Анализ решения задачи поток данных для уменьшения числа ложных срабатываний

Кошелев Владимир Константинович

Аспирант

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: vedun@ispras.ru

В настоящее время для поиска программных дефектов широко используются программные продукты, основанные на статическом анализе[2]. Одним из наиболее распространённых видов статического анализа программ с целью обнаружения дефектов является анализ потоков данных[2]. При помощи данного анализа могут быть обнаружены такие виды дефектов как разыменование нулевого указателя или утечка ресурсов. При этом анализ потоков данных является нечувствительным к путям - условия перехода на ребрах графа потока управления не учитываются. При анализе реальных программ отсутствие учёта условий перехода может повлечь за собой высокий процент ложных срабатываний. Для уменьшения процента ложных срабатываний можно, либо изменить сам алгоритм распространения фактов анализа потоков данных, либо дополнительно обработать результаты анализа потоков данных. В данной работе рассматривается именно дополнительная обработка результатов анализа с целью уменьшения числа ложных срабатываний.

Здесь и далее предполагается, что передаточные функции являются дистрибутивными, а множество фактов - конечным.

Так как передаточные функции являются дистрибутивными, то решение внутрипроцедурной задачи потоков данных может быть представлено в виде направленного графа G, вершинами которого являются пары <v, d>, где v – вершина исходного графа потока управления, а d – факт задачи потоков данных[1]. Вершина <v, d> из графа G имеет следующий смысл: в вершине v факт d верен. Ребро из вершины <v, d> в вершину <u, d'> означает, что в графе потока управления есть ребро из вершины v в вершину u и результат передаточной функция ребра <v, u> для факта d включает факт d'.

Далее рассмотрим граф G обладающий следующими свойствами:

- граф G имеет только один сток и один исток.
- все вершины графа G достижимы из стока.
- исток достижим из всех вершин графа.

Данный граф, с точки зрения поиска дефектов, имеет следующий смысл. Вершина исток является источником ошибки: например, в ней была выделена память, которая в последующим была не освобождена, или переменная, которая в будущем будет разыменована, была присвоена нулю. Вершина сток является местом возникновения ошибки, например, в ней была утеряна последняя ссылка на выделенную память или нулевая переменная была разыменована. Таким образом, любой путь в данном графе приводит к возникновению ошибки. Таким образом, чтобы говорит о том, что ошибка была обнаружена, необходимо показать, что хотя бы один путь выполнения программы проходит через граф G.

Дополнительно потребуем, чтобы все рёбра в графе G не являлись недостижимым, то есть для каждого ребра найдётся такой путь выполнения, который пройдёт по данному ребру. Назовём вершину внутренней, если она не является ни стоком ни истоком в графе G.

Далее введём понятие исходящей и входящей внутренней вершины: Назовём вершину <v, d> входящей, если найдётся такая вершина u из графа потока управления, что ни для какого факта d‘ ребро <u, d‘>, <v, d> не входит в граф G, однако ребро <u, v> присутствует в графе потока управления.

Соответственно, если найдётся такая вершина u из графа потока управления и такой факт d‘, что ребро <v, d>, <u, d‘> не входит в G, хотя <v, u> входит в граф потока управления, то исходную вершину <v, d> назовём исходящей.

Утверждение: Если в графе G нет одновременно и исходящей внутренней вершины и входящей внутренней вершины (возможно одной и той же), то найдётся путь выполнения, проходящий по графу G от источника к стоку.

Данное утверждение было использовано при поиске дефектов связанных с использованием освобождённого указателя на программном продукте tizen версии 2.2.1. В результате анализа была получена точность анализа более 60% при совпадении результатов более 80% по сравнению с современными анализаторами[2].

Литература

1. Reps T. Horwitz S. Sagiv M. Precise Interprocedural Dataflow Analysis via Graph Reachability // University of Wisconsin
2. Иванников В. П. Белеванцев А. А. Бородин А. Е. Игнатьев В. Н. Журихин Д. М. Статический анализатор Svace для поиска дефектов в исходном коде программ. Труды Института системного программирования РАН. 2014. Т. 26. № 1. Стр. 231-250.