0.1. *Гунько Т.В.* Проектирование и разработка статического анализатора промежуточных представлений "LuNA-программ"

В области параллельного программирования семантические ошибки создают серьезные проблемы, так как они не обнаруживаются компилятором и включают не только ошибки, свойственные последовательным программам, но и специфические для параллелизма. Один из подходов к обнаружению семантических ошибок — это статические анализаторы.

Код для системы автоматического конструирования параллельных программ “LuNA” [1] совмещает два языка программирования: LuNA для описания прикладного алгоритма и C++ для реализации его операций. LuNA — это высокоуровневый инструмент параллельного программирования, в котором пользователю не нужно организовывать управление процессами и потоками на низком уровне. Проблема заключается в том, что на данный момент нет статического анализатора, охватывающего логику всей программы полностью.

По этой причине была поставлена цель спроектировать и реализовать статический анализатор, который по совокупности LuNA и C++ кода мог бы выявлять семантические ошибки в местах взаимодействия языков программирования.

Принцип работы анализатора основывается на переводе кода каждого языка программирования в своё промежуточное представление “LLVM IR”. После успешного перевода необходимо объединить результаты в одно представление, отражающее полную логику исходной “LuNA-программы”. Следующим шагом станет использование библиотеки, основанной на “Data-flow” и “Monotone” анализах промежуточных представлений, чтобы отыскать ошибки потока данных.

На данный момент программа позволяет переводить полностью C++ и частично LuNA код — планируется закончить перевод и воспользоваться библиотекой “PhASAR” [2] для написания собственных проверок.

Проверки будут направлены на следующие ошибки [3]: несоответствие типов аргументов функций; нарушения безопасности памяти (dangling pointers, use-after-free или null dereferences при обмене указателями); использование неинициализированных значений (uninitialized variables, приводящее к undefined behavior); а также проблемы с calling conventions и control flow (несовместимые соглашения вызовов или unreachable code).

Научная новизна данного проекта заключается в том, что для фрагментированных программ впервые будет применен метод статического анализа, способный обнаруживать семантические ошибки «на стыке» различных языков программирования.

Научный руководитель — канд. техн. наук, доц. Власенко А. Ю.

Список литературы

[1] Malyshkin V. E., Perepelkin V. A., Schukin G. A. Scalable Distributed Data Allocation in LuNA Fragmented Programming System. // Journal of Supercomputing. 2016. P. 726-–732.

[2] Malyshkin V., Vlasenko A., Michurov M. Automated Debugging of Fragmented Programs in LuNA System. // D. Balandin et al. (Eds.): 22nd International Conference, MMST 2022. 2022. P. 266–280.

[3] Schubert P. D., Hermann B., Bodden E. PhASAR: An Inter-procedural Static Analysis Framework for C/C++. // Tools and Algorithms for the Construction and Analysis of Systems. 2019. Vol. 11428. N. 1. P. 393—410.