

СЕКЦИЯ: ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

© Е.В. Клыгина

Технологии с использованием искусственного интеллекта (ИИ) широко применяются во многих прикладных областях. Так, например, решения задач моделирования рассуждений человека, организации прямого или обратного логического вывода и других задач ИИ требуют иного подхода и программных средств, чем решения задач при помощи традиционных языков программирования. Для решения таких задач применяется логическое программирование. В широком смысле логическое программирование – это множество таких методов решения проблем, в которых используются приемы логического вывода для манипулирования знаниями, представленными в декларативной форме.

В логическом программировании можно выделить следующие направления: решение задач, требующих логического вывода, эвристическое программирование и создание решателей задач; представление знаний и разработка экспертных систем; создание новых языков логического программирования; создание вычислительных систем новых поколений [1].

Логическое программирование связывают с системами программирования, основанными на использовании специальных классов логических формул и специальных методов логического вывода. Одним из представителей языков логического программирования является Пролог. Программа на Прологе отлична от

программы в традиционном понимании, т. к. не содержит управляющих конструкций (условных операторов, операторов цикла и т. д.). Она представляет собой модель некоторого фрагмента предметной области, о котором идет речь в решаемой задаче. В программе на Прологе не надо задавать определенную последовательность действий, приводящих к решению задачи, необходимо описать задачу и основные правила для ее решения. Пролог-система с помощью встроенного механизма логического вывода сама найдет необходимое решение. Вследствие декларативного характера Пролога, программист начинает работу с хорошо структурированного описания задачи, а не с выработки сложных последовательностей команд.

Основы логического программирования изучаются в рамках курса «Интеллектуальные информационные системы» в институте математики, физики и информатики Тамбовского государственного университета им. Г.Р. Державина.

ЛИТЕРАТУРА

1. Адаменко А.Н., Кучуков А.М. Логическое программирование и Visual Prolog. СПб.: БХВ – Петербург, 2003. 992 с.
2. Марселлус Д. Программирование экспертных систем на Турбо Прологе. М.: Финансы и статистика, 1994. 256 с.

ТЕХНОЛОГИЯ ОБУЧЕНИЯ И КОНТРОЛЯ ЗНАНИЙ С ИСПОЛЬЗОВАНИЕМ ТЕСТОВОЙ ОБОЛОЧКИ «TESTIC»

© В.В. Кузнецов

Тестовая оболочка TestIC 1.1 является простейшим конвертером тестовых заданий, находящихся в текстовом файле. Главное преимущество ее в том, что она не предлагает никакого интерфейса для подготовки базы вопросов. Благодаря использованию только одного типа тестовых заданий (TestIC 1.1 может создавать тесты из вопросов закрытого типа с четырьмя вариантами ответов) предельно упрощен процесс предварительной разметки текстового файла с базой вопросов.

На самом деле, он не требует никакого специального языка разметки текста вопросов. Все вопросы и ответы предлагаются тестируемому в случайном порядке, что позволяет сильно разнообразить варианты представления вопросом теста испытуемым.

База вопросов может быть создана, например, в текстовом редакторе Блокнот. Обязательными, при создании базы вопросов, являются следующие требования:

- 1) кодировка Windows для файла базы вопросов;
- 2) расширение *.txt для файла базы вопросов, содержащего тестовые задания;
- 3) все тестовые задания базы вопросов – это задания закрытого типа. При этом на каждый вопрос предполагается ровно 4 варианта ответа.

С целью защиты информации от несанкционированного доступа оболочка используется вместе с кодировщиком, который кодирует как базу вопросов, так и файл с результатами ответов. Кодировщик TestIC'a создает в текущем каталоге файл TestIC.txt – это файл закодированной базы вопросов. Результаты тестирования будут видны как в окне самой программы, так и записаны в отдельный текстовый файл. Правда, в этом

файле они будут в закодированном виде. Для их декодирования необходимо воспользоваться еще раз кодировщиком TestIC'a.

Практика использования тестовой оболочки на занятиях показала, во-первых, что у студентов возросла мотивация подготовки к практическим занятиям. Во-вторых, у оболочки вдруг обнаружились недокументированные возможности обучения во время контроля. Это явление наблюдается при использовании тестов большого объема, достаточно полно охватывающих какой-либо раздел или тему, в ситуации, когда студентам в процессе ответов на вопросы теста разрешается пользоваться лекциями, справочной и учебной литературой.

АВТОМАТИЗИРОВАННАЯ ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

© Д.В. Лопатин, Е.С. Чиркин

В ходе компьютеризации физических экспериментов наиболее разумным является разделение программного обеспечения на два независимых блока. Первый блок отвечает за регистрацию, первичное преобразование и накопление информации. Программное обеспечение второго блока непосредственно обрабатывает и анализирует экспериментальные данные. Этим достигается независимость аппаратно-программного обеспечения.

Целью данной работы было создание программного продукта, предназначенного для автоматической и

полуавтоматической обработки экспериментальных результатов.

Представленный программный продукт позволяет осуществлять всю предварительную обработку экспериментальных данных:

- импорт данных из файлов различных форматов, в том числе и используемых популярными пакетами обработки информации;
- проверку входной информации и ее коррекцию;
- инструментальное построение, сравнение, масштабирование, слияние и анализ графиков.

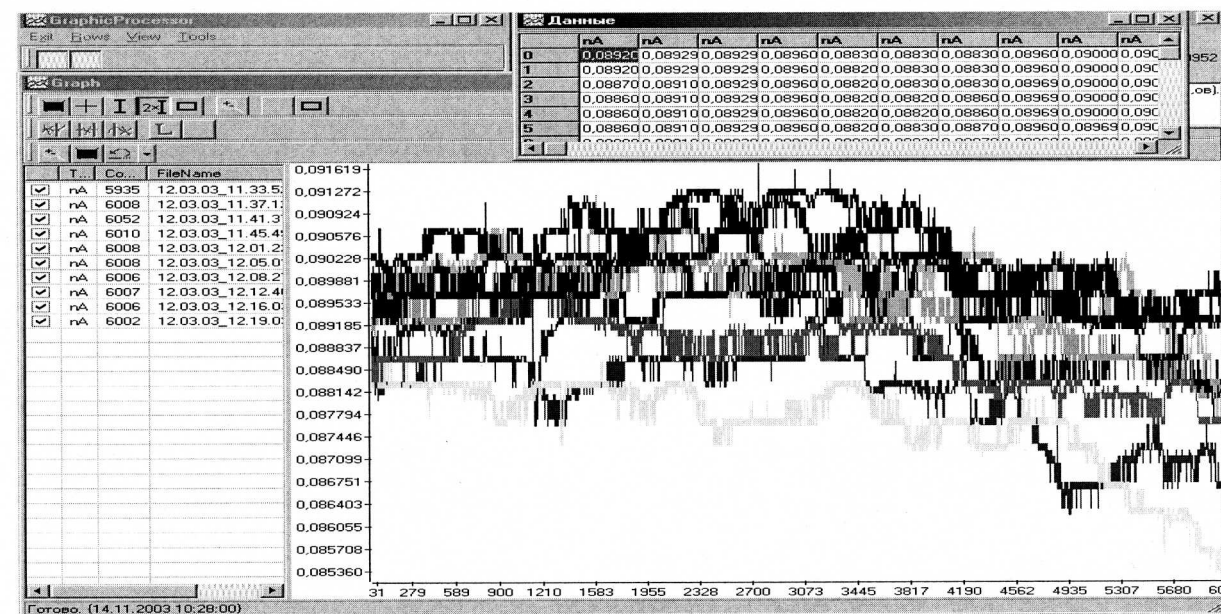


Рис. 1. Пакет автоматической обработки экспериментальных данных