

АЛГОРИТМИЧЕСКОЕ ОПИСАНИЕ МЕХАНИЗМА ЛОГИЧЕСКОГО ВЫВОДА ИНТЕЛЛЕКТУАЛЬНОГО СИМУЛЯТОРА ПРОМЫШЛЕННОГО РОБОТА

Частиков А.П.¹, Тотухов К.Е.¹

¹ФГБОУ ВПО «Кубанский государственный технологический университет», Краснодар, Россия (350072, Краснодар, ул. Московская, д. 2), e-mail: adm@kgtu.kuban.ru

Дано обоснование актуальности задачи создания интеллектуального компьютерного симулятора для повышения эффективности программирования промышленного робота. Рассмотрены характерные особенности создания механизма логического вывода для экспертной системы с двумя типами фактов – первичными и вторичными. Предложен оригинальный принцип предварительного построения списков правил, с помощью которых можно осуществлять перебор базы знаний в ширину. Учтены определённые трудности, связанные с применением данного механизма логического вывода на вычислительных машинах. Предложен программный подход, который позволил бы избежать данных затруднений. Детально описана структура и функциональный состав созданной дедуктивной машины. Наиболее сложные процедуры работы механизма логического вывода показаны в форме блок-схем алгоритмов на соответствующих рисунках. Полученное алгоритмическое обеспечение является достаточным для программной реализации данного механизма логического вывода на языке высокого уровня. Описан опыт экспериментального использования данного алгоритма. В заключение отмечена принципиальная возможность внедрения созданного механизма логического вывода в состав интеллектуальной информационной системы симуляции промышленного робота.

Ключевые слова: Алгоритм, Механизм логического вывода, Компьютерная симуляция, Промышленный робот, Экспертная система

ALGORITHMIC DESCRIPTION OF THE INFERENCE ENGINE OF INDUSTRIAL ROBOT'S INTELLIGENT SIMULATOR

Chastikov A.P.¹, Totukhov K.E.¹

¹Kuban State Technological University, Krasnodar, Russia, (350072, Krasnodar, Moskovskaya street, 2), e-mail: adm@kgtu.kuban.ru

The actuality of the task of creation of intelligent computer simulator for advancing of industrial robot's programming is justified. The main features of creation of the inference engine for the expert system with two types of facts (primary and secondary) are reviewed. There is offered an original principle of preliminary generation of lists of rules. With this list of rules the inference engine may perform the breadth-first search in a knowledge base. There are considered certain difficulties connected with implication of the created inference engine on computing machines. The programmatic approach which allows the avoidance of those difficulties is offered. The structure and the functions of the created inference engine are described in details. Its most complex working procedures are depicted on related figures as flowcharts. The obtained algorithmic description is enough for realization of this inference engine on a high-level programming language. The experience of practical implementation of this algorithm is described. As an assumption there is noticed the possibility of integration of the created inference engine in the intelligent information system of industrial robot's simulation.

Keywords: Algorithm, Inference engine, Computer simulation, Industrial robot, Expert system

Введение

Разработка компьютерных симуляторов роботов является актуальным направлением, призванным повысить эффективность создания управляющих программ, применяемых в робототехнике [3]. В работах [4, 6-9] приведены теоретические описания интеллектуального компьютерного симулятора промышленного робота. Разработана математическая модель робота, создана база знаний (БЗ) и построено дерево логического вывода.

Цель

В данной статье описывается создание механизма логического вывода (МЛВ) для экспертной системы с двумя типами фактов в составе интеллектуального симулятора промышленного робота.

Материалы и методы

В проведённых ранее исследованиях [6-7] было принято решение разделить факты экспертной системы на две категории: первичные и вторичные. Первичные факты формулируются на основе числовых параметров, характеризующих робота в процессе симуляции. Вторичные описывают суть совершаемых роботом действий с точки зрения технологического процесса. Поскольку интеллектуальная система создаётся с целью анализа технологических операций робота, очевидно, что МЛВ должен определять порядок перебора правил относительно вторичных фактов. При создании БЗ не отдавалось предпочтений каким-либо одним действиям робота по отношению к другим, как более приоритетным. Поэтому при переборе правил МЛВ стоит придерживаться стратегии просмотра в ширину.

Разработаем формальное описание МЛВ с указанными выше требуемыми свойствами, предварительно рассмотрев составные компоненты алгоритма. Пусть под списком 1 подразумевается совокупность фактов, на основании которых МЛВ определяет список 2 правил, которые должны быть просмотрены согласно стратегии перебора в ширину. В списке 1 могут храниться только вторичные факты. МЛВ должен найти в БЗ все правила, у которых каждому вторичному факту в ЕСЛИ-части (LHS-части) может быть найден тождественный факт в списке 1, и внести эти правила в список 2. Далее правила списка 2 просматриваются на предмет срабатывания. После завершения просмотра правил списка 2 совершается повторное построение списка 1. Для этого МЛВ должен занести все вторичные факты из ТО-частей (RHS-частей) правил списка 2 в предварительно очищенный список 1. На основании нового варианта списка 1 производится повторное построение списка 2 с последующим выполнением правил. Процедура должна повторяться до тех пор, пока МЛВ не обнаружит, что построить список 2 по списку 1 не удалось. Это возможно в том случае, если вторичные факты ТО-частей правил списка 2 не найдены в ЕСЛИ-частях ни одного из правил БЗ. Это позволяет считать, что достигнуты конечные точки логического вывода. Порядок расположения элементов в списках 1 и 2 не имеет принципиального значения.

Поскольку содержимое БЗ на протяжении процесса логического рассуждения не меняется, целесообразно предварительно получить все возможные варианты списка 2 и сохранить их в памяти в соответствующем порядке. Это позволит экономить ресурсы ЭВМ, избегая повторных построений списков 1 и 2. Множество возможных вариантов содержимого списка 2 назовём списками логического вывода.

Опишем технические процедуры, которые реализуют данный принцип.

- процедура 1. Ввод начального факта в список 1. Это действие должно быть совершено перед началом построения списков логического вывода. Цель – обозначить исходную точку начала логического вывода в базе знаний. В пустой список 1 будет введён факт «Схват в начальном положении»[6];
- процедура 2. Построение варианта списка 2 на основании текущего содержимого списка 1. Предварительно список 2 должен быть очищен от предыдущего содержимого;
- процедура 3. Сохранение текущего содержимого списка 2 в память, в область хранения всех списков логического вывода;
- процедура 4. Повторное построение списка 1 на основании списка 2. Предварительно список 1 должен быть очищен от предыдущего содержимого.

Принцип выполнения данных процедур следующий. Процедура 1 выполняется первой и только один раз. Далее выполняется процедура 2. Дадим подробное представление о процедуре 2 с помощью блок-схемы алгоритма, изображённой на рисунке 1.

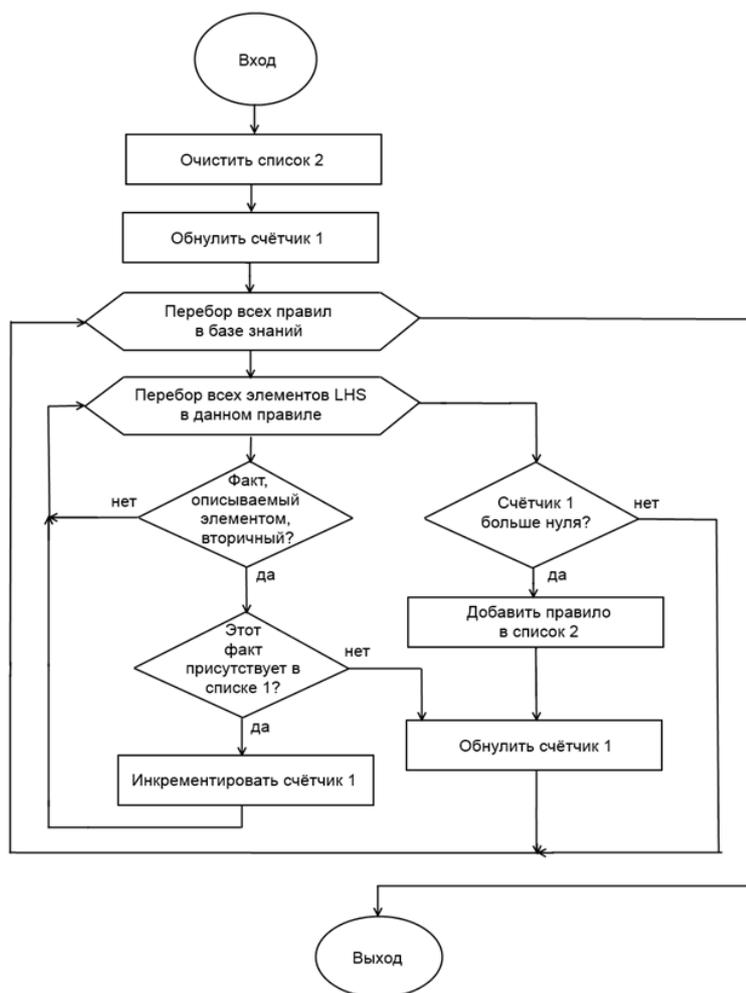


Рисунок 1 – Блок-схема процедуры 2

Процедура 2 с помощью двух циклов производит перебор всех правил БЗ и для каждого правила перебор всех элементов LHS. Во втором цикле используется целочисленная переменная «Счётчик 1». С помощью неё процедура контролирует количество совпадений

элементов LHS элементами списка 1. Если по завершению перебора элементов LHS будет найдено хотя бы одно совпадение с фактами списка 1, правило будет занесено в список 2.

Далее вызывается процедура 3, которая сохраняет список 2 в область хранения списков логического вывода с соблюдением порядка расположения. После этого вызывается процедура 4, призванная получить новый вариант списка 1 на основании текущего содержимого списка 2. Рассмотрим алгоритм процедуры 4 на рисунке 2.

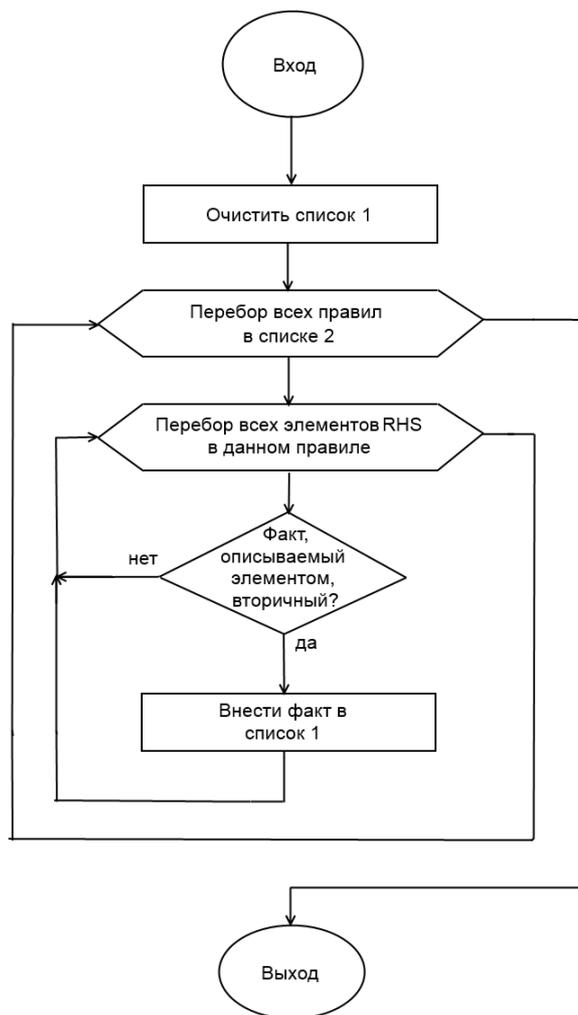


Рисунок 2 – Блок-схема процедуры 4

Процедура 4 производит перебор всех правил списка 2 и для каждого правила перебор всех элементов RHS. Учитывая, что процесс поиска в ширину производится относительно вторичных фактов экспертной системы, необходимо добавлять в список 1 только факты данного типа из RHS-частей правил списка 2.

По завершении выполнения процедуры 4, т.е., когда будет получен новый состав списка 1, вновь производится запуск процедуры 2, которая построит новый вариант списка 2 по списку 1. Если процедуре 2 не удастся построить новый список 2, это будет

интерпретировано как отсутствие фактов списка 1 в LHS-частях всех правил базы знаний и, следовательно, как причина прекратить построение списков логического вывода.

Описанные процедуры позволяют построить списки логического вывода в ширину. Таким образом, МЛВ получает совокупность списков логического вывода, с помощью которых он может теперь осуществлять рассуждения в БЗ. Блок-схема алгоритма процедуры, которая осуществляет перебор этих списков и выполнение сработавших в них правил, представлена на рисунке 3.

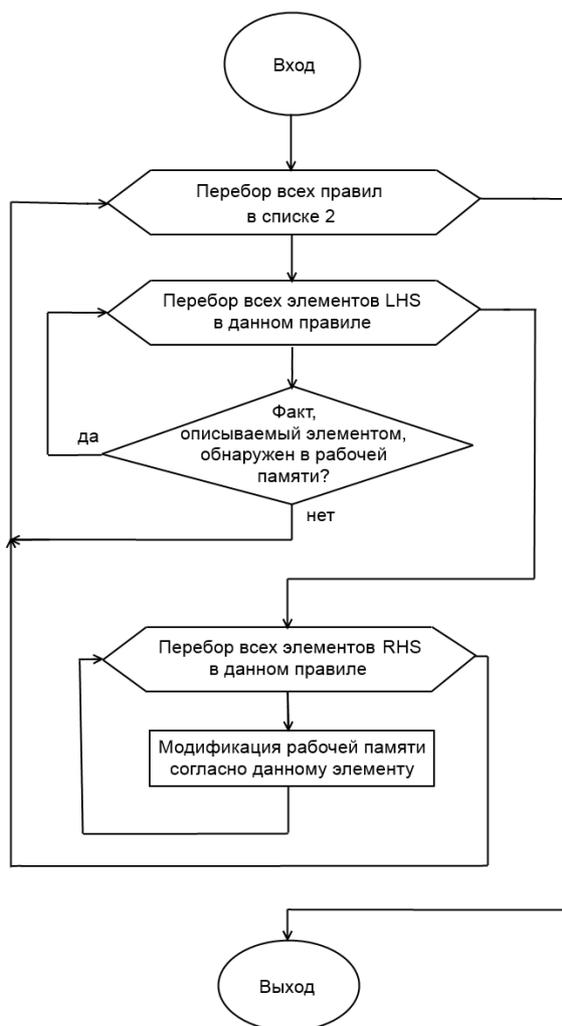


Рисунок 3 – Блок-схема процедуры выполнения правил

Процедура выполнения правил представляет собой три цикла, связанных сложными зависимостями друг с другом. Основной цикл производит перебор всех правил в списке 2. Под списком 2 подразумевается конкретный экземпляр совокупности списков логического вывода в ширину, извлечённый из области хранения в соответствующем порядке. Основным способом выбора списка является последовательный их перебор от первого к последнему, согласно их расположению в памяти[1-2, 5]. Для каждого правила в списке 2 процедура выполнения проверяет соответствие каждого элемента LHSодержимому рабочей памяти

экспертной системы. В случае полного совпадения правило считается сработавшим и выполняется третий цикл, который вносит в рабочую память экспертной системы элементы RHSданного правила. По завершению перебора всех списков логического вывода процедура должна запускаться вновь для первого списка. В результате МЛВ будет осуществлять циклический непрерывный перебор списков логического вывода и выполнение сработавших правил.Это создаст искомый эффект поиска решения в БЗ продукционного типа, характерный для экспертной системы.

Результаты и их обсуждение

Данный алгоритм был реализован программно на языке высокого уровняС# в среде программирования MSVisualStudio. Результатом выполнения программы должно было быть создание двумерного массива, содержащего списки, в каждом из которых хранится некоторое количество правил. Полученному программному прототипу была предложена БЗ, описанная в [6]. Результаты обработки БЗ,сохранённые в оперативной памяти, представлены на рисунке 4.

Lists2_Rules		Count = 12
[0]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 5
[1]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 4
[2]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 4
[3]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 3
[4]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 4
[5]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 5
[6]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 10
[7]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 10
[8]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 9
[9]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 9
[10]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 5
[11]	{WindowsFormsApplication1.Form1.List2}	
rules_ids		Count = 4
Raw View		

Рисунок 4 – Содержимое оперативной памяти программного прототипа

На рисунке показано размещение объекта *List2_Rules* в памяти. Этот объект представляет собой коллекцию из списков. На первой строке рисунка видно, что общее

количество сформированных списков равно двенадцати ($Count = 12$). В последующих строках отображены сами списки. Каждый список типа *rules_ids* представляет собой массив целочисленных идентификаторов, располагая которыми, МЛВ способен извлечь соответствующие правила из БЗ для обработки. Количество правил в каждом из списков различно. Оно изменяется от трёх и четырёх правил в четвёртом и пятом списках до девяти и десяти – в девятом и восьмом списках соответственно. В целом достигнутый результат можно считать удовлетворительным, поскольку расположение правил в списках соответствует необходимому согласно дереву логического вывода БЗ[7], за исключением некоторых уточнений.

Следует отметить, что принципы, описанные в данной статье, были частично применены для идентификации механизмов реализации операторов генетического алгоритма в экспертных системах продукционного типа[10]. Полученный результат также был признан положительным, поскольку позволил проводить исследования в продукционных системах с фактами двух типов.

Заключение

Разработанное авторами алгоритмическое обеспечение делает возможной программную реализацию МЛВ в интеллектуальной информационной системе симуляции промышленного робота. Подобная реализация полностью удовлетворяет требованиям экспертной системы, использующей факты двух типов в правилах. Благодаря этому, симулятор сможет самостоятельно делать заключения о технологическом процессе, осуществляемом роботом в процессе симуляции. Таким образом, создаваемая интеллектуальная информационная система симуляции позволит повысить эффективность разработки программ для промышленного робота.

Список литературы

1. Малыхина М.П., Бегман Ю.В. Нейросетевая экспертная система на основе прецедентов для решения проблем обслуживания абонентов сотовой сети / Известия высших учебных заведений. Северо-Кавказский регион. Серия: Технические науки. – 2009. - № 3. – С. 6-9
2. Частиков А.П., Алешин А.В., Частикова В.А. Выявление аномалий в базах знаний интеллектуальных систем // Труды Международной научно-технической конференции «Пятьдесят лет развития кибернетики». – СПб: Изд-во «Нестор», 1999.
3. Частиков А.П., Глушко С.П., Тотухов К.Е. Компьютерная симуляция программного управления виртуальным роботом: монография. – LAP LAMBERT AcademicPublishing, 2012. – 145 с.: ил.

4. Частиков А.П., Глушко С.П., Тотухов К.Е. Система тестирования и отладки управляющих программ для промышленного робота // Журнал «Перспективы науки». – 2010- № 9(11). –С. 47-51
5. Частиков А.П., Дедкова Т.Г., Алешин А.В. Системы искусственного интеллекта. От теории к практике. – Краснодар, 1998.
6. Частиков А.П., Тотухов К.Е. Создание базы знаний для интеллектуального анализа поведения виртуального робота. // Вопросы современной науки и практики. Университет им. В.И. Вернадского (ВАК) / Ассоциация «Объединённый университет им. В. И. Вернадского». – Тамбов, 2013. – С.76-81.
7. Частиков А.П., Тотухов К.Е., Урвачев П.М. Дерево логического вывода интеллектуальной системы функционирования виртуального робота // Современные проблемы науки и образования (ВАК). – 2013. - № 2; URL: www.science-education.ru/108-8976.
8. Частиков А.П., Тотухов К.Е., Урвачев П.М. Интеллектуальная диагностика состояния виртуального робота с программным управлением // Современные проблемы науки и образования (ВАК). – 2012. - № 6; URL: www.science-education.ru/106-7507.
9. Частиков А.П., Тотухов К.Е., Урвачев П.М. Теоретические основы интеллектуальной диагностики виртуального робота // Современные проблемы науки и образования (ВАК). – 2013. - № 1; URL: www.science-education.ru/107-8310.
10. Частикова В. А. Идентификация механизмов реализации операторов генетического алгоритма в экспертных системах производственного типа/В. А. Частикова//Научный журнал КубГАУ [Электронный ресурс]. – Краснодар: КубГАУ, 2012. -№ 75 (01). – Шифр Информрегистра: 0421200012/0024. – Режим доступа: <http://ej.kubagro.ru/2012/01/pdf/17.pdf>.

Рецензенты:

Видовский Л.А., д.т.н., профессор, зав.кафедрой ИСП, Кубанский государственный технический университет, г.Краснодар.

Ключко В.И., д.т.н., профессор, кафедра ИСП, Кубанский государственный технический университет, г.Краснодар.

Попов Ф.А., д.т.н., профессор, зам. директора по информационным технологиям, Бийский технологический институт (филиал) ГОУ ВПО «Алтайский государственный технический университет им. И.И.Ползунова», г. Бийск.