



ВОССТАНОВЛЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ФАЙЛОВЫХ ОПЕРАЦИЙ С ПРИМЕНЕНИЕМ ТЕОРИИ ГРАФОВ ПРИ ПРОВЕДЕНИИ КОМПЬЮТЕРНЫХ ИССЛЕДОВАНИЙ

В статье представлен способ решения задачи по восстановлению последовательности операций, производимых над файлом, в ОС Windows с применением теории графов. Процесс изменения временных отметок представлен в виде ориентированного графа, в котором вершинами являются состояния временных отметок исследуемого файла, а ребрами — операции, производимые над ним. Для восстановления последовательности файловых операций необходимо определить все возможные маршруты между вершинами, соответствующими возможному начальному и имеющемуся конечному состояниям временных отметок. В работе рассмотрены и описаны алгоритмы поиска маршрутов в глубину и ширину. Сделан вывод о предпочтительности поиска в глубину. Результаты его применения продемонстрированы на не-скольких примерах.

Ключевые слова: временные отметки, файловые операции, компьютерная криминалистика, оргграф перехода, поиск в ширину, поиск в глубину.

RECOVERY THE SEQUENCE OF FILE OPERATIONS USING GRAPH THEORY IN COMPUTER FORENSICS INVESTIGATIONS

The paper presents a method for solving the problem of recovery the sequence of file operations in the Windows operating system using graph theory. The process of changing timestamps is represented as a directed graph, in which the vertices are the states of the timestamps of the file, and the edges are the file operations that were performed on the file. To restore the sequence of file operations, you need to determine all possible routes between the vertices. The paper considers and describes algorithms for searching routes in depth and width. The conclusion is made about the preference of the search in depth. The results of its application are demonstrated in several examples.

Keywords: timestamps, file operations, computer forensics, transition digraph, breadth-first search, depth-first search.

Опыт проведения компьютерных исследований и экспертиз показывает, что в процессе их выполнения эксперту-криминалисту приходится решать задачи, связанные с восстановлением последовательности файловых операций (ФОп), совершенных пользователем ПЭВМ в отношении интересующих файлов, путем исследования метаданных файловой системы, в том числе временных отметок (ВО) файлов.

Анализ результатов цикла экспериментов [1, 2], проведенных в операционной системе (ОС) Windows, позволяет утверждать, что при совершении действий над файлами ВО изменяются по фиксированным алгоритмам, запрограммированным в ядре ОС.

С целью сопоставления возможных последовательностей ФОп наблюдаемым вариантам конечных состояний ВО файлов разработана модель изменения значений ВО [3], основанная на математическом аппарате теории автоматов. Модель наглядно демонстрирует, к каким состояниям ВО могут привести различные варианты последовательностей ФОп. На рис. 1 изображен пример моделирования процесса изменения ВО, осуществляемый при помощи таблицы переходов между состояниями ВО, где x_i — ФОп, s_j — состояния ВО. На рис. 1 представлен фрагмент таблицы переходов. Полная таблица построена на ос-

нове всего множества выявленных закономерностей изменения ВО [1, 2] и содержит 458 состояний [3]. Также в [3] приведена таблица соответствия x_i названиям ФОп.

В примере последовательное выполнение ФОп $x_6 \rightarrow x_8 \rightarrow x_{10}$ над файлом с начальным состоянием ВО s_0 привело к конечному состоянию ВО s_{50} :

$$s_0(x_6) \rightarrow s_6, s_6(x_8) \rightarrow s_8, s_8(x_{10}) \rightarrow s_{50}.$$

Под начальным состоянием будем считать состояние ВО файла s_0 , который был создан и над которым файловые операции еще не совершались.

В ходе проведения криминалистических исследований эксперт может получить только конечные состояния ВО файлов. Поэтому методика восстановления ФОп заключается в определении конечного состояния (известного эксперту), а затем поиске ФОп и состояний, которые могли привести к имеющемуся конечному состоянию ВО по таблице переходов (рис. 2). При этом некоторые состояния могут быть обнаружены в таблице переходов на пересечении нескольких ФОп и состояний ВО, что приведет к появлению различных вариантов последовательностей ФОп.

Задачу поиска промежуточных состояний можно решить, используя теорию графов в два этапа. Для этого на первом этапе по таблице переходов необходимо построить

$x(t)$	$S(t)$															
	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	...	s_{457}
x_1	s_1	s_1	s_{19}	s_1	s_1	s_1	s_{19}	s_{19}	s_{19}	s_1	s_{19}	s_{19}	s_{19}	s_{19}	...	s_{19}
x_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	...	s_2
x_3	s_3	s_3	s_3	s_3	s_3	s_{34}	s_3	s_{40}	s_3	s_3	s_{53}	s_{57}	s_{40}	...	s_{348}	
x_4	s_4	s_4	s_4	s_{22}	s_4	s_{35}	s_4	s_{41}	s_{49}	s_{22}	s_{22}	s_{54}	s_{58}	s_{41}	...	s_{431}
x_5	s_5	s_5	s_5	s_5	s_5	s_5	s_5	s_{42}	s_{42}	s_5	s_5	s_5	s_5	s_{42}	...	s_5
x_6	s_6	s_{12}	s_{20}	s_{23}	s_{30}	s_{23}	s_{38}	s_{43}	s_{38}	s_{51}	s_{43}	s_{38}	s_{59}	s_{62}	...	s_{211}
x_7	s_7	s_{13}	s_{13}	s_{24}	s_{24}	s_{24}	s_7	s_7	s_7	s_7	s_7	s_7	s_{13}	s_{13}	...	s_{24}
x_8	s_8	s_{14}	s_{14}	s_{25}	s_{31}	s_{25}	s_8	s_{44}	s_8	s_{44}	s_{44}	s_8	s_{14}	s_{63}	...	s_{90}
x_9	s_9	s_{15}	s_{21}	s_{26}	s_{26}	s_{36}	s_{39}	s_{45}	s_9	s_{39}	s_{55}	s_{60}	s_{64}	...	s_{457}	
x_{10}	s_0	s_{10}	s_{10}	s_{27}	s_{32}	s_{23}	s_0	s_{46}	s_{50}	s_{52}	s_{52}	s_{38}	s_{12}	s_{65}	...	s_{456}
x_{11}	s_{10}	s_{17}	s_{17}	s_{28}	s_{28}	s_{37}	s_{10}	s_{47}	s_{47}	s_{10}	s_{10}	s_{56}	s_{61}	s_{66}	...	s_{434}
x_{12}	s_{11}	s_{18}	s_{18}	s_{29}	s_{33}	s_{29}	s_{11}	s_{48}	s_{11}	s_{48}	s_{48}	s_{11}	s_{18}	s_{67}	...	s_{94}

Рис. 1. Моделирование процесса изменения ВО при выполнении последовательности ФОп $x_6 \rightarrow x_8 \rightarrow x_{10}$

$x(t)$	$S(t)$															
	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	...	s_{457}
x_1	s_1	s_1	s_{19}	s_1	s_1	s_1	s_{19}	s_{19}	s_{19}	s_1	s_{19}	s_{19}	s_{19}	s_{19}	...	s_{19}
x_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	s_2	...	s_2
x_3	s_3	s_3	s_3	s_3	s_3	s_{34}	s_3	s_{40}	s_3	s_3	s_{53}	s_{57}	s_{40}	...	s_{348}	
x_4	s_4	s_4	s_4	s_{22}	s_4	s_{35}	s_4	s_{41}	s_{49}	s_{22}	s_{22}	s_{54}	s_{58}	s_{41}	...	s_{431}
x_5	s_5	s_5	s_5	s_5	s_5	s_5	s_5	s_{42}	s_{42}	s_5	s_5	s_5	s_5	s_{42}	...	s_5
x_6	s_6	s_{12}	s_{20}	s_{23}	s_{30}	s_{23}	s_{38}	s_{43}	s_{38}	s_{51}	s_{43}	s_{38}	s_{59}	s_{62}	...	s_{211}
x_7	s_7	s_{13}	s_{13}	s_{24}	s_{24}	s_{24}	s_7	s_7	s_7	s_7	s_7	s_7	s_{13}	s_{13}	...	s_{24}
x_8	s_8	s_{14}	s_{14}	s_{25}	s_{31}	s_{25}	s_8	s_{44}	s_8	s_{44}	s_{44}	s_8	s_{14}	s_{63}	...	s_{90}
x_9	s_9	s_{15}	s_{21}	s_{26}	s_{26}	s_{36}	s_{39}	s_{45}	s_9	s_{39}	s_{55}	s_{60}	s_{64}	...	s_{457}	
x_{10}	s_6	s_{16}	s_{16}	s_{27}	s_{32}	s_{23}	s_6	s_{46}	s_{50}	s_{52}	s_{52}	s_{38}	s_{12}	s_{65}	...	s_{456}
x_{11}	s_{10}	s_{17}	s_{17}	s_{28}	s_{28}	s_{37}	s_{10}	s_{47}	s_{47}	s_{10}	s_{10}	s_{56}	s_{61}	s_{66}	...	s_{434}
x_{12}	s_{11}	s_{18}	s_{18}	s_{29}	s_{33}	s_{29}	s_{11}	s_{48}	s_{11}	s_{48}	s_{48}	s_{11}	s_{18}	s_{67}	...	s_{94}

Рис. 2. Определение ФОп и состояния ВО, на пересечении которых находится конечное состояние ВО s_{50}

граф переходов [4], где вершины — состояния s_i , а ребра — операции x_j , производимые над файлом, петли свидетельствуют об операциях, возможно, производившихся над файлом неопределенное количество раз. На втором этапе в построенном орграфе переходов произвести полный перебор всех возможных маршрутов между конечной вершиной s_i , представляющей конечное состояние ВО, и начальной вершиной s_0 , соответствующей начальному состоянию ВО.

Построенный на первом этапе граф является однонаправленным (оргграф) и наглядно представляет переходы между состояниями s_i . Так, переходу к состоянию s_{50} , представленному на рис. 1, в графическом виде соответствует оргграф, приведенный на рис. 3.

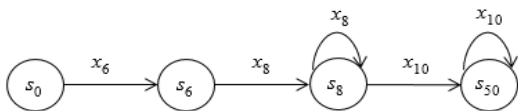


Рис. 3. Оргграф перехода из состояния s_0 в состояние s_{50}

Исчерпывающий оргграф переходов (рис. 4) построен по полной таблице переходов в программной среде MATLAB с примени-

ем функции digraph [5]. В качестве входных аргументов функции использовалась матрица смежности, сгенерированная на основе таблицы переходов. Сформированный таким образом оргграф имеет 458 вершин и 3654 ребра.



Рис. 4. Оргграф переходов, построенный по полной таблице переходов

На втором этапе для полного перебора всех возможных маршрутов между вершинами можно применять широко распространенные алгоритмы обхода графов: поиск

в ширину и глубину. Обход графа — это переход от одной его вершины к другой в поисках связей этих вершин, где в качестве связей имеются в виду ребра графа. В процессе обхода вершины могут находиться в трех состояниях: непросмотренные, просмотренные, использованные. Изначально все вершины имеют статус непросмотренных.

Алгоритм поиска в ширину (breadth-first search, BFS) подразумевает поуровневое исследование графа. Для описания поиска в ширину вводится очередь Q для хранения вершин. Поиск начинается с некоторой начальной вершины u . Эта вершина помещается в очередь Q и с этого момента считается *просмотренной*. Смежные с u вершины u_1, u_2, \dots, u_p , помещаются в очередь и получают статус *просмотренных*, а вершина u удаляется из очереди и получает статус *использованной*. Далее смежные с u_1, u_2, \dots, u_p вершины помещаются в очередь и получают статус *просмотренных*, а вершины u_1, u_2, \dots, u_p удаляются из очереди и получают статус *использованных*. Вершины просматриваются в порядке возрастания их расстояния от началь-

ной вершины. В тот момент, когда очередь Q окажется пустой, поиск в ширину обойдет все вершины графа [6].

При поиске в глубину (depth-first search, DFS) производится перечисление вершин «вглубь», пока это возможно. Поиск начинается с некоторой начальной вершины u и с этого момента, она считается *просмотренной*. Если среди вершин, смежных с u , существует еще непросмотренная вершина w , тогда w объявляется *просмотренной*, и поиск продолжается из этой вершины. Если все вершины, смежные с u , просмотрены, тогда u объявляется *использованной* вершиной. Если в графе не осталось непросмотренных вершин, то поиск заканчивается. Если же осталась непросмотренная вершина, то поиск продолжается из нее [6].

На рис. 5 представлен процесс обхода орграфа алгоритмами поиска в ширину (а) и в глубину (б) из вершины s_{50} . Данные алгоритмы были реализованы в MATLAB с помощью функций `bfsearch` [7] и `dfsearch` [8] соответственно.

На рис. 5 сплошной линией выделены ре-

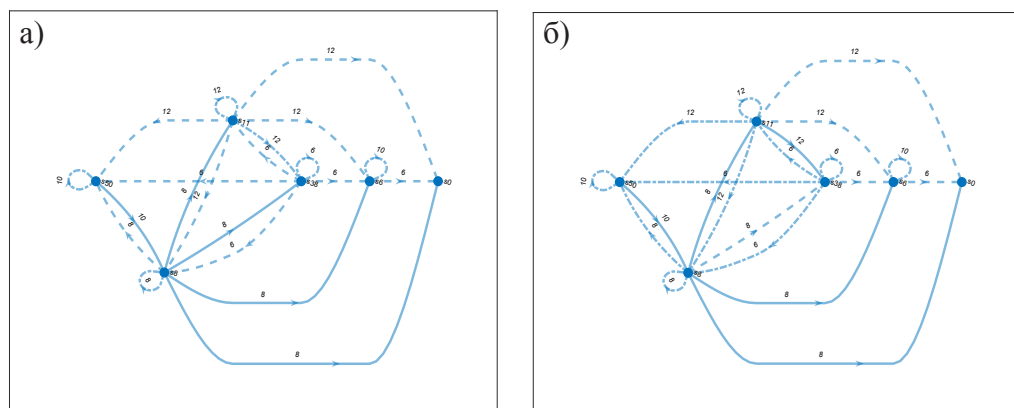


Рис. 5. Процесс обхода орграфа алгоритмами поиска в ширину и в глубину

бра, которые ведут к непросмотренным вершинам; штрихпунктирной линией — ребра, которые ведут к просмотренным вершинам; штриховой линией — ребра, которые ведут к использованным вершинам. Пошаговый процесс обхода в ширину и глубину графа представлен в таблицах 1 и 2 соответственно.

Оба алгоритма приводят к идентичному результату, имеют одинаковую трудоемкость и позволяют обнаруживать все маршруты, соединяющие две вершины: начальную и конечную. При этом поиск в ширину зачастую используется для нахождения кратчайшего маршрута между точками. Поиск в глубину

применяется для определения любого возможного маршрута между двумя вершинами и обнаружения циклов на графе.

Следует обратить внимание, что специфика орграфа переходов между состояниями ВО такова, что оба алгоритма приводят к появлению циклов. С точки зрения восстановления последовательности ФOp циклы и петли интерпретируются как повторение некоторых ФOp неопределенное количество раз. С одной стороны для полноты картины эту информацию надо учитывать. С другой стороны, в ходе компьютерного исследования следствие интересует в первую очередь по-

Пошаговый процесс обхода в ширину

№ шага	Вершины, находящиеся в очереди Q	Статус вершины	Смежные вершины	Статус смежной вершины	Ребро, соединяющее вершины из очереди и смежную	Тип линии на рис. 5, а
1	s50	просмотренная				
2	s50		s8	непросмотренная	s50s8	сплошная
3	s50	использованная				
4	s8	просмотренная				
5			s0	непросмотренная	s8s0	сплошная
6			s6	непросмотренная	s8s6	сплошная
7			s11	непросмотренная	s8s11	сплошная
8			s38	непросмотренная	s8s38	сплошная
9			s50	использованная	s8s50	штриховая
10	s8	использованная				
11	s0	просмотренная				
12	s6	просмотренная				
13	s11	просмотренная				
14	s38	просмотренная				
15	s0		—			
16	s0	использованная				
17	s6		s0	использованная	s6s0	штриховая
18	s6	использованная				
19	s11		s0	использованная	s11s0	штриховая
20			s6	использованная	s11s6	штриховая
21			s8	использованная	s11s8	штриховая
22			s38	просмотренная	s11s38	штрихпунктирная
23			s50	использованная	s11s50	штриховая
24	s11	использованная				
25	s38		s6	использованная	s38s6	штриховая
26			s8	использованная	s38s8	штриховая
27			s11	использованная	s38s11	штриховая
28			s50	использованная	s38s50	штриховая
29	s38	использованная				

Таблица 2

Пошаговый процесс обхода в глубину

№ шага	Рассматриваемые вершины	Статус вершины	Смежные вершины	Статус смежной вершины	Ребро, соединяющее вершины из очереди и смежную	Тип линии на рис. 5, б
1	s50	просмотренная				
2	s50		s8	непросмотренная	s50s8	сплошная
3	s8	просмотренная				

№ шага	Рассматриваемые вершины	Статус вершины	Смежные вершины	Статус смежной вершины	Ребро, соединяющее вершины из очереди и смежную	Тип линии на рис. 5, 6
4	s8		s0	непросмотренная	s8s0	сплошная
5	s0	просмотренная				
6	s0		—			
7	s0	использованная				
8	s8		s6	непросмотренная	s8s6	сплошная
9	s6	просмотренная				
10	s6		s0	использованная	s6s0	штриховая
11	s6	использованная				
12	s8		s11	непросмотренная	s8s11	сплошная
13	s11	просмотренная				
14	s11		s0	использованная	s11s0	штриховая
15			s6	использованная	s11s6	штриховая
16			s8	просмотренная	s11s8	штрихпунктирная
17			s38	непросмотренная	s11s38	сплошная
18	s38	просмотренная				
19	s38		s6	использованная	s38s6	штриховая
20			s8	просмотренная	s38s8	штрихпунктирная
21			s11	просмотренная	s38s11	штрихпунктирная
22			s50	просмотренная	s38s50	штрихпунктирная
23	s38	использованная				
24	s11		s50	просмотренная	s11s50	штрихпунктирная
25	s11	использованная				
26	s8		s38	использованная	s8s38	штриховая
27			s50	просмотренная	s8s50	штрихпунктирная
28	s8	использованная				
29	s50	использованная				

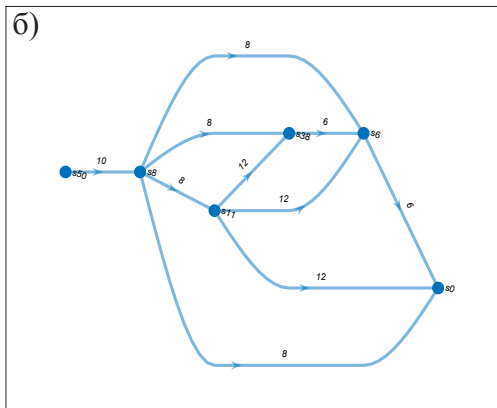
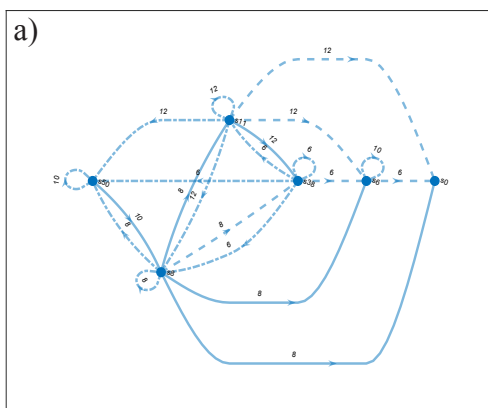


Рис. 6. Процесс обхода орграфа алгоритмом поиска в глубину с циклами и петлями (а) и без циклов и петель (б)

следовательность и факт выполнения ФОп, а не количество повторения однотипных операций. Поэтому, чтобы избежать избыточности информации циклы и петли из графа целесообразно исключать, что возможно при применении поиска в глубину. Для этого исключаются ребра, которые ведут к просмотренным вершинам (штрихпунктирная линия). На рис. 6, б изображен оргграф процесса обхода в глубину без циклов и петель.

Оргграф, построенный на рис. 6, б наглядно демонстрирует, какие ФОп могли быть совершены над исследуемым файлом. При этом последовательность ФОп восстанавливается от последней выполненной над файлом операции к первой. Между конечной и начальной вершинами могут быть обнаружены несколько маршрутов, что приведет к выявлению различных возможных вариантов последовательностей ФОп. Так, для примера, приведенного на рис. 6, б между вершинами, соответствующими состояниям ВО s_{50} и s_0 , возможны следующие варианты маршрутов:

$$\begin{aligned}
 & s_{50}(x_{10}) \rightarrow s_{8'} s_8(x_8) \rightarrow s_{6'} s_6(x_6) \rightarrow s_0. \\
 & s_{50}(x_{10}) \rightarrow s_{8'} s_8(x_8) \rightarrow s_{38'} s_{38}(x_6) \rightarrow s_{6'} s_6(x_6) \rightarrow s_0. \\
 & s_{50}(x_{10}) \rightarrow s_{8'} s_8(x_8) \rightarrow s_{11'} s_{11}(x_{12}) \rightarrow s_{38'} s_{38}(x_6) \rightarrow s_{6'} s_6(x_6) \rightarrow s_0. \\
 & s_{50}(x_{10}) \rightarrow s_{8'} s_8(x_8) \rightarrow s_{11'} s_{11}(x_{12}) \rightarrow s_{6'} s_6(x_6) \rightarrow s_0. \\
 & s_{50}(x_{10}) \rightarrow s_{8'} s_8(x_8) \rightarrow s_{11'} s_{11}(x_{12}) \rightarrow s_0. \\
 & s_{50}(x_{10}) \rightarrow s_{8'} s_8(x_8) \rightarrow s_0.
 \end{aligned} \quad (1)$$

В (1) все варианты маршрутов начинаются с операций x_{10} и x_8 . Таким образом, для состояния ВО s_{50} однозначно оказываются определены последняя ФОп (x_{10}) и предпоследняя ФОп (x_8).

Ниже представлены примеры процесса обхода оргграфа из вершин, соответствующих состояниям ВО s_{32} и s_{71} .

Пример на рис. 7, а также демонстрируют однозначное определение последней и предпоследней ФОп, что очевидно следует из особенностей таблицы переходов: состояния ВО повторяются только по горизонтали для всех ФОп, за исключением x_6 (перемещение/переименование) и x_{10} (изменение атрибутов) [3].

Для уменьшения количества вариантов возможных маршрутов можно произвести их постобработку путем исключения некоторых ребер и вершин по имеющимся исходным условиям. Например, если известно, что все операции производились в ОС Windows 7, тогда можно исключить ребра, соответствующие ФОп, производимых в других версиях ОС Windows. В примере на рис. 7, б можно исключить все ребра, соответствующие ФОп $x_{10'}$, в результате количество маршрутов уменьшится на 6.

Таким образом, решение задачи по восстановлению последовательности ФОп сводится к выполнению двух этапов, которые необходимо произвести один раз. Сначала следует сформировать оргграф переходов по полной таблице переходов (рис. 4), который является универсальным для 12 ФОп в ОС Windows [1, 2]. Затем для всех вершин оргграфа (состояний ВО) произвести поиск в глубину и исключить циклы и петли.

Данный процесс был запрограммирован в MATLAB. Для имеющейся таблицы переходов построение оргграфа и расчет маршрутов для всех вершин заняло 8 минут на ПЭВМ «ПК Аквариус» (Intel(R) Core(TM) i5-4670 CPU @ 3,40GHz, 3,40 GHz, ОЗУ 16,00 Гб). В результате была сформирована таблица, на вход которой эксперт может подавать конечное состояние ВО исследуемого файла s_r , а на выходе получать набор возможных вариантов после-

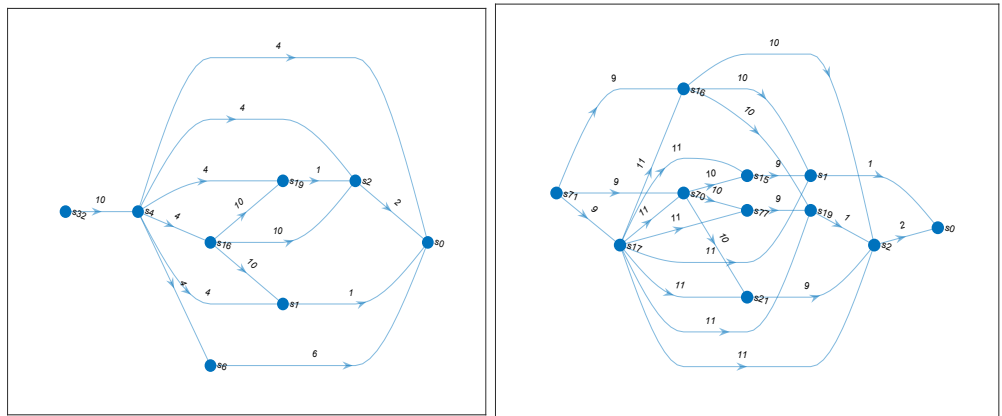


Рис. 7. Процесс обхода оргграфа алгоритмом поиска в глубину из вершин, соответствующих состояниям ВО s_{32} (а) и s_{71} (б)

довательностей ФOp. Так как последовательности ФOp заранее посчитаны и помещены в таблицу, их формирование на выходе программы происходит практически мгновенно.

Для эксперта этот процесс осуществляется «прозрачно». Ему остается только исключить явно невозможные ФOp по известным дополнительным условиям.

Литература

1. Духан Е.И., Князева Н.С. Методика и результаты исследования изменений временных отметок файловых объектов. // Радиотехника. 2020. Том 84, № 2 (4). С. 64-72.
2. Knyazeva N., Khorkov D., Vostretsova E. Building Knowledge Bases for Timestamp Changes Detection Mechanisms in MFT Windows OS. // Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). 2020. С. 553-556.
3. Knyazeva N., Duhan E. Timestamp Change Model in Windows OS. // Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). 2020. С. 623-626.
4. Советов Б.Я. Моделирование систем: Учебник для вузов / Б. Я. Советов, С. А. Яковлев. 5-е изд. М.: Высш. шк., 2007. 343 с.
5. Graph with directed edges – MATLAB [Электронный ресурс] // URL: <https://ch.mathworks.com/help/matlab/ref/digraph.html> (дата обращения: 01.04.2021).
6. Асанов М.О., Баранский В.А., Расин В.В. Дискретная математика: графы матроиды, алгоритмы // Ижевск: НИЦ «РХД». 2001. 288 с.
7. Breadth-first graph search – MATLAB bsearch [Электронный ресурс] // URL: <https://ch.mathworks.com/help/matlab/ref/graph.bsearch.html> (дата обращения: 01.04.2021).
8. Depth-first graph search – MATLAB dfsearch [Электронный ресурс] // URL: <https://ch.mathworks.com/help/matlab/ref/graph.dfsearch.html> (дата обращения: 01.04.2021).

References

1. Duhan E., Knyazeva N. Methodology and results of the study of changes in the timestamps of file objects [Metodika i rezul'taty issledovanija izmenenij vremennyh otmetok fajlovyh obektov]. // Radio engineering [Radiotekhnika]. 2020. Vol. 84, no 2 (4). pp. 64-72.
2. Knyazeva N., Khorkov D., Vostretsova E. Building Knowledge Bases for Timestamp Changes Detection Mechanisms in MFT Windows OS. // Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). 2020. pp. 553-556.
3. Knyazeva N., Duhan E. Timestamp Change Model in Windows OS. // Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). 2020. pp. 623-626.
4. Sovetov B.Ya. Simulation of systems: Textbook for universities [Modelirovanie system: Uchebnik dlja vuzov] / B.Ya. Sovetov, S.A. Yakovlev 5-ye izd. M.: High school [Vysshaja shkola], 2007. 343 p.
5. Graph with directed edges – MATLAB. Available at: <https://ch.mathworks.com/help/matlab/ref/digraph.html> (accessed 01 April 2021).
6. Asanov M.O., Baranskij V.A., Rasin V.V. Discrete mathematics: graphs, matroids, algorithms [Diskretnaja matematika: grafy matroidy, algoritmy] // Izhevsk: NIC «RHD». 2001. 288 p.
7. Breadth-first graph search – MATLAB bsearch. Available at: <https://ch.mathworks.com/help/matlab/ref/graph.bsearch.html> (accessed 01 April 2021).
8. Depth-first graph search – MATLAB dfsearch. Available at: <https://ch.mathworks.com/help/matlab/ref/graph.dfsearch.html> (accessed 01 April 2021).

КНЯЗЕВА Наталия Сергеевна, старший преподаватель учебно-научного центра «Информационная безопасность», Уральский Федеральный Университет им. первого Президента России Б.Н. Ельцина. 620002, г. Екатеринбург, ул. Мира, 32. E-mail: npalceva@inbox.ru.

KNYAZEVA Natalija, Senior lecturer of educational and scientific center «Information security», Ural Federal University named after the first President of Russia B.N.Yeltsin, 620002, Yekaterinburg, Mira str., 32. E-mail: npalceva@inbox.ru.