

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет информационных технологий

Кафедра компьютерных технологий

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Образовательная программа: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

«РЕФАКТОРИНГ ТРАНСЛЯТОРА ЯЗЫКА REFLEX НА ОСНОВЕ АВТОМАТИЧЕСКОЙ
ПАРСЕР-ГЕНЕРАЦИИ»

(название темы прописными буквами)

утверждена распоряжением проректора по учебной работе №___ от «__»_____ 20__ г.

скорректирована распоряжением проректора по учебной работе №___ от «__»_____ 20__ г.

Бастрыкина Алена Алексеевна, группа 16203

(Фамилия, Имя, Отчество студента, группа)

(подпись студента)

«К защите допущена»

Руководитель ВКР

Заведующий кафедрой,

ученая степень, звание

ученая степень, звание

должность, место работы в НГУ

...../.....

...../.....

(ФИО) / (подпись)

(ФИО) / (подпись)

«.....».....20...г.

«.....».....20...г.

Дата защиты: «.....».....20...г.

Новосибирск, 20__г.

Оглавление

Введение.....	3
1. Анализ предметной области.....	4
1.1. Анализ специфики языка Reflex.....	4
1.2. Анализ методов создания трансляторов предметно-ориентированных языков.....	5
1.3. Сравнительный анализ ПО для автоматической парсер-генерации.....	6
1.4. Формулировка требований к транслятору.....	9
Список литературы.....	10

Введение

Использование языков общего назначения для программирования цифровых систем управления ведет к росту сложности программной архитектуры и алгоритмов, увеличению стоимости разработки и затрудняет поддержку таких систем.

Для решения этой проблемы в Институте автоматики и электрометрии СО РАН был разработан процесс-ориентированный язык Reflex [4]. В настоящее время, планируется создание IDE для языка Reflex с возможностью интеграции дополнительных компонентов для анализа и преобразования программ на Reflex, например: интерактивный редактор кода, модуль кодогенерации, модули динамической и статической верификации.

Для языка Reflex был разработан транслятор в процедурный язык C. Парсер этого транслятора был написан на языке C по грамматике языка Reflex. Такая реализация затрудняет сопровождение транслятора и его интеграцию со сторонними модулями.

Необходимо предложить более гибкую реализацию транслятора, с возможностью предоставления API для взаимодействия с новыми компонентами.

Цель работы — рефакторинг транслятора языка Reflex с использованием методов автоматической парсер-генерации.

Для достижения этой цели были поставлены следующие задачи:

1. Анализ специфики языка Reflex;
2. Сравнительный анализ методов создания трансляторов DSL-языков и ПО для автоматической парсер-генерации;
3. Формулировка требований к транслятору языка Reflex, определение архитектуры транслятора;
4. Выбор формата внутреннего представления программы (AST);
5. Описание грамматики языка Reflex;
6. Выбор алгоритмов семантического анализа и кодогенерации;

7. Реализация транслятора.

Использование инструментов парсер-генерации при разработке трансляторов предметно-ориентированных языков дает преимущество в случае модификации или расширения синтаксиса языка, поскольку часть работы по изменению транслятора в таких случаях может быть выполнена автоматически.

1. Анализ предметной области

1.1. Анализ специфики языка Reflex

Процесс-ориентированный язык Reflex предназначен для программирования систем управления в области промышленной автоматизации. Отличительными особенностями таких систем являются: наличие объекта управления, непрерывное воздействие на систему управления, а так же реакция системы на внешние события, определяемая алгоритмом управления [1].

Язык Reflex обеспечивает необходимые средства для описания задач, обладающих такими особенностями. Программа на языке Reflex состоит из множества процессов, каждый процесс характеризуется набором состояний. Предоставляются средства для организации взаимодействия и синхронизации процессов: запуск и остановка процессов, переменные, разделяемые между процессами, операции с временными интервалами. Также, в языке Reflex существует спецификация для удобного описания входных и управляющих сигналов.

Язык Reflex иногда называют «Си с процессами» [4], в силу схожести с процедурным языком C. Reflex был создан как диалект языка C [4], что упрощает его преобразование в компилируемый язык C, а также предоставляет возможность использования функций из библиотек, написанных на языке C.

1.2. Анализ методов создания трансляторов предметно-ориентированных языков

Предметно ориентированный язык программирования (англ. domain-specific language, DSL) – язык программирования, предназначенный для решения круга задач определенной предметной области. Предметно ориентированные-языки облегчают разработку и поддержку программных систем с особой спецификой, позволяя не прибегать к сложным средствам языков общего назначения [5, 6].

Распространенным подходом, который был использован при создании языка Reflex, является трансляция предметно-ориентированного языка в какой-либо язык общего назначения. Для языка Reflex на данный момент существуют алгоритмы преобразования в компилируемый язык C и интерпретируемый Python [3].

Условно, процесс трансляции исходного кода из одного языка в другой можно разделить на следующие этапы [2]:

1. Лексический анализ. При лексическом анализе исходный код программы преобразуется в поток лексем.
2. Синтаксический анализ. На этом этапе происходит распознавание синтаксической структуры и порождение промежуточного представления или объектной модели языка.
3. Генерация кода, в ходе которой осуществляется семантический анализ компонент промежуточного представления или объектной модели языка и перевод промежуточного представления или объектной модели в объектный код.

Наряду с основными этапами процесса трансляции возможна также оптимизация промежуточного представления программы и оптимизация объектного кода.

В настоящее время, существует большое множество инструментов, облегчающих процесс разработки трансляторов для формальных языков. В частности, к таким средствам относятся *парсер-генераторы*. Парсер-

генераторы принимают на вход описание грамматики входного языка в нужной нотации и порождают программу на определенном языке (парсер), которая может осуществлять синтаксический анализ исходного кода. Парсер-генераторы используются при разработке языков с простой грамматикой, парсеры для языков программирования со сложной структурой реализуются без автоматической генерации кода. Ввиду простоты языка Reflex, при реализации транслятора целесообразно использовать такие инструменты.

Так же, среди средств создания предметно-ориентированных языков, хотелось бы выделить полноценные программные среды для разработки DSL. Такие инструменты предоставляют не только возможность генерации парсеров, но еще и возможность генерации редактора для языка с подсветкой синтаксиса и ошибок, а так же предоставляют средства для облегчения кодогенерации. Примерами таких сред являются Xtext [7] и JetBrains MPS [6].

1.3. Сравнительный анализ ПО для автоматической парсер-генерации

Для сравнительного анализа были выбраны следующие инструменты:

- ANTLR 4 [8] – парсер-генератор;
- GNU Bison – парсер-генератор;
- Xtext [7] – среда для разработки DSL, в том числе предоставляющая парсер-генератор.

Обобщенные результаты анализа представлены в Таблице 1.

Сравнение инструментов происходило по следующим пунктам:

- Язык сгенерированного парсера

С помощью всех перечисленных средств можно сгенерировать парсер на кроссплатформенном языке Java, Bison так же поддерживает C++ и C, ANTLR4 дополнительно поддерживает сравнительно большое множество языков.

- Среда исполнения

При выборе средства большим преимуществом является кроссплатформенность. Средой исполнения Xtext и ANTLR4 является Java-

машина, что делает эти средства кроссплатформенными, Bison может исполняться только на Unix-подобной платформе.

- Среда разработки (IDE)

Для Xtext и ANTLR4 существуют полноценные IDE с редактором и визуальным отображением грамматик, что упрощает разработку. Для GNU Bison не существует подобных средств.

- Анализатор лексем

Xtext и ANTLR генерируют лексический анализатор при генерации парсера. Недостаток Bison в том, что для сгенерированного им парсера нужен сторонний анализатор (например, lex).

- Описание грамматики

Все средства используют РБНФ для описания грамматик. Xtext предоставляет специальный язык для описания грамматик, ANTLR и Bison используют похожие нотации.

- Структура синтаксического дерева

Bison не предусматривает автоматической генерации синтаксического дерева. Однако структура синтаксического дерева и алгоритм ее генерации может определить сам пользователь. Выделение памяти под структуру и ее заполнение нужно описать в файле с грамматикой с помощью блоков кода, выполнение которых происходит при посещении парсером нетерминальных символов. Такой подход увеличивает время на разработку и является неудобным с точки зрения отладки транслятора.

Парсер, сгенерированный ANTLR4, порождает структуру ParseTree в результате работы. Эта структура представляет собой синтаксическое дерево с возможностью обхода в глубину и получения текстового содержания узлов.

Парсер Xtext в результате синтаксического анализа создает промежуточное представление исходного кода программы в виде структуры EMF (Eclipse Modelling Framework).

	Xtext	ANTLR4	GNU Bison
Язык сгенерированного парсера	Java	C#, Java, Python, JavaScript, C++, Swift, Go	C, C++, Java
Платформа или среда исполнения	JVM	JVM	UNIX-like
IDE	Плагин для Eclipse IDE	ANTLR Studio, плагины для Eclipse и IntelliJ IDEA	нет
Анализатор лексем	Генерируется вместе с парсером	Генерируется вместе с парсером	Для работы требуется сторонний анализатор (lex)
Описание грамматики	Специальный DSL для описания грамматики	Нотация, похожая на Yacc	Yacc
Структура синтаксического дерева	Модель EMF	Класс ParseTree с методом walk, возможность добавлять различные обработчики на посещение узлов при обходе	Конкретный формат синтаксического дерева определяется пользователем, его создание нужно описать в файле с грамматикой

Таблица 1. Сравнительный анализ ПО

Огромным преимуществом средства Xtext для поставленной задачи является то, что оно позволяет автоматически сгенерировать редактор с подсветкой синтаксиса. Таким образом, отпадает необходимость разработки редактора для RIDE 2.0. По другим пунктам сравнения, Xtext не уступает ANTLR4 и Bison, либо превосходит их. Поэтому, по результатам анализа, было выбрано средство Xtext.

1.4. Формулировка требований к транслятору

В результате анализа предметной области, были сформулированы следующие требования:

- Синтаксис для парсера должен быть англоязычным, с ключевыми словами в нижнем регистре;
- Транслятор должен производить синтаксический и семантический анализ, и сообщать об ошибках, указывая их локализацию в исходном коде;
- Целевым языком транслятора должен быть C;
- Транслятор должен быть реализован на языке Java, грамматика должна быть описана в нотации, воспринимаемой средством Xtext.

Список литературы

1. Зюбин В. Е. Процесс-ориентированное программирование: Учеб. пособие / В. Е. Зюбин. – Новосибирск. Новосиб. гос. ун-т., 2011. – 194 с.
2. Основы разработки трансляторов: конспект лекций / А. И. Легалов – URL: <http://www.softcraft.ru/translat/lect/> – Текст: электронный
3. Зюбин В. Е. Процесс-ориентированная технология программирования: модели, языки и инструментальные средства для спецификации алгоритмов управления сложными техническими системами / В. Е. Зюбин – Новосибирск, 2013. – С 7.
4. Зюбин В.Е Язык «Рефлекс» – диалект Си для программируемых логических контроллеров / В.Е. Зюбин – 6 с.
5. Martin Ward. Language Oriented Programming. /— Computer Science Department, Science Labs, 1994. – 19 p.
6. Языково-ориентированное программирование: следующая парадигма / Сергей Дмитриев – URL: <http://rsdn.org/article/philosophy/LOP.xml/> – Текст: электронный
7. Xtext - Documentation / Eclipse Foundation. – URL: <https://www.eclipse.org/Xtext/documentation/> – Текст: электронный
8. ANTLR / Terence Parr. – URL: <https://www.antlr.org/> – Текст: электронный