УДК 004.441

РАЗРАБОТКА МЕТОДА ДЛЯ МЕЖЪЯЗЫКОВОГО СЕМАНТИЧЕСКОГО АНАЛИЗА ТЕКСТОВ ПРОГРАММ

Орловский М.Ю. (ИТМО)

Научный руководитель – кандидат технических наук, доцент Логинов И.П. (ИТМО)

Введение.

Современные программные проекты, в отличие от многих программных проектов прошлого, гораздо чаще состоят из набора разных (порой разительно) языковых решений, предназначенных для решения определенного круга задач.

Так как используемые в проекте модули программного кода, за редким исключением, практически всегда семантически связаны между собой, то возникает необходимость поддержания согласованности таких модулей. Таким образом, в рамках тезиса рассматривается проблема поддержания согласованности модулей, реализованных на различных языках программирования.

Основная часть.

В рамках исследования решено было сосредоточится на наиболее полезных сценариях использования, поддерживающих разработчика в процессе кодирования. В качестве эталона для этого был выбран Language Server Protocol [1]. Исходя из сценариев использования метода можно сформировать следующие требования:

- отзывчивость (задержка анализа до трех секунд при использовании кэшированных данных);
- гибкость (возможность проведения разных объемов анализа);
- универсальность (независимость от применяемых в проекте технологий);
- корректность (анализ должен производить минимальное количество ложноположительных результатов).

Согласно данным сценариям, предполагается следующий вид анализа: уникальные (в рамках области видимости) идентификаторы выявляются для дальнейшего их связывания между модулями, реализованными на разных языках, по принципу определение - ссылка. Для обеспечения такого анализа необходимо решить четыре общие задачи: обеспечение анализа операционного окружения; извлечение программного кода для анализа; обеспечение уникальности идентификаторов и их областей видимости; корректная и наиболее полная типизация идентификаторов исходя из семантики языка, в котором идентификатор фигурирует.

Для конкретного операционного окружения возможна реализация специального модуля, анализирующего его и поставляющего его анализ. Поставляемая информация может являться отражать значения системных переменных или версии системных библиотек. Модуль, совершающий такой анализ является *провайдером конфигурации*. Задача по извлечению структуры проекта из операционного окружения ложится на провайдера конфигурации, а само извлечение будет производиться модулем *извлечения кода*.

Обеспечение уникальности идентификаторов реализуется через фреймворк *графов областей видимости*, используемый, например, в Spoofax [2]. Основная идея фреймворка – создание системы, позволяющей конфигурировать специфичные правила разрешения имен и областей видимости с приведением их к общей форме через введение *ограничений видимости*. Также, извлекая набор *ограничений типизации*, можно типизировать идентификаторы для обеспечения корректности дальнейшего связывания. Модуль, реализующий такой анализ является модулем *трансляции*. Такие ограничения будут разрешаться логическим решателем, например Z3 [3].

Все вышеприведенные модули связаны определенным протоколом взаимодействия и конфигурацией. Конфигурацию будем называть *онтологией*, так как она будет содержать общие для предметной области определения: типы, языки и параметры окружения.

В качестве иллюстрации работы метода рассмотрим реализацию *мультиязыкового языкового сервера*.

Предположим, что существует проект, реализованный на HTML и JavaScript, представляющий собой простое веб-приложение. В коде JavaScript с использованием DOM API происходит запрос к определенному узлу дерева, объявленного статически в HTML. Инструментальное средство путем запроса к LSP адаптеру получает анализ такого кода на межъязыковом уровне – связь между кодом JavaScript и тегами HTML. То есть запрос через DOM API в результате статически связывается с определенной локацией в HTML документе. Такой подход позволяет получить информацию о наличии связи, а также её назначение (так как известна пара связываемых языков). В результате пользователь видит всплывающее окно в IDE с данными о такой связи, что может быть использовано, например, в автодополнении. Конкретную реализацию анализа данных такого рода можно предоставить инструментальному средству.

Заключение.

Таким образом, метод соответствует заявленным требованиям и является самодостаточным решением, позволяющим реализовывать различные анализаторы межъязыковых зависимостей. В качестве дальнейшего развития данного метода предполагаются следующие направления исследования:

- использование методов машинного обучения для определения языка по входной строке, такое дополнение позволит повысить полноту анализа;
- использование формы кода, позволяющей вовлекать в анализ не только идентификаторы, но и выражения (например SSA форма [4]).

1. Официальная страница Language Server Protocol – [Электронный ресурс]. – Режим доступа: https://microsoft.github.io/language-server-protocol/ (дата обращения: 02.02.2024).
2. Описание графов областей видимости в Spoofax – [Электронный ресурс]. – Режим доступа: https://spoofax.dev/references/statix/scope-graphs/ (дата обращения: 02.02.2024).
3. Официальный ресурс проекта Z3 – [Электронный ресурс]. – Режим доступа: https://github.com/Z3Prover/z3 (дата обращения: 21.01.2024).
4. R. A. Kelsey, ‘A correspondence between continuation passing style and static single assignment form’, San Francisco, California, USA, 1995, pp. 13–22.