Далее выполняется анализ признаков из АСП с целью оценки степени их информативности с точки зрения разделения формальных образов эталонов классов в многомерном признаковом пространстве принятия решений.

В дальнейшем интерес будут представлять только те признаки, значения которых одновременно и относительно слабо варьируют внутри каждого отдельного класса возле среднего арифметического, и демонстрируют неоднородность для всевозможных пар при соответствующем межклассовом сравнении.

Отметим, что за счет слабого варьирования значений соответствующего признака внутри класса обеспечивается компактность эталонов образов классов, а неоднородность при межклассовом сравнении обеспечивает разделение этих образов в пространстве решений.

Предложенный подход позволяет на основе соответствующих SQL-запросов осуществить формирование обучающей выборки и далее в автоматическом режиме на основе анализа данных обучающей выборки выполнить процедуру построения пространства принятия решений.

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПАРСИНГА HTML-ДОКУМЕНТА

Родченко Т. В.

ГрГУ им. Я. Купалы, Гродно, Беларусь, e-mail: taras1409@gmail.com

В процессе разработки веб-приложений часто приходиться сталкиваться с задачей парсинга html-документов, то есть преобразования “неудобных” данных в “удобные”. Поскольку число соответствующих html-документов, которые необходимо обработать достаточно большое, то и задача автоматизации процедуры парсинга является актуальной.

Парсер – это компьютерная программа, которая должна предусматривать сбор, анализ и преобразование информации к структурированному виду. Парсер должен предусматривать выполнение трех основных процедур:

1. Получение контента в исходном виде. Процедура предусматривает скачивание кода веб-страницы с целью извлечения необходимых данных. Для реализации этой процедуры используется библиотека HtmlAgilityPack.
2. Извлечение и преобразование данных. Происходит извлечение данных из полученного предыдущей процедурой кода страницы. Для извлечения данных используются паттерны (шаблоны), а потому для работы программы необходимо наличие соответствующей библиотеки паттернов, которую пользователь может пополнять и редактировать самостоятельно.
3. Генерация результата. Заключительная процедура обработки, которая связана с выводом и записью полученных предыдущей процедурой данных в требуемый формат. Результат либо записывается в базе данных, либо сохраняется в виде xml- или csv-файла.

Чтобы пропарсить одну страницу вручную разработчику необходимо потратить в среднем от полутора до двух часов на один html-документ. Использование предлагаемой системы парсинга позволяет сократить время обработки в 5-10 раз, в зависимости от

того существует ли нужный паттерн в библиотеке или же разработчику придется его создавать.

Основное время создания парсера занимает разработка алгоритма извлечения необходимой информации из документа. Код html-документа представлен в виде тэгов, которые могут иметь атрибуты, а так же могут быть контейнерами, то есть содержат в себе другие тэги или текст. Обрабатываемый html-документ содержит набор различных паттернов, каждый из которых представляет собой набор тэгов в определенной последовательности.

Суть предлагаемой системы заключается в сравнении кода html-документа с паттерном, который пользователь выбирает из соответствующей библиотеки. Такой подход позволяет при совпадении паттерна с частью кода html-документа извлечь необходимую информацию, отфильтровать и преобразовать ее к нужному виду.

Таким образом, данная система позволяет автоматизировать процесс парсинга html-документов, что в итоге повышает эффективность и сокращает расходы при решении подобного рода задач.

РЕАЛИЗАЦИЯ СЕТИ ПЕТРИ НА ОСНОВЕ СИСТЕМЫ КЛАССОВ С ЦЕЛЬЮ ЗАЩИТЫ ПРОГРАММ ОТ ВЗЛОМА

Рыженко Т. В.

Запорожский Национальный Технический Университет, Запорожье, Украина,
e-mail: dakuma@mail.ru

Большинство защит сводится к максимальному сокрытию и недопущения хакера к условному переходу, который проверяет авторизацию копии (например, серийный номер, хеш и т.д.). При этом обертка может состоять из самых изощренных анти-отладочных приемов, программа может быть запакована множеством упаковщиков – в середине всего этого находится один или несколько критических для взлома переходов. В статье [1] высказана идея о сокрытия условного перехода при помощи некоторой сети Петри. Там же можно найти пример такой сети (рис. 1):

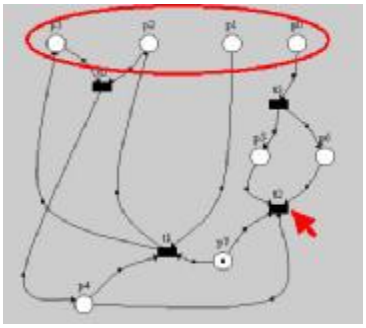

Рис. 1. Сеть Петри – мост отмеченный красной стрелкой срабатывает при определенной комбинации фишек в обведенных красным позиции.

Основная суть метода заключается в том, что срабатывание выделенного моста сети Петри связывается с переходом на защищаемый участок кода. Каждому мосту соответствует свой поток кода. Таким образом, хакер не сможет под отладчиком найти и пропатчить ключевой переход среди десятков подобных потоков. Переход на сообщение же об ошибке может быть выведено, например, по таймауту совсем в