МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Томский государственный университет
Горно-Алтайский государственный университет
Институт оптики атмосферы им. В.Е. Зуева СО РАН

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ИССЛЕДОВАНИИ СЛОЖНЫХ СТРУКТУР

МАТЕРИАЛЫ ДЕСЯТОЙ РОССИЙСКОЙ КОНФЕРЕНЦИИ С МЕЖДУНАРОДНЫМ УЧАСТИЕМ

Томск
Издательский Дом Томского государственного университета
2014

Секция 5. ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

О ВОЗРОЖДЕНИИ И РАЗВИТИИ ЛЯПАСА

Г.П. Агибалов

Национальный исследовательский Томский государственный университет, Томск, Россия
agibalov@isc.tsu.ru

ЛЯПАС, или Логический Язык для Представления Алгоритмов Синтеза [1], известный в западном научном мире как Russian Programming Language, родился в начале 1960-х гг. в Советском Союзе, а именно в Томском государственном университете (ТГУ) под руководством А.Д. Закревского, был реализован на всех отечественных ЭВМ и ряде американских компьютеров, стал базовым языком программирования в системах автоматизированного проектирования дискретных управляющих устройств в министерствах электронной и радиопромышленности СССР, получил статус международного языка программирования в странах соцлагеря и с уничтожением последнего и ликвидацией производства отечественной вычислительной техники вышел из употребления в 1990-х гг.

В настоящее время, в связи с нарастающей угрозой информационной безопасности России, проистекающей от использования в компьютерных системах (КС) управления сложными объектами и технологическими процессами (в энергетике, промышленности, на транспорте) недоверенного программно-аппаратного обеспечения (ПАО), заимствованного у потенциального противника и содержащего недокументированные закладки, через которые возможна утечка одной информации и навязывание другой, в том числе разрушительной, и обнаружить которые, даже в свободном ПАО, часто бывает невозможно, в особенности, когда оно «запутано» средствами обфускации, возрождение ЛЯПАСа с целью создания на его базе собственного (доверенного) ПАО для автоматического синтеза безопасных КС логического управления стало для России жизненно важной научно-технической проблемой [2].

Кафедра защиты информации и криптографии ТГУ занимается данной проблемой с 2010 г. За это время ею проделано следующее [3]: проведена ревизия ЛЯПАСа, выразившаяся в «осовременивании» его алфавита и некоторых его операций, отразившемся в его названии по имени vЛЯПАС (от reVised LYaPAS); разработан и эксплуатируется в научных исследованиях и учебном процессе компилятор vЛЯПАСа в язык Ассемблера под ОС Linux; построено криптографическое расширение vЛЯПАСа – язык ЛЯПАС-Т (от exTended LYAPAS) с арифметикой длинных чисел, логикой длинных булевых векторов и операциями криптографического назначения; разработаны проекты процессора для аппаратной реализации ЛЯПАСа-Т и препроцессора для трансляции программ на ЛЯПАСе-Т в исполняемый код процессора; построена и промоделирована на компьютере принципиальная схема процессора на базе ПЛИС для базового подмножества vЛЯПАСа; разработаны и исследованы в компьютерном эксперименте программы на vЛЯПАСе для ряда алгоритмов криптографической защиты управляющей информации – AES, ГОСТ, ElGamal, KASUMI, PUA и др.

В дальнейшие планы возрождения и развития ЛЯПАСа входят следующие мероприятия:

1) *разработка и исследование математического и программного обеспечения* – математической модели безопасной КС логического управления; компилятора ЛЯПАСа-Т; безопасной ОС на vЛЯПАСе для запуска программ на ЛЯПАСе-Т; алгоритмов и программ на ЛЯПАСе-Т для логического управления, логического синтеза управляющих автоматов и криптографической защиты управляющей информации;

2) *исследование и разработка аппаратного обеспечения* – процессора ЛЯПАСа-Т (его архитектуры, исполняемого кода, алгоритма функционирования, логической схемы); препроцессора для него; архитектуры ЛЯПАС-машины – специализированного компьютера с процессором ЛЯПАСа-Т; реактивной ОС на vЛЯПАСе, управляющей взаимодействием процессора ЛЯПАС-машины с аппаратными модулями системы управления;

3) *реализация процессора ЛЯПАСа-Т на базе ПЛИС и* (*или*) *заказных интегральных схемах* – описание процессора на VHDL; его отладка путём компьютерного моделирования по компонентам и в целом; автоматический синтез и компьютерное моделирование логической схемы процессора;

4) *издательская деятельность*: издание периодического сборника алгоритмов на ЛЯПАСе-Т;

5) *производство ЛЯПАС-машины*: опытно-конструкторские работы; изготовление опытного образца.

Литература

1. *Торопов Н.Р.* Язык программирования ЛЯПАС // Прикладная дискретная математика. 2009. № 2(4). С. 9–25.
2. *Агибалов Г.П.* К возрождению Русского языка программирования // Прикладная дискретная математика. 2012. № 3(17). С. 77–84.
3. *Агибалов Г.П.*, *Панкратова И.А.*, *Липский В.Б.* О криптографическом расширении и его реализации для Русского языка программирования // Прикладная дискретная математика. 2013. № 3 (21). С. 93–104.

МЕТОД КЛАССИФИКАЦИИ ТЕКСТОВ ХУДОЖЕСТВЕННОЙ ЛИТЕРАТУРЫ НА ОСНОВЕ R-МЕРЫ

М.Ф. Ашуров[1], *В.В. Поддубный*[2]

Национальный исследовательский Томский государственный университет, Томск, Россия
[1]therevenge.amf@gmail.com, [2]vvpoddubny@gmail.com

Рассматривается задача потоковой классификации текстов естественного языка на основе использования *R*-меры. В потоковых методах классификации текст *X* представляется как последовательность (поток) из *n* элементов (символов, слов, словосочетаний и т.п.) [1]. Задача классификации текстов возникает, например, при поиске жанровых, авторских и др. стилевых особенностей текстов. При этом предполагается наличие определённого количества текстов каждого класса (жанрового, авторского или др.) для обучения классификатора. Обучение состоит в объединении текстов каждого класса в супертексты. Отнесение испытуемого текста к какому-либо классу производится по максимуму меры его близости к супертексту того или иного класса. Усечённая *R*-мера близости [1] учитывает все возможные повторения всех подстрок длин от k_1 до k_2 испытуемого текста длины n в супертексте (в отличие от не усечённой, для которой $k_1 = 1$, $k_2 = n$):

$$R(X \mid S) = r(X \mid S) / N,\ r\left(X \mid S\right) = \sum_{k=k_1}^{k_2} c_k\left(X \mid S\right),\ N = \left(2(n+1) - (k_1 + k_2)\right)\left(k_2 - k_1 + 1\right) / 2,$$

$$c_k\left(X \mid S\right) = \sum_{i=k}^{n} c\left(x_{i-k+1} \ldots x_n \mid S\right),\ c\left(x_{i-k+1} \ldots x_n \mid S\right) = \begin{cases} 1, x_{i-k+1} \ldots x_n \subset S \\ 0, x_{i-k+1} \ldots x_n \not\subset S \end{cases},$$

где *X* – испытуемый текст, *S* – супертекст исследуемого класса, *k* – длина подстроки поиска, *N* – число подстрок. Целью работы было исследование качества потоковой классификации по этой мере.

Анализ классификатора, основанного на усечённой *R*-мере, проводился нами на текстах русской художественной прозы 19 века. Всего было использовано 9 авторов, представлявших авторские классы, около 100 произведений которых были выбраны в качестве «обучающей» выборки. Образованные этими произведениями супертексты каждого класса были уравнены по объему. Тексты тестовой (контрольной) выборки каждого класса были составлены тоже равного объема, порядка 100 тыс. символов каждый, и в равном количестве, при этом некоторые из них были получены из текстов обучающей выборки, но большая часть – из других произведений, не участвовавших в обучении.

Для проведения тестирования качества потокового метода классификации был спроектирован и реализован на языке C# программный модуль, позволяющий проводить классификацию текстов на основе *R*-меры. Проблема выбора параметров k_1 и k_2 решалась с учётом особенностей текстов на естественных языках. Минимальная длина подстроки $k_1 = 10$ символам, так как более короткие подстроки начинают совпадать с большинством слов русской речи, обычно используемых всеми авторами, что не позволяет выделить стилевые особенности разных авторов. Максимальная длина подстроки $k_2 = 45$ символам – такая длина подстроки может включать в себе даже словосочетание из 3 или 4 слов. Использование большей длины подстроки видится малополезным, так как повтор одной и той же длинной фразы у разных авторов представляется крайне маловероятным (проблема плагиата полностью исключалась из рассмотрения). Качество классификации по каждому классу оценивалось по текстам контрольной выборки *F*-мерой – средним гармоническим между полнотой (долей текстов, правильно приписываемых классу из всех текстов этого класса) и точностью (долей текстов, правильно приписываемых классу из всех текстов, приписываемых этому классу) [1]. Среднее значение *F*-меры по всем классам принималось за оценку качества потоковой классификации в целом. Улучшить качество классификации можно исключением подстрок, которые присутствуют (или отсутствуют) во всех классах одновременно. Такой отфильтрованный набор подстрок (признаков авторского стиля текстов) был бы,