

УДК 519.7

©2010. И.С. Грунский, С.В. Сапунов

ИДЕНТИФИКАЦИЯ ВЕРШИН ПОМЕЧЕННЫХ ГРАФОВ

Рассматривается задача определения мобильным агентом своего положения в среде моделируемой графом с помеченными вершинами. Агент может перемещаться по дугам графа и наблюдать метки вершин. Введены конечные множества слов в алфавите меток, отличающие одну вершину графа от всех других его вершин, названные ее идентификаторами. Найдены условия существования, оценки сложности идентификаторов и разработаны методы их построения. Разработаны полиномиальные методы построения и проведения экспериментов по определению начальной вершины графа, основанные на построении идентификаторов всех его вершин.

Ключевые слова: граф с помеченными вершинами, идентификатор вершины, диагностический эксперимент.

1. Введение. Развитие теории формальных языков и конечных автоматов вызвало интерес к графам с помеченными элементами (вершинами, дугами, точками "прикосновения" дуги к вершине и т.п.) [1, 2]. Результаты исследований таких графов получили широкий спектр приложений. В частности, графы с помеченными элементами (помеченные графы) оказались удобным средством для исследования топологических операционных сред мобильных агентов (роботов, поисковых программ и т.п.) [3]. При планировании двигательного поведения мобильных агентов (МА) возникают три взаимосвязанные фундаментальные задачи: задача построения модели неизвестной среды (exploration), задача определения положения робота в известной среде (robot selflocation) и задача проверки соответствия неизвестной среды и ее модели (карты) (map validation) [4]. Заметим, что эти задачи имеют соответствующие аналоги в теории автоматов [5].

Центральной из перечисленных задач является задача самолокализации, так как от ее решения зависит решение двух других задач. Состоит она в следующем: МА, обладая картой (помеченным графом) среды, должен установить соответствие между вершиной на карте и неизвестной ему априори областью среды, в которой он первоначально находился. При изучении задачи самолокализации возник ряд подходов, зависящих от того, какие элементы графа агент наблюдает и метки каких элементов он может изменить. В работах [3, 4] рассматривается задача самолокализации МА, у которого операционная среда моделируется конечным неориентированным графом с помеченными концами ребер. МА, находясь в некоторой вершине графа, единственным образом нумерует концы ребер, инцидентных этой вершине. МА не отличает вершины графа друг от друга. Если существует путь из одной вершины графа в другую его вершину, то он описывается как последовательность пар номеров попарно смежных ребер. Авторы предлагают алгоритм определения начальной вершины известного n -вершинного графа за $O(n^5)$ шагов при условии, что робот использует один переносной маркер. Основная идея алгоритма состоит в построении $O(n^2)$ гипотез о начальной вершине и разметке инцидентных ей ребер. Далее, МА проверяет

каждую гипотезу за время $O(n^3)$. При этом он восстанавливает граф из начальной вершины, параллельно сравнивая восстановленную часть графа с известной ему моделью. Недостатком предложенных модели и алгоритма самолокализации является невозможность однозначного определения положения робота на графе с симметрией (нетривиальным автоморфизмом).

В данной работе в качестве топологической модели операционной среды рассматриваются конечные ориентированные графы. Вершины этих графов заранее помечены и эти метки агент не меняет. Такая модель возникает, например, при качественной навигации МА (qualitative navigation) [8]. Основой такой навигации служат ориентиры, которые разбивают линиями действия, каждая из которых проходит через два ориентира, операционную среду на выпуклые многоугольники, являющиеся областями информационной эквивалентности. Каждая такая область является, в свою очередь, вершиной скелетного графа, а общая сторона двух многоугольников порождает ребро между соответствующими вершинами скелетного графа. Если каждой вершине сопоставить упорядоченную по углу наблюдения последовательность ориентиров, то получим неориентированный граф с помеченными вершинами.

2. Основные определения. Конечным ориентированным графом с помеченными вершинами (помеченным орграфом) назовем четверку $G = (G, E(G), M, \mu_G)$, где G – конечное множество вершин, $|G| = n$, $E(G) \subseteq G \times G$ – конечное множество ребер, M – алфавит меток, $|M|, \mu_G : G \rightarrow M$ – сюръективная функция разметки. Последовательность меток вершин $w = \mu_G(g_1) \dots \mu_G(g_k)$, соответствующую некоторому пути $g_1 \dots g_k$ в графе G , назовем словом длины k , порожденным вершиной g_1 . Через $d(w)$ обозначим длину слова w . Инверсией слова $w = x_1 \dots x_k$ назовем слово $w^{rev} = x_k \dots x_1$. Слово v называется подсловом слова w , если существуют слова u_1 и u_2 (возможно пустые) такие, что $w = u_1 v u_2$. Если u_1 пусто, то слово v также называется начальным отрезком или префиксом $\text{pre}_k(w)$ слова w длины k , где $k = d(v)$. Обозначим через M^+ множество всех непустых слов в алфавите M . Пусть $W \subseteq M^+$. Высотой W назовем наибольшую из длин, входящих в него слов, его кратностью – величину $|W|$, его объемом – сумму длин всех входящих в него слов.

Для слов $u, w \in M^+$ введем их композицию $u \circ w$ равную uw' , если $u = u'x$, $w = xw'$, $x \in M$, и не определено в противном случае. Введем частичную операцию $\star : G \times M^+ \rightarrow 2^G$ соотношением: для любой вершины $g \in G$ и любого слова $w \in M^+$ через $g \star w$ обозначим множество всех вершин $h \in G$ таких, что существует путь из g в h , помеченный словом $w = \mu(g) \circ w \circ \mu(h)$. Распространим операцию \star на подмножества множества G естественным образом: если $G' \subseteq G$, то $G' \star w = \bigcup_{g \in G'} (g \star w)$.

Языком L_g вершины g назовем множество всех слов, порожденных этой вершиной. Будем говорить, что вершины $g, h \in G$ неотличимы, т.е. $(g, h) \in \varepsilon$, если $L_g = L_h$. Отношение ε рефлексивно, симметрично и транзитивно, т.е. является отношением эквивалентности. Фактор-граф G/ε графа G по отношению ε назовем приведенным графом. Всякому множеству вершин $G' \subseteq G$ поставим в соответствие множество слов $L_{G'} = \bigcup_{g \in G'} L_g$.

Помеченный орграф G назовем детерминированным орграфом или D-орграфом,

если для любой вершины $g \in G$ и любых вершин $s, t \in \Gamma_{(g)}^-$ из $s \neq t$ следует, что $\mu(s) \neq \mu(t)$ (здесь $\Gamma_{(g)}^-$ обозначает множество всех вершин, являющихся концами дуг, исходящих из g). В противном случае G назовем ND-орграфом. В [9] показано, что оценка длины кратчайшего слова, различающего две вершины D-орграфа равна $n - m + 1$ и совпадает с верхней оценкой длины кратчайшего слова, различающего состояния конечного всюду определенного детерминированного автомата Мура [6]. Простой D-граф G назовем сильно детерминированным или SD-графом, если для него выполняются следующие ограничения: 1) для любых вершин $g, h \in G$ если $(g, h) \in E(G)$, то $(h, g) \in E(G)$; 2) для любой вершины $g \in G$ и любых вершин $s, t \in O_{(g)}$ из $s \neq t$ следует, что $\mu(s) \neq \mu(t)$ (здесь $O_{(g)} = \Gamma_{(g)}^- \cup \{g\}$).

3. Эксперименты с графами. Экспериментом с графом G относительно априорной информации \mathbf{I} , цели \mathbf{C} и средств \mathbf{S} назовем процесс, состоящий из трех этапов: 1) построение теста $P \subset M^+$ на основе \mathbf{I} и \mathbf{C} ; 2) получение мобильным агентом (МА) экспериментальных данных \mathbf{W} на основе P и \mathbf{S} ; 3) вывод заключений о свойствах графа на основе \mathbf{W} и \mathbf{I} . Априорная информация - это класс графов, к которому принадлежит G . Основное отличие экспериментов с графами от экспериментов с автоматами [5] состоит в методах и средствах \mathbf{S} получения экспериментальных данных, а именно, возможностях МА перемещаться по графу, воспринимать локальную информацию о вершинах и ставить в них дополнительные стираемые/нестираемые метки.

Эксперимент назовем диагностическим, если априори полностью известен граф G и МА установлен в произвольную начальную вершину этого графа, а целью является определение этой вершины, то есть отличие этой вершины от всех других вершин. Допуская вольность речи, множество \mathbf{W} будем также называть экспериментом и говорить, что эксперимент \mathbf{W} порождается тестом P .

Будем рассматривать следующие меры сложности экспериментов. Операционной кратностью эксперимента назовем величину, равную, в зависимости от способа проведения эксперимента, либо количеству экземпляров МА, использованных при получении \mathbf{W} , либо количеству установок МА экспериментатором в начальную вершину исследуемого графа. Если используется только один экземпляр МА (одна установка), то эксперимент назовем простым. Операционной высотой эксперимента назовем наибольшую из длин путей, которые проходит каждый экземпляр МА по исследуемому графу в ходе эксперимента.

Из определения эксперимента следует, что он осуществляется посредством прохождения МА некоторого связанного с тестом P множества путей по графу G из вершины g . Будем полагать, что, находясь в любой вершине $h \in G$, МА воспринимает множество слов $Ob^+(h) = M^k \cap L_h$, где $M^k \subseteq M^+$ является множеством всех слов длины не больше k в алфавите меток M , и вычисляет множество слов $Ob^-(h) = M^k - L_h = M^k - Ob^+(h)$. При $k = 1$ полагаем, что МА воспринимает только метку текущей вершины. Таким образом, проходя путь $g_1 \dots g_l$ в графе G , МА воспринимает множество слов $Ob^+(g_1 \dots g_l) = \bigcup_{i=1}^l \mu(g_1) \dots \mu(g_i) \circ Ob^+(g_i)$ и вычисляет множество слов $Ob^-(g_1 \dots g_l) = \bigcup_{i=1}^l \mu(g_1) \dots \mu(g_i) \circ Ob^-(g_i)$. В дальнейшем, если не оговорено противное, будем считать, что $k = 2$.

Пусть $w \in M^+$. Обозначим через $\delta_g(w)$ множество всех путей из вершины g , которые помечены словом w . По произвольному множеству $P \subseteq M^+$ и $g \in G$ построим пару $\overline{W}_g = \{W_g^+, W_g^-\}$, где $W_g^+ = \bigcup_{w \in [P]} Ob^+(\delta_g(w))$ и $W_g^- = \bigcup_{w \in [P]} Ob^-(\delta_g(w))$, а $[P]$ является замыканием множества P по всем непустым начальным отрезкам слов из P . Экспериментом W с графом G , порожденным тестом P , назовем семейство $\{\overline{W}_g\}_{g \in G}$. Эксперимент W назовем диагностическим точно тогда, когда для любых вершин $g, h \in G$ из $g \neq h$ вытекает $\overline{W}_g \neq \overline{W}_h$, то есть $W_g^+ \neq W_h^+$ или $W_g^- \neq W_h^-$. В этом случае тест P также назовем диагностическим.

4. Идентификаторы вершин. Идентификатором вершины $g \in G$ назовем конечное множество слов $W_g \subseteq M^+$ такое, что для любой вершины $h \in G$ равенство $W_g \cap L_g = W_g \cap L_h$ выполняется тогда и только тогда, когда $g = h$.

Ясно, что всякий идентификатор W_g можно представить как объединение $W_g^+ \cup W_g^-$ двух множеств $W_g^+ = W_g \cap L_g$ и $W_g^- = W_g - L_g$. Идентификатор W_g будем называть *позитивным*, если $W_g^- = \emptyset$ и *негативным*, если $W_g^+ = \emptyset$.

Рассмотрим условия существования идентификаторов вершин D-графов.

Теорема 1. *Равносильны следующие утверждения:*

1. Существует идентификатор вершины $g \in G$.
2. Существует идентификатор вершины $g \in G$ высоты не больше $n - m + 1$ и кратности не больше $n - 1$.
3. $\varepsilon(g) = \{g\}$.

Доказательство. Импликация $2 \Rightarrow 1$ очевидна.

Покажем, что из утверждения 1 следует утверждение 3. Пусть W_g — идентификатор вершины g , h — некоторая вершина и $h \in \varepsilon(g)$. Из последнего соотношения следует, что $L_g = L_h$ и тогда $W_g \cap L_g = W_g \cap L_h$. По определению идентификатора $g = h$, то есть $\varepsilon(g) = \{g\}$.

Покажем далее, что из утверждения 3 следует утверждение 2. Пусть $G - \{g\} = \{g_1, \dots, g_{n-1}\}$. Так как $\varepsilon(g) = \{g\}$, то для любой вершины g_i , $1 \leq i \leq n-1$, существует слово $w_i \in L_g \oplus L_{g_i}$. Пусть $W = \{w_1, \dots, w_{n-1}\}$. Очевидно, что для любой вершины g_i выполняется $L_g \cap W \neq L_{g_i} \cap W$, то есть W является идентификатором g . Ясно, что $|W| = n - 1$. По [9] длина слова w_i не превосходит $n - m + 1$.

Теорема доказана. \square

Рассмотрим алгоритм построения идентификаторов всех вершин произвольного D-графа G . Основная идея алгоритма состоит в построении последовательности $\pi_1, \pi_2, \dots, \pi_k$ разбиений множества вершин G до тех пор, пока $\pi_k = \pi_{k+1}$. Одновременно для каждого разбиения $\pi_i = \{K_1, K_2, \dots, K_r\}$, $r \leq n$, строится семейство $\{W_1, W_2, \dots, W_r\}$ множеств слов в алфавите M так, что каждому классу $K_j \in \pi_i$, $1 \leq j \leq r$, соответствует множество $W_j \subseteq M^+$. Финальному разбиению π_k соответствует семейство идентификаторов всех вершин графа G .

Алгоритм 1. ПОСТРОЕНИЕ ИДЕНТИФИКАТОРОВ ВЕРШИН

Вход. D-граф $G = (G, E, M, \mu)$.

Выход. Семейство $\{W_{g_1}, \dots, W_{g_n}\}$ идентификаторов всех вершин графа G .

Метод.

1. Построим разбиение $\pi_1 = \{X_1, X_2, \dots, X_m\}$, каждый класс которого состоит из всех одинаково помеченных вершин, то есть $X_i = G \star x_i$, где $1 \leq i \leq m$, $x_i \in M$. Инициализируем множество АК активных классов и положим $AK = \pi_1$. С каждым классом $X_i \in \pi_1$ свяжем множество слов W_{X_i} и положим $W_{X_i} = \{x_i\}$.

2. Положим $k = 1$.

3. Положим $\pi_{k+1} = \pi_k$.

4. Если $AK = \emptyset$ и $\pi_k = \pi_{k+1}$, то финальное разбиение $\pi_k = \{K_1, K_2, \dots, K_r\}$, $r \leq n$, уже построено. Для каждого класса $K_i \in \pi_k$, $1 \leq i \leq r$, и для каждой вершины $g \in K_i$ положим $W_g = W_{K_i}$. Алгоритм прекращает работу.

5. Если $AK = \emptyset$ и $\pi_k \neq \pi_{k+1}$, то положим $k = k + 1$.

6. Если $AK \neq \emptyset$, то выберем класс $B \in AK$ и положим $AK = AK \setminus B$.

7. Положим $B' = \emptyset$, $U' = \emptyset$, $U'' = \emptyset$.

8. Для каждой метки $x \in M$ такой, что существует вершина $g \in B$, для которой выполняется $g \star \mu(B) x = \emptyset$, положим $U' = U' \cup \{x\}$.

9. Положим $U'' = M \setminus U'$.

10. Если $U' = \emptyset$, то перейдем к 16.

11. Выберем метку $x \in U'$ и положим $U' = U' \setminus \{x\}$.

12. Для каждой вершины $g \in B$ такой, что $g \star \mu(B) x = \emptyset$, положим $B' = B' \cup \{g\}$.

13. Если $B' = B$ и $U' \neq \emptyset$, то положим $B' = \emptyset$ и возвратимся к 11.

14. Если $B' = B$ и $U' = \emptyset$, то перейдем к 16.

15. Положим $B'' = B \setminus B'$, $W_{B'} = W_{B''} = W_B \cup \{\mu(B) x\}$ и перейдем к 23.

16. Если $U'' = \emptyset$, то возвратимся к 4.

17. С каждой меткой $x \in U''$ свяжем класс K_x такой, что $\mu(K_x) = x$, $B \star \mu(B) x \cap K_x \neq \emptyset$ и высота множества W_{K_x} наименьшая из возможных.

18. Выберем метку $x \in U''$, у которой высота W_{K_x} наименьшая, и положим $U'' = U'' \setminus \{x\}$.

19. Для каждой вершины $g \in B$ такой, что $g \star \mu(B) x = K_x$, положим $B' = B' \cup \{g\}$.

20. Если $B' = B$ и $U'' \neq \emptyset$, то положим $B' = \emptyset$ и возвратимся к 18.

21. Если $B' = B$ и $U'' = \emptyset$, то возвратимся к 4.

22. Положим $B'' = B \setminus B'$, $W_{B'} = W_{B''} = W_B \cup \mu(B) W_{K_x}$.

23. Положим $\pi_{k+1} = (\pi_{k+1} \setminus B) \cup B' \cup B''$ и возвратимся к 4.

Анализ алгоритма 1 начнем со следующей теоремы.

Теорема 2. Алгоритм 1 строит семейство $\{W_g\}_{g \in G}$ идентификаторов всех вершин произвольного D -графа G , для которых существует идентификатор.

Доказательство. Алгоритм останавливается тогда, когда все классы разбиения неактивны. Класс становится неактивным, если он не разделился на два подкласса. Поэтому алгоритм остановится, по крайней мере, после того, как будет получено разбиение, состоящее из одноэлементных классов.

Покажем, что финальное разбиение, построенное алгоритмом 3.1, является ε -эквивалентным разбиением множества G . Пусть $g, h \in G$, $\mu(g) = \mu(h)$ и $(g, h) \notin \varepsilon$. Тогда существует кратчайшее слово wxy , где $w \in M^*$, $x, y \in M$, такое, что $wx \in L_g \cap L_h$ и $wxy \in L_g \oplus L_h$. Так как одна из вершин $g \star wxy$ или $h \star wxy$ не существует, то на

некоторой итерации алгоритма 1 вершины $g \star wx$ и $h \star wx$ попадут в разные классы. Тогда на некоторой следующей итерации вершины $g \star w$ и $h \star w$ также попадут в разные классы. Далее, для любого k , $1 < k < d(w)$, вершины $g \star pre_k(w)$ и $h \star pre_k(w)$ будут попадут в разные классы на соответствующих итерациях алгоритма. Так как вершины $g \star pre_2(w)$ и $h \star pre_2(w)$ находятся в разных классах, то по окончании работы алгоритма вершины g и h также находятся в разных классах финального разбиения. Так как вершины g и h были выбраны произвольно, то тем самым доказано, что финальное разбиение, построенное алгоритмом 1 является ε -эквивалентным разбиением множества G .

Покажем, что по окончании работы алгоритма будет получено семейство идентификаторов всех вершин графа G , то есть для любой вершины $g \in G$ множество слов W_g является ее идентификатором. Пусть $g, h \in G$. Если $\mu(g) \neq \mu(h)$, то по построению $W_g \cap L_g \neq W_g \cap L_h$, так как $W_g \cap L_g \neq \emptyset$ и $W_g \cap L_h = \emptyset$. Пусть $\mu(g) = \mu(h)$. Если g и h принадлежат одному и тому же классу ε -неотличимости, то $W_g = W_h$ по построению и, следовательно, $L_g \cap W_g = L_h \cap W_g$. Пусть $\mu(g) = \mu(h)$ и вершины g и h принадлежат разным классам ε -неотличимости. Покажем, что существует слово $w \in W_g \cap W_h$ такое, что $w \in L_g \oplus L_h$. Пусть вершины g и h принадлежат одному и тому же классу X разбиения π_{k-1} и $\mu(X) = x$. Пусть, далее, при построении разбиения π_k класс X разбивается на два подкласса X' и X'' такие, что $g \in X'$, $h \in X''$. Тогда для некоторой метки $y \in M$ или (1) $g \star xy = \emptyset$ и $h \star xy \neq \emptyset$, или (2) существует класс $Y' \in \pi_k$ такой, что $\mu(Y') = y$, $g \star xy \in Y'$ и $h \star xy \notin Y'$ или $h \star xy = \emptyset$. Если в случае (1) $g \star xy = \emptyset$ или в случае (2) $h \star xy = \emptyset$, то $xy \in W_g \cap W_h$ и искомое слово $w = xy$. Пусть $h \star xy \in Y''$, где $Y'' \in \pi_k$. Тогда $xW_{Y'} \subseteq W_g$, $xW_{Y''} \subseteq W_h$ и существует класс Y , такой, что $Y \in \pi_{k-l}$, $Y \notin \pi_{k-l+1}$, $1 < l \leq k-1$ и Y – наименьший по включению класс для которого $Y', Y'' \subseteq Y$. Так как при построении π_{k-l+1} класс Y разбивается на два подкласса, то для некоторой метки $z \in M$ или (1) $Y' \star yz = \emptyset$ и $Y'' \neq \emptyset$, или (2) существует класс $Z' \in \pi_{k-l+1}$ такой, что $\mu(Z') = z$, $Y' \star yz \subseteq Z'$ и $(Y'' \star yz) \cap Z' = \emptyset$. Если в случае (1) $g \star xyz = \emptyset$ или в случае (2) $h \star xyz = \emptyset$, то $xyz \in W_g \cap W_h$ и искомое слово $w = xyz$. Если $h \star xyz \in Z''$, где $Z'' \in \pi_{k-l+1}$, то $xyW_{Z'} \subseteq W_g$ и $xyW_{Z''} \subseteq W_h$. Продолжая построение слова w , получим, что для некоторой метки $p \in M$ или (1) $g \star xyz \dots p = \emptyset$ и $h \star xyz \dots p \neq \emptyset$, или (2) существует класс $P' \in \pi_2$ такой, что $\mu(P') = P$, $g \star xyz \dots p \in P'$ и $h \star xyz \dots p \notin P'$ или $h \star xyz \dots p = \emptyset$. Если в случае (1) $g \star xyz \dots p = \emptyset$ или в случае (2) $h \star xyz \dots p = \emptyset$, то $xyz \dots p \in W_g \cap W_h$ и искомое слово $w = xyz \dots p$. Если $h \star xyz \dots p \in P''$, то $xyz \dots p \in W_g$ и $xyz \dots p \in W_h$. Тогда существует класс $P \in \pi_1$ такой, что $P = P' \cup P''$. Покажем, что класс P на 2-м шаге алгоритма 3.1 разбивается на подклассы тогда и только тогда, когда для некоторой метки $q \in M$ выполняется $P' \star pq = \emptyset$ и $P'' \star pq \neq \emptyset$ или наоборот $P' \star pq \neq \emptyset$ и $P'' \star pq = \emptyset$. Действительно, в разбиении π_1 каждый класс содержит все одинаково помеченные вершины, поэтому, если для любой вершины $g \in P$ выполняется $g \star pq \neq \emptyset$, то все вершины $g \star pq$ находятся в одном классе. Тогда слово $xyz \dots pq \in W_g \cap W_h$ и искомое слово $w = xyz \dots pq$. \square

Таким образом, алгоритм 1 правильно строит идентификаторы всех вершин про-

извольного D-графа G . Оценку сложности алгоритма 1 и сложности идентификаторов, построенных при его помощи дает следующая теорема.

Теорема 3. Алгоритм 1 за время $O(m \cdot n^2 + n^3)$ для каждой вершины $g \in G$ строит идентификатор W_g , высота и кратность которого равны $O(n)$.

Доказательство. В наихудшем случае на каждой итерации алгоритма 1 в отдельный класс отделяется только одна вершина. Тогда, с учетом того, что первоначальное разбиение состоит из m классов, количество итераций ограничено сверху числом $n - m + 1$.

Первоначально в каждом из множеств слов, порождаемых алгоритмом, находятся слова длины 1. На каждой итерации алгоритма максимальная длина слова среди всех слов, находящихся в этих множествах, увеличивается на 1. Таким образом, по окончании работы алгоритма, высота построенных им идентификаторов не превосходит $n - m + 1$.

Вернемся к рассмотрению наихудшего случая. Если на каждой итерации алгоритма отделяется ровно одна вершина, то множество слов, ассоциированное с еще неразличенными вершинами увеличивается на одно слово. Так как количество вершин в одном блоке не превосходит $n - m + 1$, то в указанном множестве может находиться не более $n - m + 1$ слов. Следовательно кратность построенных алгоритмом 3.1. идентификаторов не превосходит $n - m + 1$.

Наиболее трудоемкой частью каждой из итераций алгоритма 1 является объединение множеств слов. Для объединения двух множеств слов кратностью $O(n)$, высотой $O(n)$ и алфавитом мощности m достаточно воспользоваться алгоритмом лексикографической сортировки [7]. Сложность сортировки с такими параметрами равна $O(m \cdot n + n^2)$. Таким образом, сложность построения идентификаторов всех вершин графа G , с учетом верхней оценки числа итераций алгоритма 3.1., равна $O(m \cdot n^2 + n^3)$. \square

5. Диагностический эксперимент. Напомним, что эксперимент с графом G называется диагностическим, если априори полностью известен граф G и МА установлен в произвольную неизвестную ему вершину g этого графа, а целью эксперимента является определение этой вершины, то есть отличие вершины g от всех других вершин. Из определения эксперимента следует, что он осуществляется посредством прохождения МА некоторого связанного с тестом P множества путей по графу G из вершины g .

Очевидно, что если граф G не приведен, то диагностический эксперимент с ним невозможен. Действительно, если $(g, h) \in \varepsilon$, то вершины $g, h \in G$ по определению неотличимы никаким словом из M^+ . Если граф G приведен, то, как показано в разделе 2, существует k такое, что $L_g^k \neq L_h^k$ для всех $g \neq h, g, h \in G$. Тогда $P = \bigcup_{g \in G} L_g^k$ является диагностическим тестом для \mathcal{G} , поскольку порожденный P эксперимент W является диагностическим экспериментом с графом G , так как $W \cap L_g \neq W \cap L_h$, то есть $\overline{W}_g \neq \overline{W}_h$ для всех $g \neq h$. Поэтому в дальнейшем, если не оговорено противное, будем считать, что все рассматриваемые графы приведены.

Следующая теорема описывает класс диагностических тестов, определяемых идентификаторами вершин графа G . Рассмотрим множество слов $P = \bigcup_{g \in G} W_g$,

где W_g является некоторым идентификатором вершины $g \in G$. Очевидно, что для любых различных вершин $g, h \in G$ выполняется $P \cap L_g \neq P \cap L_h$. Действительно, так как $W_g \subseteq P$, то по определению идентификатора $W_g \cap L_g \neq W_g \cap L_h$, а следовательно $P \cap L_g \neq P \cap L_h$. Таким образом, справедливо следующее утверждение.

Теорема 4. Множество слов $P = \bigcup_{g \in G} W_g$ является диагностическим тестом для графа G при любом семействе идентификаторов $\{W_g\}_{g \in G}$.

Эта теорема дает метод построения диагностических тестов для помеченных графов различных типов.

Опишем подробно каждый этап диагностического эксперимента.

Для построения теста P воспользуемся алгоритмом 1. Кратность такого теста равна $O(n^2)$, его высота не превосходит $n - m + 1$, а его объем равен $O(n^3)$. Для построения теста P достаточно $O(m \cdot n^2 + n^3)$ шагов.

На втором этапе МА получает экспериментальные данные \mathbf{W} на основе теста P . Так как перемещение МА по исследуемому графу G может быть осуществлено с использованием различных стратегий (алгоритмов), то экспериментальные данные могут быть получены с использованием различных методов. Первая стратегия заключается в том, что МА, стартуя из неизвестной ему вершины h графа G , строит разбиение теста P на два класса путем непосредственной проверки наличия/отсутствия в G путей, помеченных словами из P . При этом МА анализирует окрестности вершин при их прохождении, одновременно уменьшая множество подлежащих проверке слов. Причем проверка каждого слова проводится из начальной вершины h , априори неизвестной МА. Экспериментальными данными являются полученное в результате таких проверок разбиение $\pi(h)$ теста P на два класса $P_h^+ = P \cap L_h$ и $P_h^- = P - L_h$.

Алгоритм 2. ДИАГНОСТИЧЕСКИЙ ЭКСПЕРИМЕНТ (СТРАТЕГИЯ 1)

Вход. Приведенный D-граф $G = (G, E, M, \mu)$.

Семейство $\{W_{g_1}, \dots, W_{g_n}\}$ идентификаторов всех вершин графа G .

МА установлен в неизвестную ему вершину $h \in G$.

Выход. Разбиение $\pi(h)$ теста P на два класса $P_h^+ = P \cap L_h$ и $P_h^- = P - L_h$.

Метод.

1. Положим $P_h^+ = \emptyset$, $P_h^- = \emptyset$, $P' = \emptyset$, $P'' = \emptyset$.
2. Считать метку $\mu(h)$ вершины h .
3. Для каждого слова $w \in P$, начинающегося с $\mu(h)$, положим $P' = P' \cup \{w\}$.
4. Положим $P_h^- = P \setminus \{P'\}$.
5. Если $P' = \emptyset$, то положим $P_h^+ = \emptyset$ и $P_h^- = P$. Алгоритм прекращает работу.
6. Если $P' \neq \emptyset$, то выберем слово $w \in P'$ и положим $P' = P' \setminus \{w\}$. Иначе алгоритм прекращает работу.
7. Положим $i = 1$.
8. Положим $w_i = \text{pre}_i(w)$.
9. Если $i = d(w)$, то положим $P_h^+ = P_h^+ \cup \{w\}$. Перейти к п. 6.
10. Перейти в вершину $h \star w_i$.
11. Вычислить $C = \text{Ob}^-(h \star w_i)$.

12. Для каждого слова $u \in P'$ такого, что u является начальным отрезком какого-либо слова из C , положим $P_h^- = P_h^- \cup \{u\}$ и $P' = P' \setminus \{u\}$.

13. Если $w_{i+1} \in C$, то положим $P_h^- = P_h^- \cup \{w\}$. Перейти к п. 6.

14. Положим $i = i + 1$. Перейти к п. 8.

Рассмотрим свойства алгоритма 2.

Теорема 5. *Алгоритм 2 строит разбиение диагностического теста P на два класса за $O(n^3)$ шагов.*

Доказательство. Поскольку при каждой итерации цикла из P' удаляется, по крайней мере, одно слово, то этот алгоритм всегда завершается и при $P' = \emptyset$ получаем разбиение $\pi(h)$ теста P на два класса P_h^+ и P_h^- . В наихудшем случае алгоритм построения $\pi(h)$ требует проверки всех слов из P . Следовательно, сложность такого построения равна объему P , то есть равна $O(n^3)$. \square

На данном этапе уже могут быть сделаны заключения об операционной высоте и кратности диагностического эксперимента. Поскольку, в общем случае, тест P кратный, то при выполнении второго этапа эксперимента предполагается, что или МА перед началом анализа слова $w \in P'$ переносится экспериментатором в начальную вершину h , или в h помещается новый экземпляр МА. Таким образом, для теста P кратности k и данной стратегии проведения диагностического эксперимента операционная кратность диагностического эксперимента не превосходит k , а операционная высота диагностического эксперимента равна высоте теста P .

На третьем этапе эксперимента определяется начальная вершина путем анализа экспериментальных данных. Для теста P с каждой вершиной $g \in G$ свяжем его разбиение $\pi(g)$ на два класса $P_g^+ = P \cap L_g$ и $P_g^- = P - L_g$. Ясно, что для построения семейства $\{\pi(g)\}_{g \in G}$ таких разбиений достаточно $O(n^4)$ шага. Далее, построенное экспериментально разбиение $\pi(h)$ сравнивается со всеми разбиениями из семейства $\{\pi(g)\}_{g \in G}$. Если для некоторого $\pi(g)$ выполняется $\pi(g) = \pi(h)$, то $g = h$ и эксперимент окончен. Из правил построения семейства $\{\pi(g)\}_{g \in G}$ и разбиения $\pi(h)$ следует, что последнее равенство всегда достигается. Из описания алгоритма построения разбиения $\pi(h)$ и правила вывода заключения непосредственно следует, что действительная начальная вершина h всегда будет определена правильно.

Прежде чем приступить к изложению второй стратегии, рассмотрим полезную конструкцию, позволяющую переходить от множества слов к помеченным графам. Пусть $W = \{w_1, \dots, w_l\}$ некоторое множество конечных слов в алфавите M . С каждым словом $w_i = x_{i_1} \dots x_{i_k}$, $1 \leq i \leq l$, свяжем помеченное корневое ориентированное дерево $T(w_i)$, множество вершин которого равно $\{t_{i_1}, \dots, t_{i_k}\}$, множество дуг состоит из всех пар $(t_{i_j}, t_{i_{j+1}})$, $1 \leq j < k$, метка $\mu_T(t_{i_j})$ равна x_{i_j} , а корнем является вершина t_{i_1} . С множеством W свяжем помеченный лес $F(W)$, являющийся прямой суммой всех корневых деревьев $T(w)$, где $w \in W$. Если все слова в W начинаются с одной и той же метки, то с W связывается помеченное корневое дерево $T(W)$, полученное из $F(W)$ отождествлением всех корней t_{i_1} , $1 \leq i \leq l$, и последующей детерминизацией, то есть многократным и исчерпывающим применением следующей операции: если в множество преемников некоторой вершины попадают вершины с одинаковыми метками, то такие вершины отождествляем, заменяя возникающие

кратные дуги одной дугой. Очевидно, что $T(W)$ является D-графом и $W = L_{T(W)}$.

С тестом P свяжем помеченный лес $F(P)$ корневых деревьев и детерминизируем этот лес. Обозначим через P_x множество всех слов из P , начинающихся с метки $x \in M$. Ясно, что P_x является объединением идентификаторов всех вершин $g \in G$ таких, что $\mu(g) = x$. Тогда корневое ориентированное дерево $T(P_x)$ является компонентой леса $F(P)$. Для всех $x \in M$ с деревом $T(P_x)$ свяжем множество вершин $G_x = G \star x$ (то есть, всех вершин с меткой x). Обозначим корень дерева $T(P_x)$ через t_x . Для каждого слова $w \in P_x$ вершину $t_x \star w$ дерева $T(P_x)$ назовем выделенной. С каждой выделенной вершиной $t = t_x \star w$ свяжем два множества $G(t) = \{g \in G \mid w \in L_g\}$ и $\tilde{G}(t) = \{g \in G \mid \mu(g) = x \wedge w \notin L_g\}$. Множество всех выделенных вершин дерева $T(P_x)$ обозначим через T_x .

Во второй стратегии процесс получения экспериментальных данных заключается в том, что МА, стартуя из неизвестной ему вершины h графа G , проверяет наличие/отсутствие в G путей, помеченных теми же словами, что и пути из корня дерева $T(P_{\mu(h)})$ во все выделенные вершины этого дерева. В зависимости от исхода каждой из таких проверок уменьшаются подмножества вершин графа G , связанные с выделенными вершинами дерева $T(P_{\mu(h)})$. По окончании работы алгоритма множество $G_{\mu(h)}$ содержит единственную вершину, которая и является искомой начальной вершиной графа G . Таким образом, вторая стратегия объединяет этапы получения экспериментальных данных и вывода заключений.

Алгоритм 3. ДИАГНОСТИЧЕСКИЙ ЭКСПЕРИМЕНТ (СТРАТЕГИЯ 2)

Вход. D-граф $G = (G, E, M, \mu)$.

Лес $F(P)$ корневых деревьев.

МА установлен в неизвестную ему вершину $h \in G$.

Выход. Множество $G_{\mu(h)}$.

Метод.

1. Считать метку $\mu(h)$ вершины h .
2. Выберем из леса $F(P)$ дерево $T(P_{\mu(h)})$.
3. Если $|G_{\mu(h)}| = 1$, то вершина h уже определена. Алгоритм прекращает работу.
4. Выберем выделенную вершину $s = t_{\mu(h)} \star w$ и положим $T_{\mu(h)} = T_{\mu(h)} \setminus \{s\}$.
5. Положим $i = 1$.
6. Положим $w_i = \text{pre}_i(w)$.
7. Если переход из вершины h в вершину $h \star w_i$ графа G невозможен, то перейти к п. 8, иначе перейти к п. 9.
8. Для каждой вершины $t \in T_{\mu(h)}$ положим $G(t) = G(t) \setminus G(s)$, $\tilde{G}(t) = \tilde{G}(t) \setminus G(s)$ и $G_{\mu(h)} = G_{\mu(h)} \setminus G(s)$. Перейти к п. 3.
9. Если $i = d(w)$, то для каждой вершины $t \in T_{\mu(h)}$ положим $G(t) = G(t) \cap G(s)$, $\tilde{G}(t) = \tilde{G}(t) \setminus G(s)$, $G_{\mu(h)} = G(s)$. Перейти к п. 3.
10. Положим $i = i + 1$. Перейти к п. 6.

Анализ алгоритма 3 начнем со следующей теоремы.

Теорема 6. По окончании работы алгоритма 4.2 множество $G_{\mu(h)}$ всегда содержит единственную вершину.

Доказательство. Предположим, что после некоторой итерации алгоритма $|G_{\mu(h)}| > 1$ и $T_x = \emptyset$, то есть условие окончания алгоритма не достигнуто, но дальнейшее его выполнение невозможно. Пусть $G_{\mu(h)} = \{h\} \cup G'$, где $h \notin G'$. Из построения теста $P_{\mu(h)}$ следует, что для любой вершины $g \in G'$ существует слово $w \in P_{\mu(h)}$ такое, что $w \in L_h \oplus L_g$. Тогда вершина $s = t_{\mu(h)} \star w$ является выделенной вершиной дерева $T(P_{\mu(h)})$ и, либо $h \in G(s)$ и $g \in \tilde{G}(s)$, либо наоборот $g \in G(s)$ и $h \in \tilde{G}(s)$. Так как $T_{\mu(h)} = \emptyset$, то вершина s была удалена из $T_{\mu(h)}$ в ходе некоторой итерации алгоритма. Рассмотрим как могло произойти такое удаление. Если вершина s была удалена из $T_{\mu(h)}$ по причине того, что $G(s) = \emptyset$, то на этой же итерации либо вершина h , либо вершина g была удалена из $G_{\mu(h)}$, что невозможно. Следовательно, на этой итерации МА побывал в вершине s и по этой причине эта вершина была удалена из $T_{\mu(h)}$. Тогда на этой же итерации либо вершина h , либо вершина g была удалена из $G_{\mu(h)}$, что невозможно. Из проведенных рассуждений следует, что при $T_{\mu(h)} = \emptyset$ выполняется $G_{\mu(h)} = \{h\}$. \square

В наихудшем случае на каждой итерации алгоритма из $G_{\mu(h)}$ удаляется одна вершина. Так как число вершин графа \mathcal{G} с меткой равной $\mu(h)$ не превосходит $n - m + 1$, то алгоритм оканчивает свою работу через не более чем $n - m$ итераций.

Поскольку, в общем случае, дерево $T(P_{\mu(h)})$ содержит более одной ветви, то при реализации предложенной стратегии предполагается, что МА перед началом анализа выделенной вершины $t \in T_{\mu(h)}$ переносится экспериментатором в начальную вершину $h \in G$, или в h помещается новый экземпляр МА. В наихудшем случае МА проверяет $n - m$ выделенных вершин дерева $T(P_{\mu(h)})$. Следовательно, операционная кратность диагностического эксперимента не превосходит $n - m$. Операционная высота диагностического эксперимента равна высоте множества слов $P_{\mu(h)}$.

6. Заключение. В работе решена задача различения вершин топологической модели (помеченного графа-карты) операционной среды мобильного агента. Решение заключается в построении диагностического эксперимента и способа его реализации на модели. Найдены оценки сложности соответствующих алгоритмов. С точки зрения синтеза диагностических экспериментов с помеченными графами представляет интерес исследование условий существования и методов построения оптимальных по сложности идентификаторов вершин.

1. *James A. Anderson.* Automata Theory with Modern Applications. – Cambridge University Press, 2006. – 255 p.
2. *Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled.* Model Checking. – The MIT Press, 1999. – 416 p.
3. *Dudek G., Jenkin M.* Computational Principles of Mobile Robotics. – Cambridge University Press, Cambridge, 2000. – 280 p.
4. *Dudek G., Jenkin M., Milios E., Wilkes D.* Map validation and Robot Self-Location in a Graph-Like World // Robotics and Autonomous Systems. – 1997. – Vol. 22, № 2. – P. 159–178.
5. *Zvi Kohavi.* Switching and finite automata theory. – McGraw-Hill, New York, 1970. – 592 p.
6. *Мур Э.Ф.* Умозрительные эксперименты с последовательностными машинами // Автоматы. – М.: ИЛ, 1956. – С. 179–210.
7. *Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman.* The design and analysis of computer algorithms. – Addison-Wesley, 1974, – 432 p.

8. *Levitt T., Lawton D.T.* Qualitative navigation for mobile robot // Artificial Intelligence, 1990. – v. 40. – p. 306–360.
9. *Сапунов С.В.* Контроль детерминированных графов // Труды ИПММ. – 2003. – в. 8. – С. 106-110.

I.S. Grunsky, S.V. Sapunov

Topological identifiers of vertices of vertex labeled graphs.

The self-location problem for mobile agent in the topological environment is considered. Environment is modeled by the vertex labeled graph. The agent can move by graph edges and observe labels of vertices. A finite sets of words over the vertex labels alphabet distinguishing a given vertex from other called vertex identifier is proposed. Existence conditions and complexity estimations for vertex identifiers are discovered and constructions methods are developed. Polynomial constructions and realizations methods for experiments detecting the vertex where the mobile agent start walking based on all vertices' identifiers construction are proposed.

Keywords: *vertex labeled graph, vertex identifier, distinguishing experiment.*

І.С. Грунський, С.В. Сапунов

Топологічні ідентифікатори вершин позначених графів.

Розглянуто задачу визначення мобільним агентом свого положення в середовищі, яке моделюється за допомогою графа з позначеними вершинами. Агент може пересуватися дугами графа та спостерігати позначки вершин. Введено визначення ідентифікаторів вершин, тобто скінченні множини слів в алфавіті позначок, які відрізняють одну вершину графа від усіх інших його вершин. Знайдено умови існування, оцінки складності ідентифікаторів та розроблено методи їх побудови. Розроблено поліноміальні методи побудови і проведення експериментів по визначенню початкової вершини графа, які ґрунтуються на побудові ідентифікаторів усіх його вершин.

Ключові слова: *графи з позначеними вершинами, ідентифікатор вершини, діагностичний експеримент.*

*Ин-т прикл. математики и механики НАН Украины, Донецк
sapunov sv@iamm.ac.donetsk.ua*

Получено 25.11.10