сификации происходит после подачи всех образов сканируемого файла на нейронную сеть. Отличительной особенностью алгоритма является способность НИД обнаруживать неизвестные вредоносные программы.
2. Получено приближенное выражение для оценки вероятности обнаружения вредоносной программы искусственной иммунной системой. Показано, что с увеличением количества детекторов увеличивается вероятность обнаружения. Предложена приближенная оценка количества детекторов для заданной вероятности обнаружения вредоносной программы.
3. Проведены эксперименты по тестированию нейросетевой искусственной иммунной системы. Они показали способность нейросетевых иммунных детекторов обнаруживать разнотипные неизвестные вредоносные программы. В отличие от известных антивирусных программ нейросетевая искусственная иммунная система обнаруживает в среднем в 1,5 раза больше неизвестных вредоносных программ.
4. Приведены теоретическая и экспериментальная оценки вероятности обнаружения вредоносной программы в зависимости от количества детекторов.

Разработанная система может быть использована при построении как новых систем защиты компьютеров от вредоносных программ, так и в дополнении к уже имеющимся средствам.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Киберпреступность // Центр исследования компьютерной преступности [Электронный ресурс]. – 2007. – Режим доступа: http://www.crime-research.ru/news/05.09.2007/3793/. – Дата доступа: 27.11.2007.
2. Пресс-Центр // Антивирус ВирусБлокАда [Электронный ресурс]. – 2005. – Режим доступа: http://www.anti-virus.by/press/viruses/1485.html. – Дата доступа: 25.08.2007.
3. Касперский, Е. Компьютерное зловредство / Е. Касперский. – СПб.: Питер, 2007. – 208 с.
4. Касперский, К. Записки исследователя компьютерных вирусов / К. Касперский. – СПб.: Питер, 2006. – 316 с.
5. Куприянов, А.И. Основы защиты информации / А.И. Куприянов, А.В. Сахаров. – М.: Академия, 2006. – 256 с.
6. Зайцев, О.В. Rootkits, spyware/adware, keyloggers & backdoors: Обнаружение и защита / О.В. Зайцев. – СПб.: BHV-Санкт-Перербург, 2006. – 304 с.
7. Проактивность как средство борьбы с вирусами // Интернет-безопасность [Электронный ресурс]. – 1996. – Режим доступа: http://www.viruslist.com/ru/analysis?pubid=189544544. - Дата доступа: 15.05.2008.
8. Безобразов, С.В. Нейросетевая искусственная иммунная система для обнаружения вредоносных программ: принципы построения / С.В. Безобразов, В.А. Головко // Вестник БрГТУ. Физика, математика, информатика. – 2009.
9. Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – М.: Вильямс, 2005. – 1424 с.
10. Дасгупта, Д. Искусственные иммунные системы и их применение / Д. Дасгупта; под ред. Д. Дасгупта. – М.: Физматлит, 2006. – 344 с.
11. Головко, В.А. Нейронные сети: обучение, организация, применение / В.А. Головко // Нейрокомпьютеры и их применение : учеб. пособие / В.А. Головко. – М., 2001 – 256 с.
12. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1104 с.
13. Яхъяева, Г.Э. Нечеткие множества и нейронные сети / Г.Э. Яхъяева. – М.: Бином. ЛЗ, 2008. – 316 с.

[illegible] 11.11.10

BEZOBRAZOV S.V., GOLOVKO V.A. The Neuronet Immune System for Malware Detection: the Principles of Construction

In this paper we propose the principles of the neural network immune system functioning for detection of unknown malware. Research results are submitted.

[illegible] 004.5;621.38

Бутов А.А.

[illegible]

Введение. Традиционные способы описания геометрических объектов и, в частности, многоугольников основаны на использовании методов вычислительной геометрии [1] и имеют практическое применение, например, при автоматизированном проектировании топологии интегральных схем [2, 3]. Однако в последнее время появились альтернативные способы описания многоугольников, основанные на использовании булевых формул [4, 5].

В настоящей работе предлагается способ решения одной из задач, лежащих в основе изложенного в работе [5] метода построения канонической булевой формулы. Тем самым указанный метод может быть легко доведен до формы алгоритма и, далее, переведен в форму программ на каком-либо языке программирования.

1. Основные определения, постановка задачи. Многоугольник, расположенный на плоскости, задается своей *границей* – замкнутой не пересекающейся ломаной линией, состоящей из отрезков прямых или *сторон* многоугольника. Эту границу можно определить последовательностью *угловых точек* или *вершин* многоугольника, получаемых при обходе его по границе справа: $p_1, p_2, \ldots, p_n$ (рис. 1).

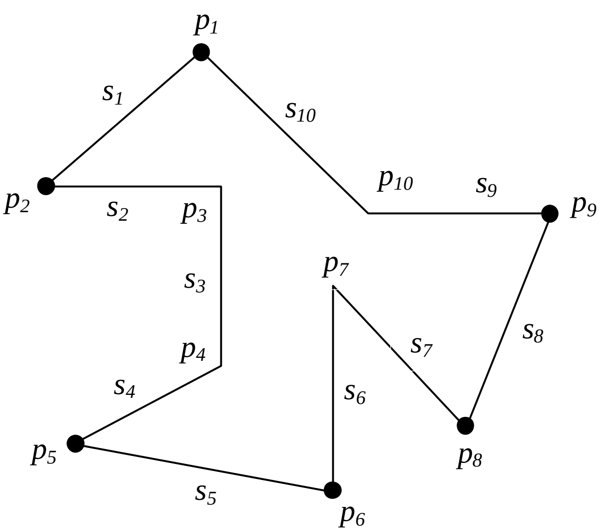

Рис. 1. Угловые точки и стороны многоугольника

***Бутов А.А.**, к.т.н., доцент кафедры экономической каибернетики Белорусского государственного университета информатики и радиоэлектроники.*
Беларусь, БГУИР, 220013, г. Минск, ул. П. Бровки, 6.

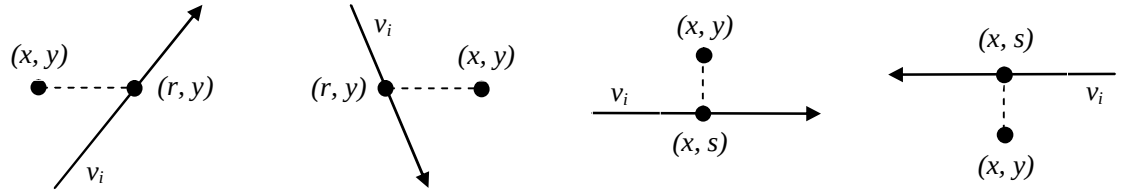

Рис. 2. Варианты ориентации точки относительно прямой

Так как каждая пара соседних угловых точек ограничивает соответствующую сторону многоугольника, то его границу можно задать также последовательностью сторон многоугольника: s_1, s_2, …, s_n, где $s_1 = (p_1, p_2)$, $s_2 = (p_2, p_3)$, …, $s_n = (p_n, p_1)$.

Вершина p_1, которая служит начальной точкой для последовательного обозначения отрезков, образующих границу многоугольника, называется *начальной*. В качестве начальной будем выбирать вершину, наиболее удаленную от координатной оси X (если таких вершин несколько, то среди них выбирается вершина, наиболее удаленная от координатной оси Y).

Вершина многоугольника называется *крайней*, если через нее можно провести прямую, не пересекающуюся ни с одним из отрезков границы за исключением двух соседних отрезков, на стыке которых эта вершина располагается. На рис. 1 таких вершин шесть, и они отмечены жирными кружками. Начальная вершина p_1, как следует из указанного выше способа ее выбора, всегда принадлежит множеству крайних вершин многоугольника.

Каждой стороне s_i многоугольника поставим в соответствие *ориентированную прямую* v_i, содержащую точки p_i и p_{i+1}. Положим, что она ориентирована от p_i к p_{i+1}.

Рассмотрим некоторую произвольную точку плоскости p, заданную парой декартовых координат (x, y). Будем считать, что точка p *расположена слева от прямой* v_i, если она принадлежит полуплоскости, расположенной слева от ориентированной прямой v_i, или лежит на прямой v_i. Все возможные варианты левостороннего расположения точки p относительно ориентированной прямой v_i представлены на рис. 2 (последние два варианта соответствуют случаю, когда прямая v_i параллельна координатной оси X).

Как и в работе [5], будем в дальнейшем обозначать отрезки ломаной буквами a, b, c, … , границу многоугольника как abc…, а полуплоскости, расположенные слева от соответствующих ориентированных прямых – буквами A, B, C, … (считая, что каждая из этих полуплоскостей включает в себя еще и все точки порождающей ее ориентированной прямой). Введем также предикаты a, b, c, … для описания положения некоторой точки p на плоскости, полагая, что $a(p) = 1$, если и только если $p \in A$.

Основываясь на таких предикатных переменных, в работе [5] описан метод построения канонической булевой формулы F, представляющей многоугольник и обладающей следующим свойством: если выполнить подстановку предикатных координат произвольной точки плоскости, то формула F примет значение 1 в случае, когда точка принадлежит данному многоугольнику, и значение 0 – в противном случае.

Необходимо, однако, отметить, что в этом методе построения формулы F многократно используется операция поиска множества крайних вершин (сначала для исходного многоугольника, затем – для более простых многоугольников, получаемых в процессе формирования формулы F).

Поскольку указанная задача поиска множества крайних вершин многоугольника является нетривиальной, то в настоящей работе эта задача рассматривается подробно и предлагается способ ее решения.

2. Каноническая булева формула многоугольника. Метод нахождения формулы для любого многоугольника, изложенный в работе [5], проиллюстрируем на примере многоугольника, изображенного на рис. 3. Метод включает в себя следующие действия.

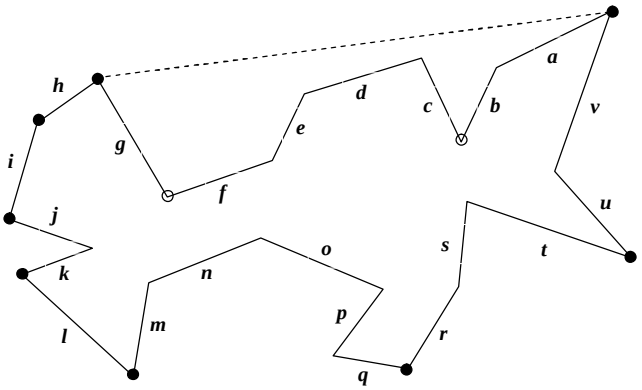

Рис. 3. Многоугольник, для которого строится булева формула

1. В многоугольнике находятся все крайние вершины (на рисунке они отмечены жирными кружками).
2. Граница многоугольника ***abcdefghijklmnopqrstuv*** разбивается крайними вершинами на фрагменты *первого ранга*, обозначаемые последовательностями образующих их отрезков: ***abcdefg***, ***h***, ***i***, ***jk***, ***l***, ***mnopq***, ***rst***, ***uv***.
3. Искомая формула F строится как конъюнкция формул, описывающих фрагменты первого ранга:

$$F = F(\boldsymbol{abcdefg}) \wedge F(\boldsymbol{h}) \wedge F(\boldsymbol{i}) \wedge F(\boldsymbol{jk}) \wedge F(\boldsymbol{l}) \wedge F(\boldsymbol{mnopq}) \wedge F(\boldsymbol{rst}) \wedge F(\boldsymbol{uv}).$$

После замены фрагментов, содержащих лишь один отрезок, символом соответствующего предиката, получаем:

$$F = F(\boldsymbol{abcdefg}) \wedge h \wedge i \wedge F(\boldsymbol{jk}) \wedge l \wedge F(\boldsymbol{mnopq}) \wedge F(\boldsymbol{rst}) \wedge F(\boldsymbol{uv}).$$

4. Фрагменты первого ранга, содержащие более одного отрезка, рассекаются на фрагменты второго ранга следующей процедурой, иллюстрируемой на примере фрагмента ***abcdefg***:

а) две концевые точки фрагмента соединяются условным отрезком (показанным на рисунке пунктирной линией), в результате чего получается многоугольник, называемый *замкнутым фрагментом*;

б) в замкнутом фрагменте отыскиваются крайние вершины (к ним не относятся концевые точки). Они отмечены на рисунке простыми кружками и делят рассматриваемый фрагмент на фрагменты второго ранга ***ab***, ***cdef*** и ***g***. Дизъюнкция формул этих фрагментов представляет формулу фрагмента в целом:

$$F(\boldsymbol{abcdefg}) = F(\boldsymbol{ab}) \vee F(\boldsymbol{cdef}) \vee F(\boldsymbol{g}).$$

Аналогично делятся фрагменты первого ранга ***jk***, ***mnopq***, ***rst*** и ***uv***. В результате этого шага получаем:

$$F = (F(\boldsymbol{ab}) \vee F(\boldsymbol{cdef}) \vee g) \wedge h \wedge i \wedge (j \vee k) \wedge l \wedge (m \vee n \vee o \vee F(\boldsymbol{pq})) \wedge (F(\boldsymbol{rs}) \vee t) \wedge (u \vee v).$$

Полученные фрагменты второго ранга, содержащие более одного отрезка, разбиваются на фрагменты *третьего ранга* и т.д.

5. Эта итеративная процедура повторяется, пока все фрагменты не разобьются на отдельные отрезки. При этом все фрагменты нечетного ранга являются вогнутыми, а фрагменты четного ранга – выпуклыми. Каждый раз формула фрагмента нечетного ранга представляется дизъюнкцией формул выпуклых фрагментов, получаемых при разбиении, а формула фрагмента четного ранга – конъюнкцией формул соответствующих вогнутым фрагментам.

Для рассматриваемого многоугольника получаем в конечном результате следующую каноническую булеву формулу:

$F = (ab \lor cd(e \lor f) \lor g)\ hi\ (j \lor k)\ l\ (m \lor n \lor o \lor pq) \land (rs \lor t) \land (u \lor v)$.

3. Поиск множества крайних вершин многоугольника. Суть предлагаемого способа поиска множества крайних вершин многоугольника заключается в том, что крайние вершины отыскиваются последовательно, одна за другой, в том порядке, в котором производится обход границы многоугольника. При этом, чтобы найти очередную крайнюю вершину p_j, необходимо лишь знать, какая крайняя вершина p_i, была найдена непосредственно перед этим (в самом начале в качестве p_i берется начальная вершина p_1, с которой начинается нумерация всех вершин многоугольника).

Процедура поиска очередной крайней вершины p_j заключается в следующем. Строим ориентированную прямую $L_{i,i+1}$, содержащую вершины p_i и p_{i+1}. Прямую ориентируем в направлении от p_i к p_{i+1}. Полуплоскость, расположенную слева от прямой $L_{i,i+1}$, обозначим через $H_{i,i+1}$. Проверяем, все ли вершины многоугольника содержит полуплоскость $H_{i,i+1}$. Если условие выполняется, то вершина p_{i+1} является крайней вершиной многоугольника и поиск заканчивается. В противном случае строим новую ориентированную прямую $L_{i,i+2}$, проходящую через вершины p_i и p_{i+2}, и проверяем, все ли вершины многоугольника содержит полуплоскость $H_{i,i+2}$. Если условие выполняется, то вершина p_{i+2} является крайней вершиной многоугольника и поиск заканчивается. Иначе строим следующую ориентированную прямую $L_{i,i+3}$ и так далее до тех пор, пока очередная полуплоскость $H_{i,i+k}$ не будет содержать в себе все вершины многоугольника. В результате будет найдена очередная (в порядке обхода границы) крайняя вершина p_j, где $j = i + k$.

После нахождения крайней вершины p_j, эта вершина используется аналогичным образом для поиска следующей по порядку крайней вершины и так далее. Процесс поиска крайних вершин заканчивается тогда, когда очередной найденной крайней вершиной оказывается начальная вершина p_1, основываясь на которой поиск и был начат.

Предложенный способ поиска множества крайних вершин проиллюстрируем на примере многоугольника, изображенного на рис. 4.

Поиск второй крайней вершины (первой крайней вершиной является начальная вершина p_1) начинается с рассмотрения вершины p_2 и проведения через p_1 и p_2 ориентированной прямой $L_{1,2}$. Полуплоскость $H_{1,2}$, расположенная слева от прямой $L_{1,2}$, содержит не все вершины многоугольника (что видно из рисунка), поэтому рассматривается следующая (в порядке обхода границы) вершина p_3 и через p_1 и p_3 проводится ориентированная прямая $L_{1,3}$. Полуплоскость $H_{1,3}$, расположенная слева от прямой $L_{1,3}$, также содержит не все вершины многоугольника, поэтому рассматривается следующая вершина p_4 и проводится ориентированная прямая $L_{1,4}$. Но и полуплоскости $H_{1,4}$ принадлежат не все вершины многоугольника, поэтому рассматривается вершина p_5 и проводится ориентированная прямая $L_{1,5}$. Так как полуплоскость $H_{1,5}$ содержит все вершины многоугольника, то вершина p_5 есть искомая крайняя вершина.

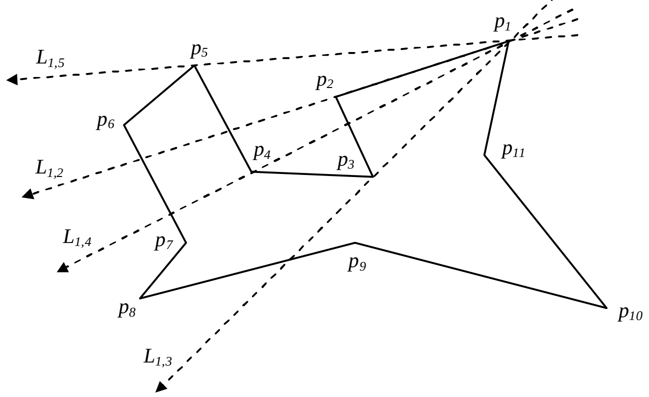

Рис. 4. Ориентированные прямые, необходимые для нахождения крайней вершины многоугольника

Далее аналогичным образом отыскивается третья крайняя вершина (причем для ее поиска проводятся ориентированные прямые, проходящие через вершину p_5) и так далее. В результате будут найдены все крайние вершины многоугольника: p_1, p_5, p_6, p_8, p_{10}.

Заключение. В настоящей работе описан способ нахождения множества крайних вершин многоугольника. Эта частная задача является одной из задач, решаемых в рамках изложенного в работе [5] метода построения канонической булевой формулы многоугольника. Тем самым указанный метод может быть легко доведен до формы алгоритма и, далее, переведен в форму программ на каком-либо языке программирования. В свою очередь, представление многоугольников булевыми формулами открывает новые возможности для решения широкого круга оптимизационных задач топологического проектирования интегральных схем путем использования развитого аппарата булевой алгебры.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Препарата, Ф. Вычислительная геометрия: введение / Ф. Препарата, М. Шеймос – Пер. с англ. – М.: Мир, 1989. – 478 с.
2. Фейнберг, В.З. Геометрические задачи машинной графики больших интегральных схем. – М.: Радио и связь, 1987. – 178 с.
3. Шестаков, Е.А. Автоматизированная система подготовки информации для формирования фотошаблонов / Е.А. Шестаков, А.А. Бутов, Т.Л. Орлова, А.А. Воронов // Искусственный интеллект. – Украина, Донецк. – 2008. – № 4. – С. 200–207.
4. Поттосин, Ю.В. Использование булевых функций для представления многоугольников // Вестник Томского Государственного университета. Управление, вычислительная техника и информатика / Ю.В. Поттосин, Е.А. Шестаков – 2008. – № 2(3). – С. 106–115.
5. Закревский, А.Д. Канонические булевы формулы многоугольников // Информатика. – 2009. – № 2. – С. 93–101.

11.11.10

BUTOV A.A. On the problem of finding a canonical Boolean formula polygon

Considered a method for finding the set of "extreme" vertices of the polygon, i.e., those points of its boundary, located at the junction of two segments, through which can be put straight line, not overlapping with any of the other segments of the border.

This particular problem is one of the objectives to be achieved within a well-known in the literature method of finding canonical Boolean formula of the polygon. Thus, this method can be easily brought to the form of the algorithm and, further, transferred into a program in some programming language.

The latter can be used, in particular, in computer-aided design of integrated circuits.