piece of work.