

РЕЦЕНЗИЯ

на выпускную квалификационную работу бакалавра
«Символьный анализ приложений с реляционными базами данных»
выполненную студентом
Старцевым Матвеем Сергеевичем

В современном программном обеспечении задачам обеспечения корректности и безопасности приложений уделяется первостепенное внимание. Автоматическая генерация тестов для фреймворка Spring, основанная на методах статического анализа и символьного исполнения, является одним из перспективных подходов, однако её применение к системам, взаимодействующим с реляционными базами данных, сопряжена с рядом проблем: сложность анализа ORM-фреймворка Spring JPA, взрыв пространства состояний и т.д.

Выпускная работа Старцева М. С. посвящена разработке и внедрению подсистемы символьного моделирования реляционных баз данных в инструмент USVM для Java-приложений, использующих Spring JPA. Автор поставил следующие задачи: обзор существующих методов, разработка символьной модели для работы с базами данных, её реализация и экспериментальное исследование.

Содержание работы

- **Первая глава:** формулировка цели и задач работы.
- **Вторая глава:** обзор ключевых подходов — символьное исполнение и его ограничения; SMT-решатели; методы статического анализа БД; архитектура USVM и механизмы аппроксимаций; особенности Spring JPA.
- **Третья глава:** разработка символьной модели реляционных таблиц, описание иерархии классов ITable и операций (Filter, Map, Join, Sort и др.) для отложенных преобразований.
- **Четвёртая глава:** детали реализации — генерация кода через BlockGenerationContext; DatabaseGenerator и JcTableCollector; решение проблемы валидации данных; обёртки ListWrapper, SetWrapper; обработка пользовательских data-классов.
- **Пятая глава:** тестирование реализованного модуля на проектах Spring-PetClinic и Spring-Test-Demo с сравнением покрытия: без модели – 0–11 %, с моделью – 0–62 %.

Достоинства работы

1. Полный обзор методов статического анализа и символьного исполнения в контексте реляционных БД.
2. Полноценная интеграция разработанной модели в инструмент USVM.
3. Свойство ленивости символьной модели значительно улучшает её эффективность.
4. Экспериментальное исследование демонстрирует работоспособность модели на базовых примерах.
5. Сложность и объем проведенной работы.

Недостатки и замечания

1. Отсутствует обсуждение теоретических и практических ограничений предлагаемого подхода, что важно для оценки его применимости в разных сценариях.
2. В тексте встречаются опечатки и стилистические ошибки: например, некорректная фраза “будет отсекает” на странице 36.
3. В экспериментальной части не приведены метрики по времени символьного исполнения с моделью и без — это препятствует оценке практической эффективности.
4. Для тестирования выбраны относительно простые проекты, что может не выявить принципиальных проблем моделирования реляционных БД.
5. Не обоснован выбор ограничения на максимальный размер таблиц (250 000 инструкций) и его влияние на полноту тестового покрытия.

Вопросы к автору

1. Какие теоретические и практические ограничения накладывает ленивый символьный подход на анализ сложных SQL-запросов и объём базы данных?
2. Планируется ли протестировать модуль на более сложных проектах (с большим числом сущностей и связей), чтобы выявить потенциальные проблемы моделирования? Какие проблемы могут быть выявлены?
3. Как повлияет на корректность и производительность анализа расширение поддержки символьной модели для дополнительных конструкций Spring JPA, таких как агрегатные функции (COUNT, GROUP BY), операторы JOIN FETCH и режим ленивой загрузки (LAZY)?

Выпускная квалификационная работа Старцева М. С. соответствует требованиям, предъявляемым к выпускным работам бакалавров по направлению «Программная инженерия», и заслуживает оценки «отлично»