Использование программы позволило проводить занятия в компьютерном кабинете. Дальнейшее использование программы мы видим в её кооперации с другими программами (их поиск и/или разработка) так, чтобы подобное взаимодействие способствовало улучшению учебного процесса.

Использование новых информационных технологий позволяет заменить многие традиционные средства обучения. В ряде случаев такая замена оказывается эффективной, так как позволяет поддерживать у учащихся интерес к изучаемому предмету, создавать информационную обстановку, стимулирующую любопытство.

Использование современных информационных технологий способствует повышению эффективности, качества процесса обучения, активности познавательной деятельности, углублению межпредметных связей, увеличению объёма и оптимизации поиска нужной информации, формированию информационной культуры, умений осуществлять обработку информации и экспериментально-исследовательскую деятельность, подготовке информационно грамотной личности [3].

Список цитируемых источников

1. Использование компьютера как инструмента образовательного процесса. URL: http://www.rusedu.info/Article598.html (дата обращения: 10.02.15).
2. Использование программ администрирования в процессе обучения. URL: http://inzhenery.su/slovar/obrazovanie/ispolzovanie-programm-administrirovaniya-v-processe-obucheniya.html (дата обращения: 12.02.15).
3. Внедрение информационных технологий в процесс обучения математике. URL: http://www.uchportal.ru/board/2-1-0-36 (дата обращения: 13.02.15).

УДК 517.05.512

Е. В. Ставер
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», Минск

СТАТИСТИЧЕСКАЯ ПРОВЕРКА СЛУЧАЙНОСТИ ДВОИЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ. АНАЛИЗ ЧАСТОТНОГО ПОБИТОВОГО И ЧАСТОТНОГО БЛОЧНОГО ТЕСТОВ NIST DRAFT SP 800-90B

Описана методика тестирования последовательностей с применением частотного блочного и частотного побитового тестов.

A method for testing sequences with the frequency of the frequency block and bit-tests.

Введение. При работе с криптографией без генераторов случайных чисел не обойтись. Одним из возможных применений таких генераторов является генерация ключей. Если последовательность случайных чисел предсказуема, то даже самый стойкий алгоритм шифрования, в котором данная последовательность будет использоваться, оказывается уязвим, например, резко уменьшается пространство возможных ключей, которые необходимо «перебрать» злоумышленнику для получения некоторой информации, с помощью которой он сможет «взломать» всю систему.

Национальный институт стандартов и технологий США (NIST) разработал набор тестов для оценки случайности последовательности чисел. О них и пойдёт речь в данной статье. Будут рассмотрены частотный блочный и частотный побитовый тесты.

Целью данной статьи является исследование математических методов для тестирования последовательностей с применением частотного блочного и частотного побитового тестов.

Объект исследования — случайные последовательности и математические методы для их тестирования.

Основная часть. Пакет статистических тестов разработан Лабораторией информационных технологий NIST [1]. В его состав входят статистические тесты, цель которых — определение меры случайности двоичных последовательностей генераторов случайных чисел. Тесты основаны на различных статистических свойствах случайных последовательностей.

Под генерацией случайных чисел подразумевается получение последовательности из двоичных знаков 0 и 1. Генераторы случайных чисел бывают случайные (физические датчики случайных чисел) и псевдослучайные (программные генераторы случайных чисел).

Первые принимают на вход некий случайный бесконечный процесс, а на выходе дают бесконечную последовательность 0 и 1. Вторые представляют собой заданную программистом детерминированную функцию, которая инициализируется и на выходе выдаёт последовательность 0 и 1.

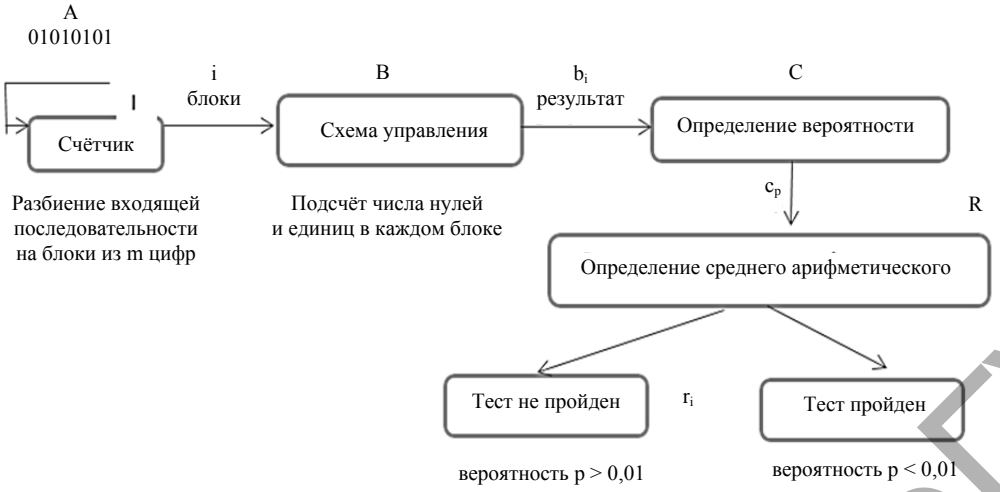

Рисунок 1 — Абстрактный автомат частотного блочного теста

Частотный блочный тест. Суть — определение доли единиц внутри блока длиной *m* бит. Цель — установить, что частота повторения единиц в блоке длиной *m* бит приблизительно равна *m* / 2 [2]. Вычисленное в ходе теста значение вероятности *p* должно быть не меньше 0,01. Если $p < 0{,}01$, последовательность не носит случайный характер. Если принять $m = 1$, данный тест переходит в частотный побитовый. Рекомендуемой длиной тестируемых последовательностей, согласно NIST, являются битовые строки по 1 000 000 бит. Всего для принятия одной из гипотез тестируется 100 последовательностей (рисунок 1).

Автомат Мили — конечный автомат, где выходные сигналы являются функциями текущего состояния и входного сигнала, в отличие от автомата Мура, в котором выходные сигналы — функции только состояния.

Абстрактный автомат получим, если укажем алфавит A, B, C, I, R и программу P как совокупность команд вида $i_i,a_j \rightarrow i_x,b_y$.

В нашем случае:

$$A=\{a_1,a_2\},\ B=\{b_1,b_2\},\ C=\{c_1,c_2\},\ I=\{i_1,i_2,\ldots,i_n\},\ R=\{r_1,r_2\}\ ,$$

где a_1 — пришёл 0;
a_2 — пришла 1;
b_1 — инкрементировать счётчик нулей;
b_2 — инкрементировать счётчик единиц;
c_1 — |количество нулей / количество единиц − 1| > 0,01;
c_2 — |количество нулей / количество единиц − 1| < 0,01;
i_1 — поместить число a в очередь n_1;
i_2 — поместить число a в очередь n_2;
i_n — поместить число a в очередь n_m;
r_1 — $|\sum c_i / i| > 0{,}01$;
r_2 — $|\sum c_i / i| < 0{,}01$.

δ: $AxI \rightarrow CxB \rightarrow CxR=\{a_1i_1 \rightarrow b_1c_1,\ a_1i_2 \rightarrow b_1c_1,\ a_1i_n \rightarrow b_1c_1,\ a_2i_1 \rightarrow b_2c_2,\ a_2i_2 \rightarrow b_2c_2,\ a_2i_n \rightarrow b_2c_2,\ b_2c_2 \rightarrow c_2r_2,\ b_2c_2 \rightarrow c_2r_1,\ b_1c_1 \rightarrow c_1r_1,\ b_1c_1 \rightarrow c_1r_2\}$.

Пример из методики NIST STS: для последовательности из 800 бит с 5%-м интервалом границы для частотного теста выбираются:

$$R = 400 + 1{,}96 / 2 \cdot SQRT(800) = 400 + 1{,}96 / 2 \cdot SQRT(800) = 400 + 1{,}96 / 2 \cdot 28{,}8 =$$
$$= 400 + 0{,}98 \cdot 28{,}8 = 400 + 28{,}224 = [373{,}427].$$

Частотный побитовый тест. Принимаем каждую единицу за плюс единицу, а каждый ноль за минус единицу и считаем сумму по всей последовательности. Есть мнение, что распределение количества успешных проходов в серии экспериментов, где в каждом эксперименте возможен успех или неуспех с заданной вероятностью, имеет биномиальное распределение (рисунок 2).

Абстрактный автомат получим, если укажем алфавит A, B и C и программу P как совокупность команд вида $b_i,a_j \rightarrow b_e,c_p$.

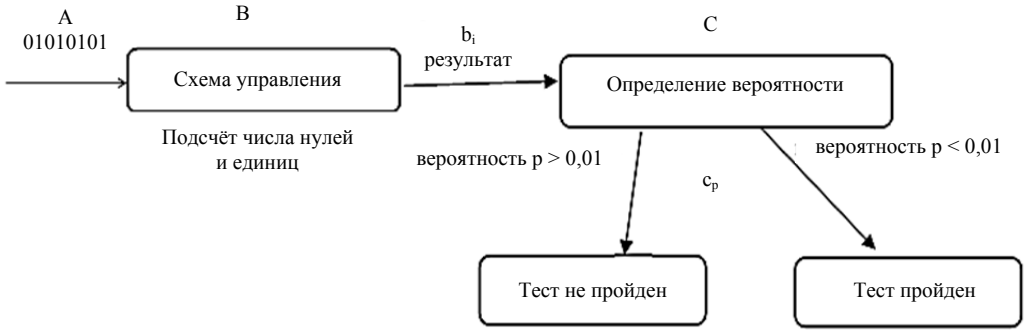

Рисунок 2 — Абстрактный автомат частотного побитового теста

В нашем случае:

$$A=\{a_1,a_2\},\ B=\{b_1,b_2\},\ C=\{c_1,c_2\}\ ,$$

где a_1 — пришёл 0;
a_2 — пришла 1;
b_1 — инкрементировать счётчик нулей;
b_2 — инкрементировать счётчик единиц;
c_1 — |количество нулей / количество единиц – 1| > 0,01;
c_2 — |количество нулей / количество единиц – 1| < 0,01.

$$\delta: BxA \rightarrow BxC=\{b_1a_1 \rightarrow b_1c_1,\ b_2a_2 \rightarrow b_2c_2,\ b_1a_1 \rightarrow b_1c_2,\ b_2a_2 \rightarrow b_2c_1\}\ .$$

Заключение. Побитовый частотный тест гарантирует, что существует примерно одинаковое количество нулей и единиц. Этот тест применяется в виде одностороннего критерия хи-квадрат.

Можно говорить о том, что дефектом частотного побитового теста является наличие большого количества нулей в последовательности, а дефектом частотного блочного теста — локализованные отклонения частоты появления единиц в блоке от идеального значения $^1/_2$.

Список цитируемых источников

1. NIST SP 800-90A. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. URL: http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf (date of access: 13.09.2015).
2. Ibid.

УДК 53(07)

Д. Р. Читая, О. И. Быстров
Государственное учреждение образования «Средняя школа № 187 г. Минска», Минск

ОПЕРАЦИОННАЯ СИСТЕМА SYSTEM — СИНХРОНИЗАЦИЯ С БУДУЩИМ

В статье ставится задача рассмотреть псевдо-операционную систему System TegrOS. В результате анализа автор доказывает эффективность использования функций данной системы, в том числе и для людей с ограниченными возможностями. Искусственный интеллект системы сможет контролировать систему защиты, автоматически находить и исправлять системные ошибки, перенастраивать и обновлять операционную систему (далее — ОС).

The article seeks to examine pseudo operating system TegrOS. As a result of analysis the author proves efficiency of use of the functions of this system, including for people with disabilities. Artificial intelligence systems will be able to control the security system, automatically find and fix system errors, reconfigure and upgrade the operating system (onward OS).

Введение. Мы живём в мире с растущими потребностями, которые приводят к совершенствованию и прогрессу. Развитие происходит во всех сферах жизнедеятельности человека. Не остаётся без внимания и модернизация ОС.