

Отзыв на выпускную квалификационную работу
«Статический вывод типов для языка
Python в интегрированной среде
разработки»

Мирошникова Владислава Игоревича

Типизация в языках программирования является общепризнанным способом повышения надёжности ПО. Наличие типовых контрактов позволяет компиляторам автоматически обнаруживать их нарушения и отвергать соответствующие программы. Кроме того, сам тип предоставляет дополнительную информацию, которая может быть использована для повышения выразительности программ. С другой стороны, общеизвестно, что, в силу фундаментальных результатов о неразрешимости, при статической типизации репрессиям неизбежно подвергнется и ряд добропорядочных программ. В связи с этим актуальной задачей является разработка промежуточных систем статической и динамической типизации.

Один из наиболее популярных языков программирования, Python, снабжен полностью динамической объектной системой типов. В программах на Python не только типы переменных могут меняться “на ходу”, но даже сами типы могут быть модифицированы, поскольку они являются сущностями времени исполнения. Это повышает выразительность языка и открывает широкие возможности для метапрограммирования. С другой стороны, те же самые свойства затрудняют разработку надёжных программ, поскольку граница между “хорошими” и “плохими” программами, определяемая в терминах типизации, оказывается размытой. Для преодоления этой трудности, во-первых, для языка Python разработана специальная подсистема типовых “подсказок” (type hints), которая позволяет разработчикам специфицировать подразумеваемые контракты, а, во-вторых, существует несколько инструментов, которые в той или иной мере решают задачу вывода типов, извлекая информацию о таких контрактах непосредственно из кода программы. Подчеркнем, что данная задача в общем виде для языка Python является алгоритмически неразрешимой.

В работе Владислава решается задача статического вывода типов в языке Python. За основу применённого подхода были взяты общеизвестные и хорошо себя зарекомендовавшие методы вывода типов, основанные на унификации и так называемом “row-полиморфизме”. Любопытно, что, несмотря на кажущуюся очевидность выбора, ни один из известных инструментов не применял такие методы в явном виде. Для достижения

результата было необходимо ознакомиться с соответствующими теоретическими результатами и творчески применить их к другому языку, в котором зачастую нарушаются инварианты, от которых зависят важные теоретические свойства типовых систем. Кроме того, необходимо было учитывать контекст выполнения данной работы для целей создания IDE. В результате проделанной работы был разработан подход к статическому выводу типов для языка Python, который учитывает все основные свойства языка. Данный подход был использован для создания промышленного инструмента и показал сравнимую эффективность с аналогичными существующими средствами.

Владислав проявил себя ответственно и профессионально, показал способность к критическому прочтению научной литературы. Работа проходила в размеренном режиме, в соответствии с согласованным графиком.

В свете вышесказанного я рекомендую данную работу к защите и предлагаю оценку **“отлично”**.

Научный руководитель
доц. кафедры системного программирования
к.ф.-м.н. Булычев Д. Ю.

