АЛГОРИТМ ОЦЕНКИ УРОВНЯ ЦИФРОВОЙ АВТОНОМИИ КОМПОНЕНТОВ ИНФРАСТРУКТУРЫ ЦИФРОВОГО ПРОСТРАНСТВА

Рыженко А. А.[1], Селезнёв В. М.[2]

DOI: 10.21681/2311-3456-2024-4-131-139

Целью работы *является разработка и формализация алгоритма оценки уровня цифровой автономии компонентов инфраструктуры цифрового пространства, позволяющие рассматривать цифровые данные как бикубическую систему хранения атрибутивной информации.*

Метод исследования: *методы мультимножества, концептуальное моделирование, алгоритмизация процессов, ресурсов и объектов.*

Результат исследования: *разработана модель и алгоритм оценки автономии цифровых ресурсов, позволяющих рассматривать данные цифровой среды не только как источник информации для обладателей, но и как микросистему, позволяющую хранить служебную и атрибутивную информацию, а также нежелательные инъекции. В качестве формального описания используется числовое представление алгебры мультимножеств, как одного из наиболее эффективного инструмента представления процессов бинарной системы данных. Полученная постановка решает актуальную проблему формализации данных – моделирование процессов изменения атрибутивной модели цифрового объекта, а также оценки возможных изменений.*

Научная новизна *заключается в разработке нового элемента концептуального моделирования деструкторов моделей – бикубическая система оценки уровня автономии цифровых объектов.*

Ключевые слова: *деструктор, моделирование, интеллектуальный агент, фасет, иерархия, правила перехода, автоном, цифровое пространство, система.*

ALGORITHM FOR ASSESSING THE LEVEL OF DIGITAL AUTONOMY OF DIGITAL SPACE INFRASTRUCTURE COMPONENTS

Ryzhenko A. A.[3], Seleznev V. M.[4]

The aim of the work *is to develop and formalize an algorithm for assessing the level of digital autonomy of digital space infrastructure components, allowing us to consider digital data as a bicubic system for storing attribute information.*

Research method: *multiset methods, conceptual modeling, algorithmization of processes, resources and objects.*

Research result: *a model and algorithm for assessing the autonomy of digital resources has been developed, allowing us to consider digital environment data not only as a source of information for owners, but also as a microsystem that allows storing service and attribute information, as well as unwanted injections. As a formal description, the numerical representation of multiset algebra is used, as one of the most effective tools for representing the processes of a binary data system. The resulting formulation solves the current problem of data formalization – modeling the processes of changing the attribute model of a digital object, as well as assessing possible changes.*

The scientific novelty *lies in the development of a new element of conceptual modeling of model destructors – a bicubic system for assessing the level of autonomy of digital objects.*

Keywords: *destructor, modeling, intelligent agent, facet, hierarchy, transition rules, autonomy, digital space, system.*

1 Рыженко Алексей Алексеевич, кандидат технических наук, доцент, доцент кафедры информационной безопасности, Финансовый университет при Правительстве РФ, г. Москва, Россия. E-mail: AARyzhenko@fa.ru

2 Селезнёв Владимир Михайлович, кандидат технических наук, заведующий кафедрой информационной безопасности, Финансовый университет при Правительстве РФ, г. Москва, Россия. E-mail: VMSeleznyov@fa.ru

3 Aleksey A. Ryzhenko, Ph.D., Associate Professor, Financial University under the Government of the Russian Federation, Moscow. E-mail: AARyzhenko@fa.ru

4 Vladimir M. Seleznev, Ph.D., Head of department of information security, Financial University under the Government of the Russian Federation, Moscow. E-mail: VMSeleznyov@fa.ru

Введение

Существует типичное заблуждение, что в цифровой среде передается информация, которую можно отправить, получить и т.д. На уровне рядового пользователя вполне можно этим и ограничиться. Обладатели информации не обязаны знать, что происходит с данными при кодировании и декодировании, а также какую дополнительную служебную информацию хранит в себе каждый файл любой операционной системы [1]. В качестве подтверждения можно проанализировать терминологию, прописанную в ключевых федеральных законах – 149-ФЗ и 152-ФЗ. Таких понятий как *информация*, *владелец*, *хозяин* и даже *пользователь* просто не существует. Единственный статус у каждого субъекта цифровой среды – *обладатель информации*. Данный уровень достаточно подробно рассмотрен в юридической литературе, особый акцент делается на разнице между обладателем и правообладателем информации [2]. С другой стороны, термин *данные* не несет в себе никакой семантической нагрузки, представлен как набор *сигналов*, передаваемых через каналы связи. Достаточно корректное определение. В результате, под данными можно понимать не только то, что отображается обладателю, но и то, что видит операционная система на прикладном уровне, и то, что может быть незаконно добавлено (инъекция) в тело файла. Именно это обстоятельство и мешает юридической системе дать точное определение – что такое цифровая информация. В качестве выхода из данной ситуации было предложено делать оценку уровня автономии каждого цифрового ресурса, что даст возможность рассматривать каждый файл не только как объект, содержащий какую-либо информацию, но и принадлежность, что является одним из ключевых критериев цифрового суверенитета [3, 4].

Ранее была рассмотрена технология обучения интеллектуального агента умной бот-сети [5], а также этапы формирования единой модели данных, основанной на единой базе правил и распределенной архитектуре баз ассоциаций [6]. Аналогичные исследования проводятся и по другим современным направлениям, где необходимо проводить оценку для формирования критериев обратной связи [7].

2. Новый подход формализации данных, основанный на алгебре процессов

Рассмотрим простую задачу оценки уровня автономии цифрового объекта на этапах жизненного цикла. Условие: для цифрового изображения, передаваемого по каналам связи, на основе атрибутивной схемы необходимо предугадать возможное заражение инъекцией вредоносом. Исходные наборы атрибутов представлены в табл. 1.

Таблица 1.

Атрибуты цифрового изображения

Внутренние атрибуты	Внешние атрибуты
1 – размер × разрешение 2 – глубина цвета 3 – тип кодирования / сжатие 4 – устройство захвата 5 – дата создания	1 – имя 2 – атрибуты 3 – размер файла 4 – источник 5 – дата изменения

Представим цифровое изображение не как графический файл, а как битовая последовательность блоков кодирования изображения. Большиинство свободно распространяемых графических форматов состоят из двух больших блоков: первый – кодовая матрица изображения, второй – атрибутивная составляющая и блок служебной информации. Как правило, второй блок заполняется изначально нулями и не содержит необходимую для самого файла информацию. Именно этим фактором и пользуются инъектологи, прописывая во второй блок другие файлы (инъекции). Как выявить вложенный файл – до сих пор не существует универсальных алгоритмов и многие антивирусные программы пропускают файлы с инъекцией. Аналогичные исследования отражены в ряде актуальных работ, например [8, 9]. Рассмотрим сначала пример использования атрибутивной модели на дереве решений, затем отображение в виде бикубической модели и перейдем к формальному описанию и примерам использования на практике.

Первый этап: на мобильном устройстве выполнен захват и формирование нового цифрового изображения. При конвертировании в доступный формат автоматически добавился первый список атрибутов из двух частей (табл. 1).

Второй этап: пользователь решил отправить фото изображение через мессенджер. Прикрепляет цифровое фото как контейнер и отправляет получателям. При передачи запускается служба аудита, встроенная в мессенджер, которая автоматически добавляет в служебный блок корректировку – другой метод сжатия (фото изображение становится меньше в байтовом эквиваленте), дата создания нового изображения и т.д. (рис. 1).

Примечание: на рисунках 1–3 модули I–V – этапы процесса анализа атрибутов, 1–5 – атрибуты таблицы 1. В кружках обозначены статусы процессов от 1 до 5, о чем будет подробнее представлено в третьей части формализации.

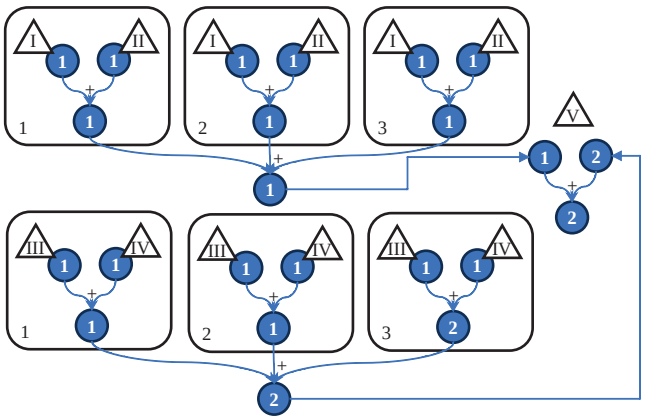

Рис. 1. Выявление вредоноса по правилу «тип кодирование – размер»

В результате пользователь получает изображение с измененным набором атрибут. Если пользователю надо только само изображение, т.е. первый блок, то операционной системе больше нужен второй описательный блок для получения служебной информации. При этом сразу необходимо учесть важный фактор: если сам отправитель умышленно сделал инъекцию в цифровое изображение, то у него могут возникнуть проблемы с отправлением с сохранением целостности файловой архитектуры. Другими словами, если отправитель отправит зараженный файл просто в ветку чата, то второй блок будет изменен встроенной службой, но если отправит как вложение файла без распознования мессенджером файла как изображения, то инъекция будет передана получателю. На рисунке 1 отображена исходная ветка дерева анализа и принятия решений решений. Встроенный агент обнаружил, что полученный файл имеет другой метод сжатия и другой размер. При этом возможно два вариант: изменения мессенджером и изменение отправителем. Происходит дальнейший анализ атрибутов.

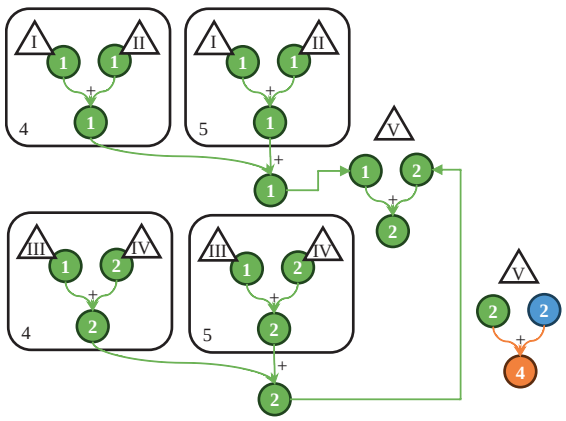

Рис. 2. Выявление вредоноса по правилу «источник – дата»

Производится проверка атрибутов источника информации и даты изменения. Если сжатие изменил мессенджер, то это отразится в атрибуте источника, если отправитель – изменений не будет. Далее происходит проверка даты и времени изменения файла. Атрибут не изменится и будет меньше даты и времени отправления отправителем, если инъекция была сделана самим отправителем. Если изменения внес мессенджер, то дата изменится на более позднюю и отразится в атрибутах (рис. 2). В результе четыре атрибута дают вариант решения (рис. 3).

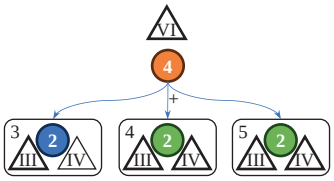

Рис. 3. Итоговый узел дерева решений для выявление вредоноса по правилам

Необходимо учесть, что функция XOR дает возможность исключения альтернативных веток с исключающим решением. В результате для исполнителя данного дерева (агента) строится простое правило:

$$((1+1)\oplus(1+1)\oplus(1+1)) + ((1+1)\oplus(1+1)\oplus(1+2)) + + ((1+1)\oplus(1+1)) + ((1+2)\oplus(1+2)) = 4 \xrightarrow[5]{} 2 + 2 + 2.$$

Аналогичные результаты можно обнаружить в ряде научных работ, например [10, 11].

Для формального описания двухкритериальной системы атрибутов можно использовать бикубические модели фасетных данных. Тогда графически описанный пример можно представить как куб с шестью фасетными гранями: три грани атрибутов отправители и три – получателя (рис. 4) [5]. На четырех гранях строятся фасеты атрибутов, на двух – деревья преобразования, представленные выше.

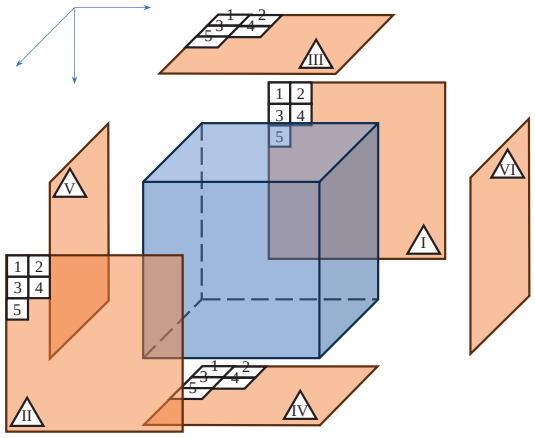

Рис. 4. Бикубическая модель атрибутов цифрового автонома

DOI: 10.21681/2311-3456-2024-4-131-139

Представленный пример отображает схематическое описание процессов построения правила дерева решения. Интересные аналогичные результаты можно наблюдать в работах по использованию алгебры мультимножеств для решения практических задач, например [12, 13]. Далее рассмотрим алгебраическую составляющую модели.

2.1. Формализация и моделирование процессов перехода состояний агентов

Разберем простейшую задачу сложения процессов при выборе альтернативного решения в узловой точке произвольного дерева решений. Алгебра синтаксической формы представления знаний предполагает следующую последовательность действий при классическом сложении: берем один закрытый процесс (т.е. процесс в пассивном режиме), открываем один активный процесс (т.е. запускаем в действие) – это «один». Открываем второй процесс – это «два» и т.д. Если предполагается, что активные и пассивные процессы разные, то решение будет в синтаксической постановке путем сложения процессов (частный случай). Тем не менее, существует и другой сценарий, когда процессы возвращаются обратно в закрытое состояние.

Например, сканер одной социальной сети был запущен по расписанию на поиск личности. Процесс закончен. Два варианта вывода результата: информация получена, и информация не получена. В первом случае проверяется количество решений. Для каждого строится отдельная ветка дерева решений. При этом параллельно второй процесс запускает поиск информации по личности в другой социальной сети, но ориентируется на варианты результатов первого поиска. Далее, для отключения возможности неразрешимых коллизий при несоответствии информации одновременных поисков решений используется семантическая алгебра процессов.

Алгоритм данной постановки, следующий: запускаем (открываем) один процесс – это один. Закрываем обратно и открываем снова – «+1», т.е. процесс произошел или добавился дважды, а результат при этом не изменился. Получаем, что «1+1=1». Данный сценарий используется в прикладной математике как логическая аддитивная Булева функция ИЛИ. Развивая процесс (не забывая о критерии целостности, т.е. всегда есть предполагаемая максимальная граница достаточности запущенных процессов) получаем, что, открыв два процесса, а потом, открыв и закрыв один из них имеем: «2+1=2». Расширяя до предела целого (допустим, узел дерева предполагает пять одновременно запущенных альтернативных процесса, т.е. равно «5»), имеем: «3+1=3», «4+1=4», «5+1=5». Здесь нет «и так далее» так как процессы в пределах целого «5» заканчиваются. Но, внутри целого можно также использовать и стандартный аддитивный сценарий сложения: «4+1=5» или «2+2=4», так как в пределах показателя целого можем использовать любые аддитивные процессы. Также поддерживается функция с нулем: «0+0=0», «0+1=1», ..., «0+5=5» (все процессы закрыты – все процессы открыты). Для того чтобы далее не путать результат классического сложения и сложения от целого обозначим первый (классический) вариант знаком «=», а результат по основанию от целого знаком «$\underset{5}{\rightarrow}$» (в данном случае, основание равно «5»). Независимо от величины целого в диапазоне от 0 до бесконечности или $[0,\infty)$, *верхняя граница* будет *не более максимального значения – значения* самого *целого*. Например, если целое (максимально допустимое количество одновременно обрабатываемых альтернатив) равно десяти, то целое равно десяти, но обозначение следствия к множеству решений будет зависеть от условия задачи, о чём будет сказано ниже.

Нижняя граница имеет при описанных условиях значение равное нулю (все процессы закрыты). Однако в задачах с отрицательной величиной (процессы не открываются – закрываются, а закрываются и открываются, т.е. обратная задача) допускается взаимообратное целому число, т.е. для данного примера нижняя граница будет равна «–5», а верхняя – «5» (целое, все процессы закрыты). В результате сценарий с одним прячущимся процессом и граничные сценарии в пределах целого *имеют одно основание*. Данная процедура возможна при уже запущенных интеллектуальных агентах обратным целевым деревом. При этом допускается, что прямое дерево будет использовать агентов обратного дерева решений. Можно предположить, что:

для любого целого верхней границей всегда будет значение (показатель) целого, нижней – ноль или взаимно-противоположное целому значение верхней границы. (П.3)

Как и ранее, далее для примера рассматриваем целое – максимум пять активных процессов, но изменим начальное условие. Если одновременно действующих открыл / закрыл процессов будет не один, а два (т.е. одновременно происходит поиск выбора решений для нескольких узловых точек прямого дерева решений), то результатом *аддитивного сложения* (например, на два) может быть, как два, так и три, и граничный – четыре (расширенный вариант функции XOR):

– «$2 + 2 \underset{5}{\rightarrow} 2$» – два процесса открыли, их же закрыли и открыли вновь – *обновляемый сценарий*. Подтверждение: $2 + 2 = (1_1 + 1_1) + (1_1 + 1_1) \underset{5}{\rightarrow} 1 + 1 = 2$. Так как $1_1 + 1_1 \underset{5}{\rightarrow} 1_1$;

– «$2 + 2 \underset{5}{\rightarrow} 3$» – два процесса открыли, один из них закрыли и открыли его же, но с другим (не равном первому) процессом – увеличивающий сценарий. Подтверждение: $2 + 2 = (1 + 1) + (1_1 + 1_1) \underset{5}{\rightarrow} 2 + 1_1 = 3$. Так как $1_1 + 1_1 \underset{5}{\rightarrow} 1_1$;
– «$2 + 2 \underset{5}{\rightarrow} 4$» – два процесса открыли, и, не закрывая предыдущие, открыли еще два – *классический сценарий*. Подтверждение: $2 + 2 = (1 + 1) + (1 + 1) = 4$.

Если действующих процессов не равное количество, т.е. в закрытии и открытии участвуют, допустим, два и три, то правило (П.1) сохраняется. Но, необходимо выполнять дополнительное условие: *максимальное количество закрытых и открытых процессов в пределах целого должно быть однозначно одинаковым*.

Особенностью предыдущего сценария является учет только предположения о целостности (П.1). Существует еще *субтрактивное сложение*, когда сумма может быть меньше любого слагаемого. Например, «$2 + 2 \underset{5}{\rightarrow} 1$» (два открытых процесса закрыли и открыли только один из них – отнимающий сценарий. Подтверждение: $2 + 2 = (1_1 + 1_1) + (1_1 + 1_1) \underset{5}{\rightarrow} 1_1 + 1_1 \underset{5}{\rightarrow} 1$. Так как $1_1 + 1_1 \underset{1}{\rightarrow} 1_1$. Рассмотрим данный сценарий (с теми же начальными условиями) на примере паевой геометрической фигуры (рис. 5).

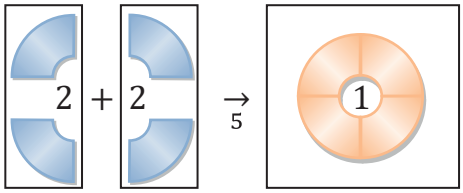

Рис. 5. Пример использования субтрактивной функции сложения внутри целого (пяти)

Раскрывая сущность целого, далее покажем, что результат любой суммы элементов целого может быть равен верхней границе, т.е. значению самого целого (рис. 6). Для рассматриваемого примера «$2 + 2 \underset{5}{\rightarrow} 5$» добавочный сценарий. Подтверждение: $2 + 2 = (1 + 1) + (1 + 1_1) \underset{5}{\rightarrow} (1 + 1) + (1 + 1_1 + 1_1) \underset{5}{\rightarrow} 5$, так как $1_1 \underset{1}{\rightarrow} 1_1 + 1_1$.

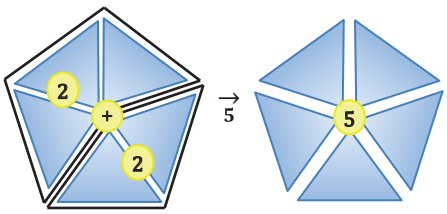

Рис. 6. Пример использования добавочного сценарий аддитивной функции сложения внутри целого пяти

Функция аддитивности и субтрактивности сохраняется и далее. Например, для целого десяти можно открывать и закрывать как все пять процессов, так и любое количество в пределах целого – пяти. Можно предположить, что:

показатель суммы определяет количество вариантов сложения, но не более целого. Ноль является нижним значением – исключение. (П.2)

Графически аддитивный алгоритм можно представить аналогично субтрактивному алгоритму сложения в пределах целого (рис. 7).

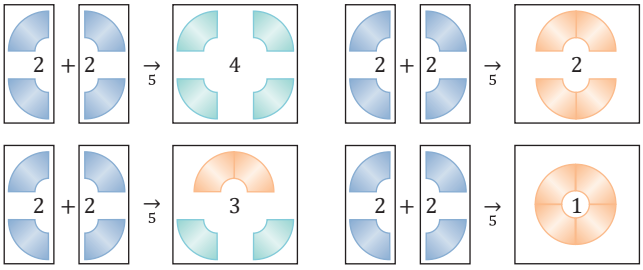

Рис. 7. Пример использования аддитивной и субтрактивной функций сложения внутри целого пяти

При изменении значения целого будет меняться количество возможных вариантов результатов сложения, и верхняя граница значений. Следовательно, можно предположить, что:

в пределах целого аддитивные и субтрактивные произвольные суммы элементов множества могут принимать значения равные одному из элементов этого множества. (П.3)

Другими словами, значение суммы может быть равно любому своему слагаемому.

аддитивной суммой любых элементов целого является множество последовательных чисел от наименьшего равного элементу до максимального равному результату суммы классического сложения в пределах целого или значению целого. (П.4)

Для правил П.3 и П.4 сумма нулей является исключением:

субтрактивной суммой любых элементов целого является множество последовательных чисел от наименьшего равного единице до максимального равному наименьшему элементу. (П.5)

Формальное описание для данного сценария может выглядеть следующим образом:

$$\sum_{i=0}^{n} a_i \underset{m}{\longrightarrow} [1,m], \text{ где } n \in [1,\infty),\ m \in [0,n] \quad (1.1)$$

a_i – элемент множества, m – целое, n – произвольный элемент.

4. От формальной теории к практике использования

Рассмотрим несколько сценариев использования описанной модели на практических примерах в цифровой среде.

Сценарий 1: используем метод формирования дерева сценария достижения итоговой цели одного потока данных. Предполагается, что на всех промежуточных узлах дерева решений используются независимые решения, и лишь некоторые должны быть частью итогового сценария. Например, в рассмотренной ранее задаче кражи личности каждый последующий этап дерева решений будет зависеть от результатов предыдущего. При этом необходимо рассмотреть все возможные варианты сбора информации, но фактически образ личности будет составляться только из тех узловых точек, которые соответствуют целевому поиску. Иными словами, если на первых этапах вариантов решений будет множество (например, ассоциативный поиск по фотографии может представить множество решений), то каждый последующий поиск будет уже ограничен предыдущим выбором соответствия (включая ID, номера привязки к документам, средствам коммуникации и т.д.).

Задача 1 (частный случай): сбор данных состоит из двух решений на узловой точке, заложено в решение *пять* этапов поиска. Пусть на *первом этапе* найдено *шесть* альтернативных решений как результат основного поиска. Ограничением заложена допустимость – *три* решения (*два* промежуточных и *третий* итоговый). Следовательно, используются все *шесть* решений, но *три* из них будут подводящие, не выпадающие из общего поиска. В другом поиске получено *пять* решений, а разрешено только *два*, но оба должны подвергнуться дальнейшему использованию (такое вполне возможно если у пользователя несколько аккаунтов в одной социальной сети). Следовательно, *три решения* из них будут подводящие к итоговому. Необходимо также учесть, что общая

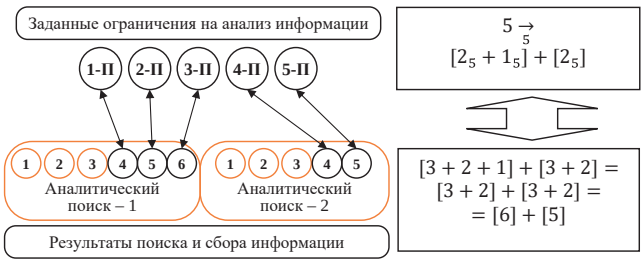

Рис. 8. Схематическое представление решения первой задачи

сумма всех процессов не может выйти за заданный верхний предел (рис. 8). *Необходимо построить и решить логическое выражение, описывающее процесс построения итогового решения*.

Обобщенное правило перехода состояний процесса поиска и выбора будет представлено следующим образом:

$$[3 + 2_5 + 1_5] + [3 + 2_5] \underset{5}{\longrightarrow} 5.$$

Согласно данному правилу, полученным промежуточным результатам поиска необходимо следовать следующим ограничениям:

- нельзя передать все результаты одного поиску;
- минимальное количество полученных решений одного поиска, используемых для проекта, не должно превышать максимальное количество допустимых решений самого проекта.

Сценарий 2: предположим, что некий «тролль» поселился в открытой ветке форума социальной сети ведомственной организации и начал вести свою пропаганду, используя слова и словосочетания негативного контента. Агенту необходимо провести анализ ленты используя контент сообщений за двое суток. Найти точки соприкосновения с официальной лентой Министерства, поставить соответствующие гиперссылки.

Новости 18.00

В 16.30 оперативный дежурный места массового скопления людей X г. Y_1 обнаружил чужеродный предмет возле кафе-столовая$_2$. Дежурный вызвал по телефону оперативную служба$_3$. В 16.40 произошел взрыв$_4$ мусорного бачка. Оперативная служба$_3$ прибыла с опозданием только в 16.45. Оставили машину за территорией организации$_2$ и чего-то ждали. Начальник оперативного штаба прибыл только в 17.00, когда уже было поздно. Спасатели$_3$ медленно стали разбирать образовавшийся завал$_4$. Пожар$_4$ продолжается, жертвы не известны. Уже уничтожен один этаж$_2$. Верхний этаж$_2$ пока горит, задымление, но спасатели что-то тушат и выносят.

Новости 8.00 следующего дня

Только к 19.00 удалось потушить пожар$_4$ на этаже$_2$ организации X_1. Уничтожен один комплекс$_2$ организации. Значительные повреждения основного сооружения. Сотрудники спасательной службы$_3$ действовали как сонные мухи. Если бы все было иначе, возможно последствия были бы намного меньше.

Проект в виде официальной новостной ленты

Вызов диспетчерской организации X_1 города Y_1 поступил в 16.35. Дежурный X_1 оставил заявку о пожаре$_4$ в районе второго этажа$_2$. В 16.38 на вызов выехала дежурная бригада спасателей$_3$. Теоретическое время прибытия – 10 мин., фактическое

составило 8 мин. На момент прибытия (16.45) произошел взрыв$_4$ мусорного бака$_2$. Пламя и осколки$_4$ раскинулись по территории объекта. Из-за отсутствия искусственных заграждений возможно дальнейшее распространение за территорию объекта. В связи с увеличением сложности чрезвычайной ситуации, собран специализированный оперативный штаб. До решения штаба о ликвидации оперативная бригада производила локализацию источника в пределах объекта. В 17.00 получены рекомендации центрального аппарата – алгоритм локализации и ликвидации. Очаги возгорания устранены к 19.00. Производится устранение последствий средствами спасательной службы$_3$.

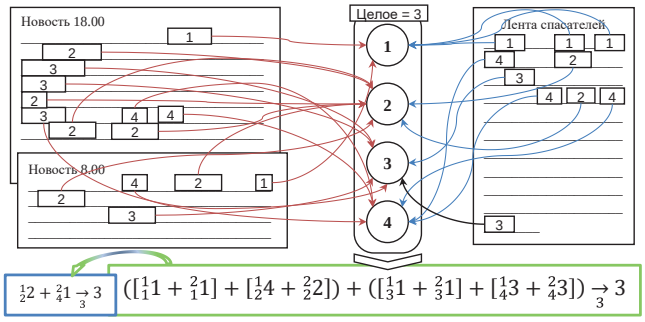

Рис. 9. Графическое представление условия задачи

Задача 2: для сообщений в пределах 100±20 слов допускается не более 3 ссылок [целое равно 3]. Второе сообщение в два раза меньше, чем первое [ссылок в два раза меньше]. Количество для каждой очередной ссылки в два раза меньше предыдущей. Остается первая попавшаяся ссылка из выбранных. *Задание:* вставить в ключевые позиции сообщения гиперссылки (рис. 9).

Разбираем сообщения (условия):

- слова из базы искусственного алфавита потенциальных источников попадаются в сообщениях 8 раз (индексы 1 и 2) $[{}^{1}_{1}1 + {}^{2}_{1}1]$ и $[{}^{1}_{2}4 + {}^{2}_{2}2]$ соответственно;
- слова из базы искусственного алфавита АСФ и типов аварий попадаются в сообщениях 8 раз (индексы 3 и 4) – $[{}^{1}_{3}1 + {}^{1}_{3}1]$ и $[{}^{1}_{4}3 + {}^{2}_{4}3]$ соответственно;
- первый алфавит используется в начале текста или в первых сообщениях (завязка), второй в конце или последних (развязка).

Общая функция перехода состояний выглядит следующим образом:

$$([{}^{1}_{1}1 + {}^{2}_{1}1] + [{}^{1}_{2}4 + {}^{2}_{2}2]) + ([{}^{1}_{3}1 + {}^{2}_{3}1\ ] + [{}^{1}_{4}3 + {}^{2}_{4}3]) \underset{3}{\longrightarrow} 3.$$

Дальше необходимо сократить количество ссылок до трех возможных:

- первая и третья скобки попадают под правило $1_3 + 1_3 \underset{3}{\longrightarrow} 1_3$ если не хотим добавить чего-то нового (в условии этого нет). Результат:

$$([{}^{1}_{1}1] + [{}^{1}_{2}4 + {}^{2}_{2}2]) + ([{}^{2}_{3}1] + [{}^{1}_{4}3 + {}^{2}_{4}3]) \underset{3}{\longrightarrow} 3;$$

- четвертая скобка имеет равные элементы, но, $3_3 + 3_3 \underset{3}{\longrightarrow} 3_3$, поглощаем первое (третье условие). Результат:

$$([{}^{1}_{1}1] + [{}^{1}_{2}4 + {}^{2}_{2}2]) + ([{}^{2}_{3}1] + [{}^{2}_{4}3\ ]) \underset{3}{\longrightarrow} 3;$$

- раскрываем скобки с одним слагаемым и внешние:

$${}^{1}_{1}1 + [{}^{1}_{2}4 + {}^{2}_{2}2] + {}^{2}_{3}1 + {}^{2}_{4}3 \underset{3}{\longrightarrow} 3;$$

убираем единичные ссылки, т.к. попадают под правило $1_3 + 1_3 \underset{3}{\longrightarrow} 1_3$:

$$[{}^{1}_{2}4 + {}^{2}_{2}2] + {}^{2}_{4}3 \underset{3}{\longrightarrow} 3;$$

- в первой скобке «4» выпадает за целое, разделяем и выносим за соответствие с проектом:

$$[{}^{1}_{2}2 + {}^{1}_{2}2 + {}^{2}_{2}2] + {}^{2}_{4}3 \underset{3}{\longrightarrow} 3;$$

- раскрываем внутренние скобки:

$${}^{1}_{2}2 + {}^{2}_{4}3 \underset{3}{\longrightarrow} 3;$$

- второе сообщение в два раза меньше первого, значит:

$${}^{1}_{2}2 + ({}^{2}_{4}2 + {}^{2}_{4}1) \underset{3}{\longrightarrow} 3;$$

- раскрываем внутренние скобки:

$${}^{1}_{2}2 + {}^{2}_{4}2 + {}^{2}_{4}1 \underset{3}{\longrightarrow} 3;$$

- во втором сообщении должно быть в два раза меньше ссылок. Выполняем третье условие, тогда:

$${}^{1}_{2}2 + {}^{2}_{4}1 \underset{3}{\longrightarrow} 3.$$

Получаем две ссылки в первом сообщении с индексом «2» и одну ссылку во втором сообщении с индексом «4». В примере индексы дописываются для удобства анализа, хотя в задачах указывать не обязательно.

Заключение

Обзор современной литературы дает недвусмысленно понять, что несмотря на многочисленные попытки хоть как-то приблизить теорию к практике, многие научные деятели до сих пор не хотят отображать в своих публикациях практику использования научных достижений. Бывает и наоборот, исследователи приводят готовые фрагменты кода без соответствующих комментариев [14]. Данный фактор

не дает возможность полностью оценить как научные, так и практические результаты научных исследований. В данной работе сделана попытка приблизить достаточно мощный математический аппарат теории мультимножеств к практическим задачам цифрового пространства. Полученные результаты уже достаточно хорошо проявили себя на практике при построении контуров защиты корпоративных систем, предоставляющих доступ из внешней среды. Также развитие получило еще одно современное направление – построение дискретных пространств бинарных данных при создании кода быстрого отклика (*Quick Response Code*) корпоративного уровня, что дает дополнительную защиту к передаваемым данным. Некоторые практические результаты по неоднозначности кодирования информации, а также сокрытия искусственных алгоритмов отражено в ряде статей, например [15–17].

Литература

1. *Рыженко А. А. Организация системы подготовки сотрудников организаций в сфере противоборства механизмам социальной инженерии // Проблемы управления безопасностью сложных систем. Материалы XXX международной конференции. Под общей редакцией А. О. Калашникова, В. В. Кульбы. Москва, 2022. С. 337–342*
2. *Правообладатель данных или обладатель информации – кого имеет в виду закон? – режим доступа: https://zakon.ru/blog/2020/11/14/pravoobladatel_dannyh_ili_obladatel_informacii__kogo_ imeet_v_vidu_zakon (дата посещения 06.06.2024 г.)*
3. *Рыженко А. А., Рыженко Н. Ю. Интеллектуальные деструкторы и мобильные банковские клиенты // Актуальные проблемы и перспективы развития экономики: Труды XXI Международной научно-практической конференции. Симферополь-Гурзуф, 20–22 октября 2022 год. / Под ред. д.э.н., д.пед.н., профессора Н. В. Апатовой. – Симферополь: Издательский дом КФУ имени В. И. Вернадского, 2022. – с. 241-242.*
4. *Рыженко А. А., Рыженко Н. Ю. Утечки данных и рейтинги банков // Теория и практика экономики и предпринимательства. Труды XX Международной научно-практической конференции. Под редакцией Н. В. Апатовой. Симферополь, 2023. С. 215-216*
5. *Рыженко А. А. Умная бот-сеть или модель интеллектуального деструктора // Вопросы кибербезопасности. 2023. № 5(57). С. 60–68. DOI: 10.21681/2311-3456-2023-5-60-68*
6. *Рыженко А. А., Селезнёв В. М. Модель систематизации классификаторов деструктивных и конструктивных событий цифрового пространства // Вопросы кибербезопасности. 2024. № 3(61). С. 113-119. DOI: 10.21681/2311-3456-2024-3-113-119*
7. *Kaushik, B., Sharma, R., Dhama, K. et al. Performance evaluation of learning models for intrusion detection system using feature selection. J Comput Virol Hack Tech 19, 529–548 (2023). https://doi.org/10.1007/s11416-022-00460-z*
8. *Hashemi, H., Samie, M. E. & Hamzeh, A. IFMD: image fusion for malware detection. J Comput Virol Hack Tech 19, 271-286 (2023). https://doi.org/10.1007/s11416-022-00445-y*
9. *Alaeiyan, M., Parsa, S. A hierarchical layer of atomic behavior for malicious behaviors prediction. J Comput Virol Hack Tech 18, 367–382 (2022). https://doi.org/10.1007/s11416-022-00422-5*
10. *Dalla Preda, M., Ianni, M. Exploiting number theory for dynamic software watermarking. J Comput Virol Hack Tech 20, 41–51 (2024). https://doi.org/10.1007/s11416-023-00489-8*
11. *Babash, A. V. XOR ciphers model and the attack to it. J Comput Virol Hack Tech 18, 275–283 (2022). https://doi.org/10.1007/s11416-022-00419-0*
12. *Karamitas, C., Kehagias, A. Improving binary diffing speed and accuracy using community detection and locality-sensitive hashing: an empirical study. J Comput Virol Hack Tech 19, 319–337 (2023). https://doi.org/10.1007/s11416-022-00452-z*
13. *Nikolopoulos, S. D., Polenakis, I. Behavior-based detection and classification of malicious software utilizing structural characteristics of group sequence graphs. J Comput Virol Hack Tech 18, 383–406 (2022). https://doi.org/10.1007/s11416-022-00423-4*
14. *Casolare, R., Fagnano, S., Iadarola, G. et al. Picker Blinder: a framework for automatic injection of malicious inter-app communication. J Comput Virol Hack Tech 20, 331–346 (2024). https://doi.org/10.1007/s11416-023-00510-0*
15. *Секреты USA в Micro QR Code M4 (часть 1). – режим доступа: https://habr.com/ru/articles/781858/ (дата посещения 06.06.2024 г.)*
16. *Секреты USA в Micro QR Code M2 (часть 2). – режим доступа: https://habr.com/ru/articles/782488/ (дата посещения 06.06.2024 г.)*
17. *Секреты USA в Micro QR Code M3 (часть 3). – режим доступа: https://habr.com/ru/articles/782772/ (дата посещения 06.06.2024 г.)*

References

1. *Ryzhenko A. A. Organizacija sistemy podgotovki sotrudnikov organizacij v sfere protivoborstva mehanizmam social'noj inzhenerii // Problemy upravlenija bezopasnost'ju slozhnyh sistem. Materialy XXX mezhdunarodnoj konferencii. Pod obshhej redakciej A. O. Kalashnikova, V. V. Kul'by. Moskva, 2022. S. 337–342*
2. *Pravoobladatel' dannyh ili obladatel' informacii – kogo imeet v vidu zakon? – rezhim dostupa: https://zakon.ru/blog/2020/11/14/pravoobladatel_dannyh_ili_obladatel_informacii__kogo_ imeet_v_vidu_zakon (data poseshhenija 06.06.2024 g.)*
3. *Ryzhenko A. A., Ryzhenko N. Ju. Intellektual'nye destruktory i mobil'nye bankovskie klienty // Aktual'nye problemy i perspektivy razvitija jekonomiki: Trudy XXI Mezhdunarodnoj nauchno-prakticheskoj konferencii. Simferopol'-Gurzuf, 20-22 oktjabrja 2022 god. / Pod red. d.je.n., d.ped.n., professora N. V. Apatovoj. – Simferopol': Izdatel'skij dom KFU im. V.I. Vernadskogo, 2022. – s. 241-242.*
4. *Ryzhenko A. A., Ryzhenko N. Ju. Utechki dannyh i rejtingi bankov // Teorija i praktika jekonomiki i predprinimatel'stva. Trudy XX Mezhdunarodnoj nauchno-prakticheskoj konferencii. Pod redakciej N. V. Apatovoj. Simferopol', 2023. S. 215-216*
5. *Ryzhenko A. A. Umnaja bot-set' ili model' intellektual'nogo destruktora // Voprosy kiberbezopasnosti. 2023. № 5(57). S. 60–68. DOI: 10.21681/2311-3456-2023-5-60-68*
6. *Ryzhenko A. A., Seleznjov V. M. Model' sistematizacii klassifikatorov destruktivnyh i konstruktivnyh sobytij cifrovogo prostranstva // Voprosy kiberbezopasnosti. 2024. № 3(61). S. 113–119. DOI: 10.21681/2311-3456-2024-3-113-119*
7. *Kaushik, B., Sharma, R., Dhama, K. et al. Performance evaluation of learning models for intrusion detection system using feature selection. J Comput Virol Hack Tech 19, 529–548 (2023). https://doi.org/10.1007/s11416-022-00460-z*

8. *Hashemi, H., Samie, M. E. & Hamzeh, A. IFMD: image fusion for malware detection. J Comput Virol Hack Tech 19, 271–286 (2023). https://doi.org/10.1007/s11416-022-00445-y*
9. *Alaeiyan, M., Parsa, S. A hierarchical layer of atomic behavior for malicious behaviors prediction. J Comput Virol Hack Tech 18, 367–382 (2022). https://doi.org/10.1007/s11416-022-00422-5*
10. *Dalla Preda, M., Ianni, M. Exploiting number theory for dynamic software watermarking. J Comput Virol Hack Tech 20, 41–51 (2024). https://doi.org/10.1007/s11416-023-00489-8*
11. *Babash, A. V. XOR ciphers model and the attack to it. J Comput Virol Hack Tech 18, 275–283 (2022). https://doi.org/10.1007/s11416-022-00419-0*
12. *Karamitas, C., Kehagias, A. Improving binary diffing speed and accuracy using community detection and locality-sensitive hashing: an empirical study. J Comput Virol Hack Tech 19, 319–337 (2023). https://doi.org/10.1007/s11416-022-00452-z*
13. *Nikolopoulos, S. D., Polenakis, I. Behavior-based detection and classification of malicious software utilizing structural characteristics of group sequence graphs. J Comput Virol Hack Tech 18, 383–406 (2022). https://doi.org/10.1007/s11416-022-00423-4*
14. *Casolare, R., Fagnano, S., Iadarola, G. et al. Picker Blinder: a framework for automatic injection of malicious inter-app communication. J Comput Virol Hack Tech 20, 331–346 (2024). https://doi.org/10.1007/s11416-023-00510-0*
15. *Sekrety USA v Micro QR Code M4 (chast' 1). – rezhim dostupa: https://habr.com/ru/articles/781858/ (data poseshhenija 06.06.2024 g.)*
16. *Sekrety USA v Micro QR Code M2 (chast' 2). – rezhim dostupa: https://habr.com/ru/articles/782488/ (data poseshhenija 06.06.2024 g.)*
17. *Sekrety USA v Micro QR Code M3 (chast' 3). – rezhim dostupa: https://habr.com/ru/articles/782772/ (data poseshhenija 06.06.2024 g.)*