

Федеральное агентство железнодорожного транспорта
Федерально бюджетное государственное образовательное учреждение
высшего образования

Иркутский государственный университет путей и сообщения

Кафедра «Информационные системы»

Научно-исследовательская практика
«Обзор литературы»

Отчёт по практике

Выполнил:
магистрант группы ПИМ.1-16-1
Арляпов С.В.

Проверил:
доцент каф. ИСиЗИ
Звонков И.В.

Иркутск
2016

Постановка задачи

Согласно поределению, данному выдающемуся учёным А.М. Летовым, стоящим у истоков современной теории управления и сделавшим многое для её развития, теория управления «есть совокупность методов позволяющих выработать и обосновать решение, которое применяется для достижения заранее поставленной цели в условиях каких-либо определённой ситуации». В частности, теория автоматического управления – «наука о методах определения законов управления какими-либо объектами, допускающих реализацию с помощью технических средств автоматики»[1].

Книга «Интеллектуальное управление динамическими системами» посвящена методам интеллектуального управления и интеллектуального анализа и проектирования динамических управляемых систем, объектами которых является конкретный класс систем – движущиеся объекты технической природы. 3 глава этой книги посвящена применению процедур автоматизации вывода как альтернативе освоенным в инженерной практике алгоритмическим процедурам. Демонстрация применения логических языков для представления знаний и интеллектуального управления приведена в главе 3 на примере управления группой пассажирских лифтов.

Из выше описанной книги берётся задача реализовать интеллектуальное управление группой пассажирских лифтов на основе системы автоматического доказательства теорем (АДТ).

Система АДТ

Одним из подходов к интеллектуализации программных систем является разработка алгоритмов обработки информации, основанных на моделировании процесса рассуждений. Наиболее формализованные подходы базируются на автоматическом построении логического вывода (ЛВ) в некоторой системе формализованных знаний (логического описания предметной области). Программные системы для поиска ЛВ называют системами автоматического доказательства теорем (АДТ), поскольку утверждения, для которых существует ЛВ, являются теоремами (в заданном исчислении)[2].

Формализации некоторых предметных областей, например, аксиоматизации математических теорий, верификация программного и аппаратного обеспечения, являются весьма громоздкими, и использование систем АДТ для построения ЛВ становится необходимым инструментом исследования. Кроме того, системы АДТ используются в качестве механизма ЛВ, например, в системах планирования действий, системах поддержки принятия решений, экспертных системах, где важным фактором является время решения задачи и другие критерии. Далее, в работе, под «поиском ЛВ», как правило, будем иметь в виду автоматизированный поиск ЛВ.

Автоматическое доказательство теорем с обоснованием его эффективности широко используется в таких областях как верификация программных и аппаратных систем, синтез программного обеспечения, решение проблем матема-

тики, логическое программирование, дедуктивные базы данных и др[2].

Система АДТ

Позитивно-образованными формулами (ПО-формулами, ПОФ) называется вид формул, для записи которых используются только позитивные типовые кванторы \forall и \exists :

Пусть X — множество переменных, и A — конъюнкт.

1. $\exists_x A$ и $\forall_x A$ есть \exists -ПОФ и \forall -ПОФ соответственно.
2. Если $F = \{F_1, \dots, F_n\}$ есть \forall -ПОФы, тогда F есть \exists -ПОФ.
3. Если $F = \{F_1, \dots, F_n\}$ есть \exists -ПОФы, тогда F есть \forall -ПОФ.
4. Любая \exists -ПОФ или \forall -ПОФ есть ПОФ.

Данные формулы не содержат операторов отрицания. Также ПО-формула является особым видом записи классических формул языка предикатов, подобно КНФ, ДНФ и др., поскольку любая формула языка предикатов первого порядка представима как позитивно-образованная формула.

Канонический вид ПО-формулы начинается с $\forall \emptyset$. Очевидно, что любая ПОФ приводима к каноническому виду.

Типовые кванторы $\forall \emptyset$ и $\exists \emptyset$ называются фиктивными, поскольку не влияют на истинность формулы и не связывают никаких переменных, а только лишь служат конструкциями сохраняющими корректную запись ПО-формулы.

Для удобства ПО-формулы представляются в древовидной форме:

$$Q_x A : \{F_1, \dots, F_n\} \equiv Q_x A : \begin{cases} F_1 \\ \dots \\ F_n \end{cases},$$

где F_i — ПО-формула, A — набор атомарных формул, Q некоторый квантор, который отличен от кванторов в начале формул F .

Некоторые части канонической ПО-формулы имеют специальные названия:

1. Корневой узел $\forall \emptyset$ называется корнем ПО-формулы;
2. Дочерние узлы корня ПО-формулы имеют вид $\exists_x A$ и называются базами ПО-формулы, конъюнкт A называется базой фактов, а вся подформула начинающаяся с базового узла называется базовой подформулой;
3. Дочерние узлы баз имеют вид $\forall_x B$ и называются вопросами к родительской базе. Если вопрос является листовым узлом $\forall_x B \equiv \forall_x B : false$, то он называется целевым вопросом.
4. Поддеревья вопросов называются консеквентами или следствиями. Следствием целевого вопроса является *false*.

Система АДТ PRISNIF

В ИДСТУ СО РАН А.А. Ларионов и Е.А. Черкашин занимаются исследованием поиска ЛВ на базе ПО-формул. Основным результатом исследований является программная система АДТ PRISNIF. Применение системы для решения

ряда задач из библиотеки TRTP показывает, что система соответствует мировому уровню в данной области.

Сама система реализована на языке программирования D, но новая версия системы АДТ разрабатывается на языке программирования Rust.

Стоит подчеркнуть, что система АДТ PRISNIF реализует поиск ЛВ на базе ПО-формул. Этот факт, задаёт к реализации программных модулей некоторый условия, которые учитывались в разработке программного продукта.

Также стоит отметить, что в ИДСТУ СО РАН лаборатория Информационно-управляющих систем нуждается в подобной системе для использования её в качестве автоматической системы управления.

SymPy

SymPy представляет собой открытую библиотеку символьных вычислений на языке Python. Цель SymPy - стать полнофункциональной системой компьютерной алгебры (CAS), при этом сохраняя код максимально понятным и легко расширяемым. SymPy полностью написан на Python и не требует сторонних библиотек.

SymPy можно использовать не только как модуль, но и как отдельную программу. Программа удобна для экспериментов или для обучения. Она использует стандартный терминал IPython, но с уже включенными в нее важными модулями SymPy и определенными переменными x , y , z .

Заключение

Используя выше представленные материалы, можно попытаться реализовать интеллектно управляемую динамическую систему на основе системы автоматического доказательства теорем на базе поитивно образованных формул.

Источники

1. С. Н. Васильев. Интеллектуальное управление динамическими системами / С. Н. Васильев, А. К. Жерлов, Е. А. Федосов, Б. Е. Федунев - М.. Физико-математическая литература, 200. - 352 с.

2. А. А. Ларионов. Программные технологии для эффективного поиска логического вывода в исчислении позитивно-образованных формул / А. А. Ларионов, Е. А. Черкашин – Иркутск : Изд-во ИГУ, 2013. – 104 с.

3. <http://www.asmeurer.com>