Анализ графов-кактусов с использованием автоматов: свойства и время распознавания

А. А. Демидова[1]

Данная работа посвящена исследованию применения автоматов со стираемыми красками для определения того, является ли произвольный связный плоский простой неориентированный граф кактусом. Приводится алгоритм для определения данного свойства, а также нижняя и верхняя оценки числа шагов, которое должен совершить автомат для завершения обхода.

Ключевые слова: Автоматы, графы, графы-кактусы.

1. Введение

Подробный обзор области теории автоматов, связанной с обходами лабиринтов, был представлен в работе [1]. Функционирование автомата в лабиринтах и на графах отличается, поскольку в последнем случае отсутствует компас, который позволил бы получать дополнительную информацию об окружающей среде [2]. Работа [3] посвящена известным результатам в области автоматов, осуществляющих обход графов.

Обходы прямоугольных лабиринтов и графов автоматами с красками рассматривались, в частности, в работах [4] – [6].

В работе [7] исследовалось применение автоматов с красками для определения того, является ли произвольный связный плоский простой неориентированный граф деревом и псевдодеревом, и было доказано, что для определения этих свойств автомату достаточно двух стираемых красок.

В данной работе рассматривается исследование автоматом того, является ли граф, обход которого он осуществляет, графом-кактусом. Графом-кактусом является такой связный граф, в котором любое ребро принадлежит не более чем одному циклу, а любые два цикла могут иметь не более одной общей вершины.

Графы-кактусы могут применяться при моделировании сетевых структур, которые не получится отразить с помощью деревьев. При этом дерево, как и псевдодерево, является частным случаем графа-кактуса, и

[1] *Демидова Анна Андреевна* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: anna.dem98@mail.ru.

Demidova Anna Andreevna — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

изначально рассматриваемый тип графов назывался деревьями Хусими в честь работавшего над ними Коди Хусими (название было дано Ф. Харари и Дж. Ю. Уленбеком в [8]).

Графы-кактусы могут использоваться для декомпозиции набора геномов на ряд цепочек и подсетей для дальнейшего индексирования и сравнения [9]. Графы этого класса также могут применяться в теории коммуникационных сетей [10] и других областях.

2. Основные понятия и формулировка результатов

Обозначим через $\mathbf{G}$ класс всех связных плоских простых неориентированных графов. Будем считать, что до начала обхода графа все его рёбра имеют серый цвет.

В данной работе рассматривается автомат $\mathcal{A}_c$, осуществляющий обход графа из класса $\mathbf{G}$ с целью определения того, является ли этот граф кактусом. Во время исследования графа автомат обладает частичной информацией о посещаемых вершинах и инцидентных им рёбрах. По умолчанию обход осуществляется по правилу левой руки.

В распоряжении автомата есть 5 стираемых красок, а также серая краска, которой изначально были окрашены все рёбра графа.

Основным результатом является Теорема 1.

Теорема 1. *Существует автомат $\mathcal{A}_c$ с 6 красками, который сможет установить, является ли произвольный граф $\mathbf{g} \in \mathbf{G}$ кактусом.*

Одним из частных случаев графа-кактуса является дерево. Далее будем предполагать, что в рассматриваемом графе есть циклы.

Произвольный цикл делит плоскость на две грани, и до установления его наличия автомат обходит подграфы, имеющие общие вершины с этим циклом и лежащие только в одной их них.

Максимальные подграфы, имеющие общие вершины с рассматриваемым циклом и лежащие в той грани, которую автомат обходил до установления его наличия, будем называть **левыми ветвлениями**.

Единственное левое ветвление, обход которого мог быть не закончен к моменту обнаружения цикла, является первым – тем, из которого автомат попал на цикл.

Максимальные подграфы, имеющие общие вершины с рассматриваемым циклом и лежащие в той грани, которая до обнаружения цикла не была посещена автоматом, будем называть **правыми ветвлениями**.

Перед обходом правых ветвлений очередного цикла автомат меняет направление движения по правилу левой руки на противоположное тому, в котором изначально обходил этот цикл. После того, как исследование правых ветвлений цикла завершено, автомат снова меняет направление обхода.

«Галочка» – пара соседних рёбер, лежащих на цикле и перекрашиваемых автоматом в фиолетовый или красный цвет перед началом обхода правого ветвления, исходящего из вершины, являющейся общей для этих рёбер.

Число «галочек» на цикле совпадает с количеством принадлежащих ему вершин. Соседние «галочки» на одном и том же цикле имеют общее ребро, поэтому при постановке очередной «галочки» автомату нужно будет перекрашивать только одно ребро.

Алгоритм, которому следует автомат $\mathcal{A}_c$, можно описать следующим образом:

1) Автомат осуществляет обход графа по правилу левой руки и наносит чёрную краску на все рёбра, по которым проходит, пока не обнаружит цикл. Наличие цикла устанавливается в ситуации, когда автомат только что покрасил некоторое ребро в чёрный цвет и оказался в вершине, которой инцидентны другие не серые рёбра.

2) Автомат ставит «галочку» в вершине, в которой было установлено наличие цикла. При постановке первой «галочки» на цикле автомат должен перекрасить последнее ребро, по которому он проходил до момента обнаружения цикла, а также первое не серое ребро справа. Оба эти ребра до перекрашивания являются чёрными, а выбор первого не серого ребра справа обусловлен тем, что правее него ещё может быть не обойдённое правое ветвление. Если рассматриваемый цикл – самый первый из обнаруженных автоматом, то для размещения «галочки» используется красный цвет, а в противном случае – фиолетовый.

3) При постановке первой «галочки» очередного цикла автомат также перекрашивает в синий или голубой цвет первое левое ребро в первом левом ветвлении этого цикла. Данное ребро будет указывать направление движения после завершения обхода правых ветвлений цикла. Автомат использует синюю краску для перекрашивания чёрных рёбер и голубую – для фиолетовых. Если соответствующее ребро является серым или красным, то перекрашивание не происходит.

4) Автомат приступает к обходу правых ветвлений рассматриваемого цикла, поменяв при этом направление движения по правилу левой руки на противоположное. Перед обходом очередного правого

ветвления автомат ставит «галочку» в вершине цикла, из которой это ветвление исходит, причём выбирать нужно будет первое не серое слева, поскольку направление движения после начала обхода правых ветвлений цикла сменилось на противоположное. Все «галочки» на графе, кроме самой первой, поставленной на первом обнаруженном цикле, имеют фиолетовый цвет.

5) После завершения обхода правых ветвлений цикла автомат переходит в его первое левое ветвление по отмеченному ранее синему, голубому или серому ребру. В случае, если возможных рёбер несколько, выбирается самое левое из них. Если ребро было голубым, то автомат перекрашивает его в фиолетовый цвет. В противном случае автомат использует чёрную краску.

6) Автомат устанавливает, что граф не является кактусом, и завершает обход в 4 ситуациях, которые будут описаны в разделе 5.

7) Если автомат не обнаружил признаков того, что граф не является кактусом, то обход завершается при возвращении из первого левого ветвления первого обнаруженного цикла к «галочке» красного цвета. В случае, если у данного цикла не было первого левого ветвления, исследование графа завершается после завершения обхода всех правых ветвлений данного цикла.

3. Возможное взаимное расположение циклов, имеющих общие рёбра, в графе, не являющемся кактусом

Лемма 1. *Допустим, что автомат при обходе графа установил наличие цикла, у которого на самом деле есть общие рёбра с другими циклами графа. Тогда эти новые циклы могут лежать только в правых ветвлениях только что обнаруженного цикла.*

Доказательство. Предположим противное – в левом ветвлении только что обнаруженного цикла есть другой цикл, имеющий с ним по крайней мере одно общее ребро. Поскольку все левые ветвления обнаруженного цикла, кроме, возможно, самого первого, уже были обойдены, а граф является плоским, то из первого левого ветвления нельзя сразу попасть ни в одну вершину уже обнаруженного цикла, не миновав ту, из которой это ветвление исходит. Следовательно, в первом левом ветвлении обнаруженного цикла не может быть цикла, который имел бы с ним общие рёбра. Таким образом, рассматриваемые циклы могут лежать только в правых ветвлениях обнаруженного цикла. □

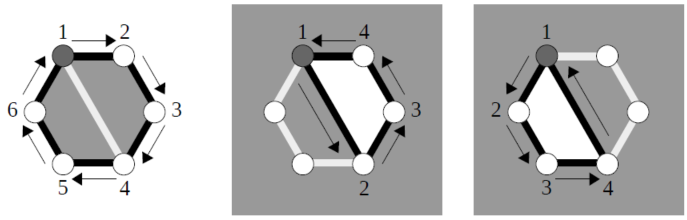

Рис. 1. Возможные варианты обхода графа, не являющегося кактусом, до обнаружения первого цикла, а также области, в которых могут лежать левые и правые ветвления

На Рисунке 1 представлены 3 варианта обхода одного и того же графа, не являющегося кактусом, в которых автомат начинает из одной и той же вершины, но в первый момент времени начинает двигаться в разных направлениях. Области, в которых лежат правые ветвления обнаруженных циклов, выделены цветом. Во всех 3 ситуациях второй цикл, мешающий графу быть кактусом, лежит в правом ветвлении.

4. Правые ветвления и «галочки»

Допустим, что автомат установил наличие цикла и приступил к обходу правых ветвлений. Перед тем, как начать исследование очередного правого ветвления, автомат перекрашивает в фиолетовый цвет рёбра, инцидентные текущей вершине цикла и лежащие на этом цикле. На Рисунках 2а-г представлены различные ситуации, отражающие результаты обхода правого ветвления, соответствующего вершине, в которой только что была поставлена «галочка». На Рисунке 2, как и на всех, которые будут представлены далее, рёбра, раскрашенные в цвета кроме серого и чёрного, отмечены соответствующими буквами: красные ребра помечены буквой «К», фиолетовые — буквой «Ф». Ситуация на Рисунке 2д невозможна. Рассмотрим все эти случаи подробнее.

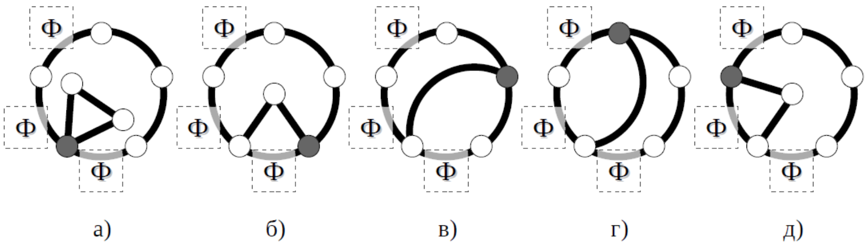

Рис. 2. Разбор ситуаций

На Рисунке 2а представлен граф-кактус. В правом ветвлении может не быть цикла, но автомат в любом случае попадает в вершину, которой инциденты 2 фиолетовых ребра, и видит слева ребро этого цвета. При этом автомат мог только что красить некоторое ребро в чёрный цвет (если в ветвлении есть цикл, имеющий общую вершину с исходным, и он был только что обойдён) или нет (если таких циклов в ветвлении нет или же цикл есть, но он не имеет общих вершин с рассматриваемым).

На Рисунке 2б представлен граф, не являющийся кактусом. Автомат, приступив к обходу правого ветвления некоторой вершины исходного цикла, попадает в другую вершину цикла, в которой ещё не была поставлена фиолетовая «галочка», однако «галочка» уже была проставлена в предыдущей вершине цикла.

На Рисунке 2в также представлен граф, не являющийся кактусом. Автомат, приступив к обходу правого ветвления некоторой вершины исходного цикла, попадает в другую вершину цикла. Этот случай отличается от представленного на Рисунке 2б тем, что «галочки» нет даже в предыдущей вершине цикла.

Представленный на Рисунке 2г случай отличается от 2б только тем, что автомат попадает из правого ветвления в вершину, правое ветвление которой должно было быть исследовано в последнюю очередь. Одно из рёбер, инцидентных этой вершине, уже было ранее окрашено в фиолетовый цвет, потому это было необходимо для того, чтобы оставить «галочку» в самой первой вершине, правое ветвление которой исследовал автомат.

Ситуация, представленная на Рисунке 2д, не может возникнуть при обходе, поскольку, если бы в графе действительно был расположенный так второй цикл, то ещё на стадии исследования правого ветвления той вершины, в которую якобы попадает автомат, он оказался бы в одной из уже рассмотренных ранее ситуаций.

5. Признаки того, что граф не является кактусом

Опишем 4 ситуации, в которых автомат будет устанавливать, что граф не является кактусом.

1) **Автомат обнаруживает новый цикл и тут же видит ровно одно фиолетовое или красное ребро** (Рисунок 3).

   У автомата есть возможность поставить первую «галочку» на обнаруженном цикле, однако он сразу прерывает обход. Если фиолетовое ребро — первое не серое справа, то следует перейти к следующему случаю.

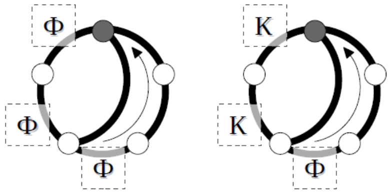

Рис. 3. Автомат устанавливает, что граф не является кактусом, и не ставит первую «галочку» на только что обнаруженном цикле

2) **Автомат обнаруживает новый цикл, и ему негде ставить второе ребро первой «галочки», поскольку первое не серое ребро справа не является чёрным** (Рисунок 4).

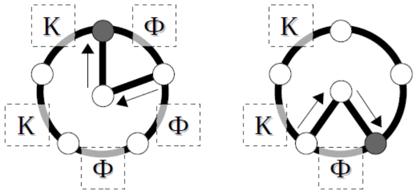

Рис. 4. Автомат устанавливает, что граф не является кактусом, так как негде поставить одно из рёбер первой «галочки» на только что обнаруженном цикле

3) **Автомат обнаруживает новый цикл, успешно ставит первую «галочку» на нём, но обнаруживает, что на данном цикле уже есть по крайней мере одно фиолетовое ребро** (Рисунок 5).

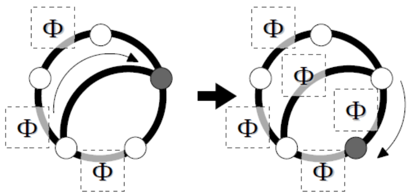

Рис. 5. Автомат устанавливает, что первая «галочка» на обнаруженном цикле на самом деле не является первой

Автомат ставит «галочку» и сталкивается с одиночным фиолетовым ребром или нечётным числом фиолетовых рёбер. Одинокое фиолетовое ребро (оно же — первое не серое справа) должно лежать на том же цикле — следовательно, это не первая «галочка» на цикле, что означает, что граф не является кактусом.

4) **Автомат ставит последнюю «галочку» на цикле (первое не серое ребро слева — фиолетовое), но, если пройти по первому не серому ребру слева, то обнаружится, что там, помимо этого ребра, чётное число фиолетовых рёбер, то есть нет следующего ребра на данном цикле — следовательно, решение автомата о том, что он завершил обход очередного цикла, является ошибочным** (Рисунок 6).

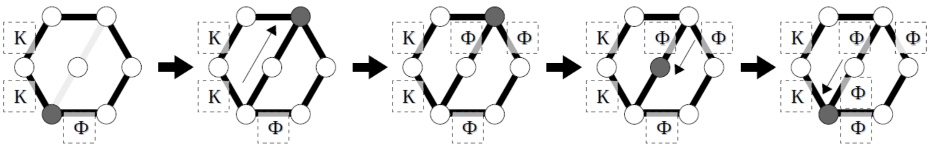

Рис. 6. Автомат устанавливает, что цикл на самом деле не обойдён до конца

Покажем, что такая ситуация не может возникнуть при обходе графа-кактуса. Допустим, что автомат завершил обход правых ветвлений некоторого цикла. В графе-кактусе «галочки», лежащие на разных циклах, не могут иметь общих рёбер. В кактусе после завершения обхода правых ветвлений цикла первое фиолетовое ребро слева будет лежать либо на том же цикле, либо в его первом правом ветвлении. Поскольку данное ветвление должно было быть обойдено полностью, оно добавляет в текущей вершине чётное число фиолетовых рёбер. При этом текущий цикл также добавляет чётное число фиолетовых или красных рёбер. Следовательно, в графе-кактусе такая ситуация не возникнет.

6. Обеспечение наличия левого ветвления в некоторой вершине рассматриваемого цикла

В алгоритме для псевдодеревьев в [7] автомат должен был оставлять нестандартную метку в одном из левых ветвлений. При повторном обходе обнаруженного цикла с целью установления наличия левых и правых ветвлений автомат получал информацию о том, есть ли какие-то ветвления. Если все вершины, посещённые автоматом, имели степень 2, то никаких ветвлений не было и граф просто являлся циклом. Если были вершины степени больше 3, причём автомат был в ситуациях, когда вершине были инцидентны хотя бы 2 чёрных ребра, то в графе были левые ветвления. Если же автомат посещал вершины степени 3, но признаков левых ветвлений не обнаружил, то в цикле есть только правые ветвления. В таком случае граф перекрашивал всё обратно в серый цвет (в силу отсутствия левых ветвлений «всё» включало в себя только обнаруженный цикл), после чего менял направление движения по правилу левой руки на противоположное, тем самым превращая старые правые ветвления в новые левые, и начинал обход заново.

В алгоритме для псевдодеревьев признаком окончания обхода было «левое ребро чёрное, правое ребро серое, сейчас автомат ничего не красил».

Рассмотрим следующую ситуацию: автомат нашёл цикл и начал исследование ещё не обойдённых его ветвлений, где обнаружил ещё один цикл. Если новый цикл лежит в первом левом ветвлении первого, то направление обхода не менялось, и для нового цикла первый будет в левом ветвлении. Если новый цикл лежит в одном из правых ветвлений исходного, то направление обхода к этому моменту уже поменялось, так что исходный цикл опять же будет лежать в левом ветвлении нового.

Наличие первых левых ветвлений у циклов необходимо для того, чтобы была возможность размещать синие или голубые рёбра. Ситуация, в которой автомату будет негде ставить эти рёбра, может возникнуть только в том случае, если наличие цикла было обнаружено в вершине степени 2 или вершине большей степени, но такой, что все инцидентные рёбра лежат в правых ветвлениях. Такая ситуация может возникнуть только в той вершине, откуда начинался обход.

7. Порядок изменения цветов рёбер при обходе графа автоматом

Автомат может перекрашивать рёбра следующим образом:

1) **Серый** $\rightarrow$ **Чёрный**: стандартная операция при обходе новых рёбер;
2) **Чёрный** $\rightarrow$ **Фиолетовый**: размещение «галочек»;
3) **Чёрный** $\rightarrow$ **Красный**: размещение первой «галочки» на первом обнаруженном цикле;
4) **Чёрный** $\rightarrow$ **Синий**: размещение синего ребра;
5) **Фиолетовый** $\rightarrow$ **Голубой**: размещение голубого ребра.
6) **Синий** $\rightarrow$ **Чёрный** и **Голубой** $\rightarrow$ **Фиолетовый**: автомат идёт туда, где ещё нужно что-то обойти, после завершения обхода правых ветвлений некоторого цикла.

Таблица 1. Возможные изменения цвета ребра за один шаг

Старый\новый цвет	Сер.	Чёрн.	Фиол.	Красн.	Син.	Гол.
Серый	X	+	–	–	–	–
Чёрный	–	X	+	+	+	–
Фиолетовый	–	–	X	–	–	+
Красный	–	–	–	X	–	–
Синий	–	+	–	–	X	–
Голубой	–	–	+	–	–	X

Невозможные действия:

1) **Серый** → **Фиолетовый** и **Серый** → **Красный**: только рёбра циклов могут быть перекрашены в фиолетовый или красный цвет. Если ребро серое, то его принадлежность какому-либо циклу ещё не установлена;

2) **Серый** → **Синий** и **Серый** → **Голубой**: в синий цвет могут быть перекрашены только чёрные рёбра, а в голубой – только фиолетовые. Использование этих красок на сером ребре могло бы привести к тому, что новый цикл будет обнаружен при перекрашивании какого-то ребра не в чёрный цвет;

3) **Любой цвет** → **Серый**: перекрашивание обратно в серый цвет невозможно;

4) **Фиолетовый** → **Чёрный** и **Фиолетовый** → **Красный**: фиолетовые рёбра не могут быть перекрашены обратно в чёрный цвет. Красных рёбер в графе всего 2, и они сразу были перекрашены из чёрных в красные;

5) **Синий** → **Фиолетовый** и **Синий** → **Красный**: синие рёбра могут сразу стать только чёрными;

6) **Голубой** → **Чёрный** и **Голубой** → **Красный**: голубые рёбра могут сразу стать только фиолетовыми;

7) **Чёрный** → **Голубой** и **Фиолетовый** → **Синий**: синяя и голубая краски нужны для того, чтобы отличать рёбра, которые раньше были чёрными и фиолетовыми;

8) **Красный** → **Любой цвет**: красные рёбра остаются красными до конца обхода;

9) **Синий** → **Голубой** и **Голубой** → **Синий**.

8. Оценки времени установления автоматом свойства графа быть графом-кактусом

В качестве одного шага алгоритма будем рассматривать проход автомата по некоторому ребру в одну сторону вместе со всеми действиями, которые могут сопровождать этот переход. Такие действия, в частности, подразумевают выбор ребра и то, нужно ли его красить. Вне зависимости от того, выбирает автомат первое ребро слева или, например, первое не серое ребро справа, на это действие вместе с проходом по ребру и возможным его перекрашиванием уходит один шаг алгоритма. Анализ ситуации в текущей вершине происходит мгновенно, то есть автомату не нужен дополнительный шаг для того, чтобы установить, что он обнаружил цикл, и так далее.

1) Рёбра, не лежащие на циклах, нужно проходить 2 раза (больше шагов нужно только для тех рёбер, которые потом становятся синими). Первое левое ветвление первого цикла в худшем случае нужно будет полностью обходить 2 раза, то есть проходить по каждому его ребру 4 раза;

2) По каждому ребру каждого цикла необходимо пройти по крайней мере 4 раза (больше шагов нужно только для тех рёбер, которые потом становятся голубыми): один раз – когда автомат проходит по нему в первый раз и красит в чёрный цвет, два раза – когда автомат ставит очередную «галочку» и возвращается в её центр, и ещё один раз – когда автомату нужно переместиться в следующую вершину цикла;

3) Каждому циклу нужно по одному синему или голубому ребру: 2 шага уходят на то, чтобы его поставить, и ещё 1 – чтобы потом перекрасить это ребро в чёрный или фиолетовый цвет;

4) В конце обхода правых ветвлений необходимо проверить, что следующее ребро на этом цикле действительно фиолетовое или красное. Для этого необходимо осуществить 2 шага.

Пусть в графе есть k рёбер и m циклов, а общее число рёбер, содержащихся в циклах, равно n. Нижняя оценка для числа шагов совпадает с полученной в работе [7] для деревьев и псевдодеревьев и следует из того, что по каждому ребру автомат пройдёт хотя бы 2 раза. Верхняя оценка числа шагов:

$$T \leq 4(k-n) + 4n + 3m + 2m, \tag{1}$$

где $4(k-n)$ – верхняя оценка числа шагов для обхода рёбер, не лежащих на циклах (умножение на 4 стоит из-за первого левого ветвления первого цикла – неизвестно, сколько там рёбер по сравнению с k),
$4n$ – число шагов для рёбер, принадлежащих циклам, от первого прохода до завершения обхода этих циклов;
$3m$ – число шагов для синих и голубых рёбер;
$2m$ – проверка.

Если в графе есть хотя бы один цикл, то $3 \leq n \leq k$. Кроме того, в каждом цикле есть по крайней мере 3 ребра, причём предполагается, что общих рёбер у циклов нет, – следовательно, $3m \leq n \leq k \Rightarrow m \leq \frac{n}{3} \leq \frac{k}{3}$. Тогда:

$$T \leq 4k + 5m \leq 4k + \frac{5k}{3} = \frac{17k}{3}. \qquad (2)$$

9. Выводы

Представленный в данной работе алгоритм для автоматов, осуществляющих обход связных плоских простых неориентированных графов, позволяет с использованием 6 красок установить, является ли рассматриваемый граф кактусом.

Список литературы

[1] Кудрявцев В. Б., Килибарда Г., Ушчумлич Ш., “Системы автоматов в лабиринтах”, *Интеллектуальные системы*, **10**:1–4 (2006), 449–562.

[2] Blum M.,Kozen D., “On the power of the compass (or, why mazes are easier to search than graphs)”, *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*, 1978, 132–142.

[3] Okhotin A., “Graph-walking automata: from whence they come, and whither they are bound”, *International Conference on Implementation and Application of Automata*, 2019, 10–29.

[4] Насыров А. З., “Об обходе лабиринтов автоматами, оставляющими нестираемые отметки”, *Дискретная математика*, **9**:1 (1997), 123–133.

[5] Голованов А. В., “Об обходе лабиринтов автоматами, оставляющими след в вершинах лабиринта”, *Интеллектуальные системы*, **3**:3–4 (1998), 193–212.

[6] Голубев Д. В., “Об обходе графов автоматами с одной нестираемой краской”, *Интеллектуальные системы*, **4**:1–2 (1999), 243–272.

[7] Демидова А. А., “Автоматный анализ свойств графа быть деревом и псевдодеревом”, *Интеллектуальные системы. Теория и приложения*, **25**:2 (2021), 111–127.

[8] Harary F., Uhlenbeck G. E., “On the number of Husimi trees: I”, *Proceedings of the National Academy of Sciences*, **39**:4 (1953), 315–322.

[9] Paten B. et al., “Cactus graphs for genome comparisons”, *Journal of Computational Biology*, **18**:3 (2011), 469–481.

[10] Zmazek B., Zerovnik J., “Estimating the traffic on weighted cactus networks in linear time”, *Ninth International Conference on Information Visualisation (IV’05)*, 2005, 536–541.

Analysis of cactus graphs using automata: properties and recognition time

Demidova A.A.

This paper is devoted to the study of using automata with erasable colors to determine whether an arbitrary connected plane simple undirected graph is a cactus. An algorithm is given for determining this property, as well as lower and upper bounds of the number of steps that the automaton must take to complete the traversal.

Keywords: Automata, graphs, cactus graphs.

References

[1] Kudryavtsev V. B., Kilibarda G., Ušćumlić Š., “Automata systems in labyrinths”, *Intelligent Systems*, **10**:1–4 (2006), 449–562 (In Russian).

[2] Blum M.,Kozen D., “On the power of the compass (or, why mazes are easier to search than graphs)”, *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*, 1978, 132–142.

[3] Okhotin A., “Graph-walking automata: from whence they come, and whither they are bound”, *International Conference on Implementation and Application of Automata*, 2019, 10–29.

[4] Nasyrov A. Z., “On traversing labyrinths by automata that leave not-erasable marks”, *Discrete mathematics*, **9**:1 (1997), 123–133 (In Russian).

[5] Golovanov A. V., “On traversing labyrinths by automata that leave a trail at the vertices of the labyrinth”, *Intelligent Systems*, **3**:3–4 (1998), 193–212 (In Russian).

[6] Golubev D. V., “On graph traversal by automata with one not-erasable paint”, *Intelligent Systems*, **4**:1–2 (1999), 243–272 (In Russian).

[7] Demidova A. A., “Automaton analysis of the properties of a graph to be a tree and a pseudo-tree”, *Intelligent Systems. Theory and Applications*, **25**:2 (2021), 111–127 (In Russian).

[8] Harary F., Uhlenbeck G. E., “On the number of Husimi trees, I”, *Proceedings of the National Academy of Sciences*, **39**:4 (1953), 315–322.

[9] Paten B. et al., “Cactus graphs for genome comparisons”, *Journal of Computational Biology*, **18**:3 (2011), 469–481.

[10] Zmazek B., Zerovnik J., “Estimating the traffic on weighted cactus networks in linear time”, *Ninth International Conference on Information Visualisation (IV’05)*, 2005, 536–541.