Автоматный анализ свойств графа быть деревом и псевдодеревом

А. А. Демидова[1]

В данной работе исследуется применение автоматов со стираемыми красками для определения свойств связных планарных неориентированных простых графов. Доказываются следующие факты: автоматам хватит двух стираемых красок для того, чтобы определить, является ли граф деревом или псевдодеревом.

Ключевые слова: Автоматы, графы, деревья, псевдодеревья.

1. Введение

Исследования автоматов, обходящих лабиринты, берут исток из работы Шеннона [1], а активное изучение поведения автоматов в лабиринтах и на графах началось после появления работ Деппа [2-3]. Подробный анализ данного направления теории автоматов представлен в работе [4].

Известно, что даже в случае рассмотрения плоских лабиринтов автоматы не могут обойти любой лабиринт без каких-либо вспомогательных средств ([5]); поэтому для того, чтобы проблема завершения обхода являлась разрешимой, необходимо либо вводить определённые ограничения на среду, которую обходит автомат, либо наделять автомат дополнительными возможностями, которые позволяют добиться разрешимости задачи для более широкого класса сред. В частности, автоматы можно усиливать за счёт наделения их способностью оставлять метки на рёбрах или вершинах графа при его обходе. В работах, в которых рассматриваются подобные усиления автоматов, эти отметки называются красками.

В работе [6] установлено существование автомата с нестираемой краской, который может обойти произвольный прямоугольный лабиринт, оставляя метки в его вершинах.

В работе [7] представлен автомат с некоторым количеством красок, «принудительно» оставляющий отметки в посещённых вершинах лабиринта.

В работе [8] рассматриваются автоматы с одной нестираемой краской, оставляющие отметки в вершинах связных простых (без параллельных рёбер и петель) ориентированных графов с ограничением на степень вершин. Автором представлен алгоритм, который позволяет автомату без

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

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

краски осуществить обход графа, который ранее был обойдён автоматом с нестираемой краской.

Псевдодеревом является связный неориентированный граф, содержащий не более одного цикла. Данный объект из теории графов впервые встречается в книге [9] и используется при решении задач линейного программирования. Помимо этого результаты, связанные с псевдодеревьями и их хроматическими числами, приведены в работе [10].

В данной работе рассматриваются автоматы, анализирующие, являются ли обходимые ими связные планарные простые неориентированные графы деревьями и псевдодеревьями. Автоматы передвигаются между вершинами графа и могут ставить стираемые отметки (краски) на рёбрах, по которым проходят.

Основными результатами данной работы являются доказательства следующих фактов: автоматам хватит двух стираемых красок для того, чтобы определить, является ли обходимый граф деревом; более того, их же хватит для того, чтобы определить, является ли он псевдодеревом.

Автор выражает благодарность профессору Э.Э. Гасанову за постановку задачи и помощь в работе.

2. Основные понятия и формулировка результатов

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

Конечным автоматом называется пятёрка $\mathcal{A}=(A, Q, B, \varphi, \psi)$, где A, B и Q являются конечными алфавитами (входным, выходным и состояний), $\varphi : Q \times A \to Q$ – функция переходов, $\psi : Q \times A \to B$ – функция выходов. Автомат с фиксированным начальным состоянием $q_0 \in Q$ – *инициальный*. Пусть A^* и B^* – множества всех слов $a = a(1)...a(n)$ и $b = b(1)...b(n)$ над алфавитами A и B. Система канонических уравнений инициального автомата имеет следующий вид:

$$\begin{cases} q(1) = q_0, \\ q(t+1) = \varphi(q(t), a(t)), \\ b(t) = \psi(q(t), a(t)). \end{cases} \tag{1}$$

Опишем два класса автоматов: $\mathfrak{A}_1$ и $\mathfrak{A}_2$.

Автоматы из класса $\mathfrak{A}_1$ осуществляют обход графа по правилу левой руки и всегда перекрашивают изначально серые рёбра графа в чёрный цвет. Если автомат находится в вершине, степень которой больше 1, то автомат выбирает в качестве следующего ребро, которое находится слева от того, по которому он попал в текущую вершину; если же степень вершины равна 1, то автомат выбирает единственное возможное ребро.

Автоматы из класса $\mathfrak{A}_2$ по умолчанию осуществляют обход графа по правилу левой руки, однако в определённые моменты они могут выбирать не левые рёбра. На вход автомата поступает неполная информация о степени текущей вершины, а также о цветах некоторых рёбер, а именно тех, которые отмечены на Рисунке 1а: текущем ребре, ребре слева, втором ребре слева, втором ребре справа и ребре справа. На этом рисунке, так же как и на всех остальных, стрелкой отмечено, откуда автомат пришёл в текущую вершину. На выходе автомат сообщает, по какому ребру следует идти в следующий момент времени и нужно ли красить его в некоторый цвет, причём в определённых ситуациях автомат может выбирать первое или второе чёрное ребро справа (см. Рисунок 1б).

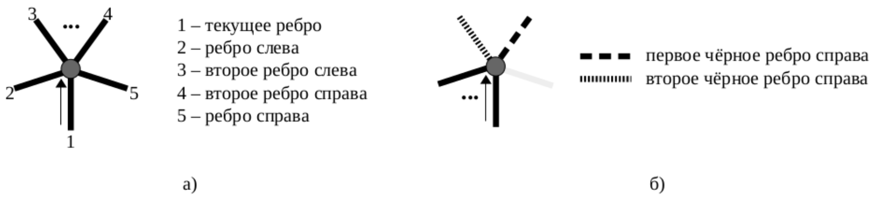

Рис. 1. Некоторые рёбра, инцидентные текущей вершине

В данной работе приводятся следующие теоремы, доказательства которых представлены в разделах 3 и 4 соответственно:

Теорема 1. *Существует автомат $\mathcal{A}_1 \in \mathfrak{A}_1$ с двумя красками, который сможет установить, является ли произвольный граф $\mathbf{g} \in \mathbf{G}$ деревом.*

Теорема 2. *Существует автомат $\mathcal{A}_2 \in \mathfrak{A}_2$ с двумя красками, который сможет установить, является ли произвольный граф $\mathbf{g} \in \mathbf{G}$ псевдодеревом.*

3. Поиск циклов в графе автоматом с двумя красками

Данный раздел содержит доказательство Теоремы 1.

Доказательство. Проведём доказательство следующим образом: сначала опишем компоненты входного и выходного векторов, а также вектора состояний; далее представим алгоритм, которому должен следовать автомат; наконец, приведём обоснование того, что алгоритм работает.

В произвольный момент времени t на вход автомату подаётся вектор $\vec{a}(t) = (a_1(t), ..., a_6(t))$, компоненты которого содержат следующую информацию:

- a_1 – степень вершины ($a_1(t) = 1$, если степень равна 1; $a_1(t) = 2$, если степень равна 2; $a_1(t) = 0$, если степень больше 2);
- a_2 – серые рёбра ($a_2(t) = 0$, если их нет; $a_2(t) = 1$, если есть ровно 1; $a_2(t) = 2$, если их больше 1);
- a_3 – чёрные рёбра ($a_3(t) = 0$, если их нет; $a_3(t) = 1$, если есть ровно 1; $a_3(t) = 2$, если их 2; $a_3(t) = 3$, если их больше 2);
- a_4 – цвет ребра слева ($a_4(t) = 0$, если оно серое; $a_4(t) = 1$, если оно чёрное);
- a_5 – цвет ребра справа ($a_5(t) = 0$, если оно серое; $a_5(t) = 1$, если оно чёрное);
- a_6 – цвет текущего ребра ($a_6(t) = 0$, если оно серое; $a_6(t) = 1$, если оно чёрное).

Выход автомата в момент времени t – вектор $\vec{b}(t) = (b_1(t), b_2(t), b_3(t))$, где:

- b_1 – выбор цвета, в который нужно красить следующее ребро:
  1) $b_1(t) = 0$, если его не надо красить;
  2) $b_1(t) = 1$, если нужен чёрный цвет;
- b_2 – выбор ребра, по которому автомат должен перейти в следующий момент времени:
  1) $b_2(t) = 0$ – выбор ребра, по которому автомат только что пришёл в вершину;
  2) $b_2(t) = 1$ – выбор левого из рёбер, инцидентных вершине;
- b_3 содержит информацию о результате исследования графа автоматом:
  1) $b_3(t) = 0$, если автомат ещё не определил, какими свойствами обладает граф;
  2) $b_3(t) = 1$, если автомат установил, что граф является деревом;
  3) $b_3(t) = 2$, если автомат определил, что граф не является деревом.

Состояние автомата в момент t определяется лишь одной компонентой $q(t)$, содержащей информацию о том, красил ли автомат только что ребро ($q(t) = 0$, если нет, и $q(t) = 1$ в противном случае).

Мы не будем приводить строгое описание функций φ и ψ.

Алгоритм, которому должен следовать автомат $\mathcal{A}_1$, можно описать следующим образом:

1) Автомат осуществляет движение по графу по правилу левой руки. Он красит рёбра, по которым проходит, в чёрный цвет.

2) Если в некоторый момент обхода графа автомат видит среди рёбер, инцидентных текущей вершине, хотя бы два чёрных ребра, одно из которых он только что красил, то обнаружено наличие цикла. В таком случае граф не является деревом, и автомат завершает обход.

3) В случае отсутствия циклов в графе автомат завершает обход в тот момент, когда впервые переходит в вершину степени 1 по ребру, которое уже было покрашено в чёрный цвет ранее. Граф является деревом.

Перейдём к обоснованию работы алгоритма.

Автомат осуществляет обход графа по правилу левой руки. Если в графе присутствует цикл, то его наличие будет устанавливаться в ситуациях, когда автомат покрасил некоторое ребро в чёрный цвет и увидел среди рёбер, инцидентных текущей вершине, хотя бы ещё одно чёрное ребро. Таким образом, в случае наличия в графе цикла автомат с двумя красками сможет его обнаружить и прекратить обход графа.

В случае, если цикла в графе нет, автомату нужно будет определить, когда это дерево будет обойдёно полностью. Возможны два случая:

1) Степень корня равна 1;

2) Степень корня больше 1.

Рассмотрим первый случай. Ребро, инцидентное корню, было покрашено в чёрный цвет в начальный момент времени. Когда автомат во время обхода переходит в произвольную вершину степени 1, единственное инцидентное ребро ей имеет чёрный цвет. Автомат обладает информацией о том, красил ли он только что это ребро. Если красил, то текущая вершина является листом дерева; в противном случае автомат во второй раз оказался в корне. К моменту возвращения в корень автомат полностью обошёл дерево по правилу левой руки. Таким образом, автомат завершит обход графа.

Теперь перейдём к рассмотрению второго случая. Если степень корня больше 1, то автомат прекратит обход после второго посещения первого листа. Второе посещение первого листа можно отличить от первых попаданий во все остальные листья аналогично второму посещению корня в случае, если степень корня равна 1. К указанному моменту автомат полностью обойдёт дерево по правилу левой руки и так же, как и в первом случае, завершит обход графа.

□

4. Анализ псевдодеревьев автоматом с двумя красками

Данный раздел содержит доказательство Теоремы 2.

Доказательство. Доказательство приведём в том же порядке, что и в Теореме 1.

В произвольный момент времени t на вход автомату подаётся вектор $\vec{a}(t) = (a_1(t), ..., a_7(t))$, компоненты которого содержат следующую информацию:

- a_1 – степень вершины ($a_1(t) = 1$, если степень равна 1; $a_1(t) = 2$, если степень равна 2; $a_1(t) = 3$, если степень равна 3; $a_1(t) = 4$, если степень равна 4; ; $a_1(t) = 5$, если степень равна 5; $a_1(t) = 0$, если степень больше 5);
- a_2 – серые рёбра ($a_2(t) = 0$, если их нет; $a_2(t) = 1$, если есть ровно 1; $a_2(t) = 2$, если их больше 1);
- a_3 – чёрные рёбра ($a_3(t) = 0$, если их нет; $a_3(t) = 1$, если есть ровно 1; $a_3(t) = 2$, если их 2; $a_3(t) = 3$, если их больше 2);
- a_4 – цвет ребра слева ($a_4(t) = 0$, если оно серое; $a_4(t) = 1$, если оно чёрное);
- a_5 – цвет ребра справа ($a_5(t) = 0$, если оно серое; $a_5(t) = 1$, если оно чёрное);
- a_6 – цвет текущего ребра ($a_6(t) = 0$, если оно серое; $a_6(t) = 1$, если оно чёрное);
- a_7 – цвет второго слева ребра ($a_7(t) = 0$, если оно серое; $a_7(t) = 1$, если оно чёрное).

Выход автомата в момент времени t – вектор $\vec{b}(t) = (b_1(t), b_2(t), b_3(t))$, где:

- b_1 – выбор цвета, в который нужно красить следующее ребро:
  1) $b_1(t) = 0$, если его не надо красить;
  2) $b_1(t) = 1$, если нужен серый цвет;
  3) $b_1(t) = 2$, если нужен чёрный цвет;
- b_2 – выбор ребра, по которому автомат должен перейти в следующий момент времени:
  1) $b_2(t) = 0$ – выбор ребра, по которому автомат только что пришёл в вершину;
  2) $b_2(t) = 1$ – выбор левого из рёбер, инцидентных вершине;
  3) $b_2(t) = 2$ – выбор второго слева из рёбер, инцидентных вершине, если её степень больше 2;
  4) $b_2(t) = 3$ – выбор второго справа чёрного ребра среди инцидентных вершине, если её степень больше 2;
  5) $b_2(t) = 4$ – выбор правого из чёрных рёбер, исходящих из вершины, если её степень больше 2;
  6) $b_2(t) = 5$ – выбор правого из рёбер, инцидентных вершине;
- b_3 содержит информацию о результате исследования графа автоматом:
  1) $b_3(t) = 0$, если автомат ещё не определил, какими свойствами обладает граф;
  2) $b_3(t) = 1$, если автомат установил, что граф является деревом;
  3) $b_3(t) = 2$, если автомат установил, что граф является псевдодеревом с одним циклом;
  4) $b_3(t) = 3$, если граф обнаружил, что в графе больше одного цикла.

Состояние автомата в момент t определяется вектором $\vec{q}(t) = (q_1(t), ..., q_8(t))$ со следующими компонентами:

- q_1 – направление обхода графа ($q_1(t) = 0$, если обход осуществляется по правилу левой руки, и $q_1(t)$=1, если по правилу правой руки);
- q_2 – красил ли автомат только что ребро ($q_2(t) = 0$, если нет, и $q_2(t) = 1$ в противном случае);
- q_3 – стадия, на которой находится автомат в исследовании свойств графа. Стадиями в данном случае могут быть обход графа в поисках цикла, второй обход по циклу в поисках ветвлений и т.д.;

- q_4 – красил ли автомат некоторое ребро в серый цвет до текущего момента обхода;
- q_5 – наличие ветвлений в вершинах цикла, отличных от той, в которой было установлено существование цикла. До перехода к стадии поиска ветвлений во время второго прохода по циклу $q_5(t) = 0$. Автомату нужна информация только о наличии хотя бы одного ветвления. Если при повторном проходе по циклу по правилу левой руки автомат оказывается в вершине, степень которой больше 2, то он обнаруживает наличие ветвления. Ветвление может быть левым или правым: левое обнаруживается ($q_5(t) = 2$), если некоторой вершине инцидентны хотя бы 3 чёрных ребра; если в вершине есть правое ветвление ($q_5(t) = 1$), то правое ребро должно быть серого цвета. Приоритетом обладают левые ветвления, так что, если автомат обнаружит сначала некоторое правое ветвление, а потом – левое, то в результате $q_5(t) = 2$;
- q_6 – наличие левого ветвления в вершине, в которой было обнаружено существование цикла в графе. До перехода к стадии поиска ветвлений во время второго прохода по циклу $q_6(t) = 0$. $q_6(t) = 1$, если либо соответствующей вершине инцидентны по крайней мере 2 чёрных ребра, либо левое ребро является серым;
- q_7 – наличие правого ветвления в вершине, в которой было обнаружено существование цикла в графе. До перехода к стадии поиска ветвлений во время второго прохода по циклу $q_7(t) = 0$. $q_7(t) = 1$, если в соответствующей вершине правое ребро – серое;
- q_8 – индикатор того, встречались ли автомату вершины, степень которых больше 2.

Мы не будем приводить строгое описание функций φ и ψ.

Алгоритм, которому следует автомат $\mathcal{A}_2$, можно описать следующим образом:

1) Автомат осуществляет движение по графу по правилу левой руки, нанося чёрную краску на все рёбра, по которым проходит, пока не обнаружит наличие цикла. Если к моменту второго посещения первого листа цикл не был обнаружен, то граф является деревом.

2) Автомат перекрасит ребро, по которому только что прошёл, в серый цвет. После этого он вернётся в вершину, в которой было обнаружено наличие цикла, и осуществит второй обход цикла с целью установить, были ли в цикле левые (лежащие вне грани, ограниченной циклом, и раскрашенные при первом проходе по циклу) и

правые (лежащие внутри грани, ограниченной циклом, и не обойдённые при первом проходе по циклу) ответвления. Поскольку последовательности рёбер при двух проходах по циклу будут совпадать, то первым левым серым ребром, которое встретится автомату в ходе движения по правилу левой руки, будет то, которое он перекрашивал в серый цвет.

3) Возможны следующие ситуации:

   а) Если автомат не обнаружил ни левых, ни правых ветвлений, то граф является циклом, и, следовательно, псевдодеревом. Автомат завершает обход.

   б) Если ветвления были, но среди них не было левых (в том числе отсутствовало левое ветвление в вершине, в которой было обнаружено наличие цикла), автомат, двигаясь по правилу левой руки в том же направлении, что было первоначально, перекрасит цикл обратно в серый цвет. Вернувшись в вершину цикла, из которой был начат обход, автомат начнёт поиск вершины, степень которой больше 2. Перейдя в соответствующее правое ветвление, автомат начнёт красить рёбра в чёрный цвет. Ситуация сведена к случаю, когда обход графа начинается из вершины, лежащей внутри грани, ограниченной циклом.

   в) Если левые ответвления были, то дальнейшие действия автомата зависят от наличия левого и правого ветвлений в вершине, в которой было обнаружено наличие цикла. Если левое есть, а правого нет, то автомат просто переходит по обнаруженному им серому ребру, не крася его; во всех остальных случаях автомат дополнительно перекрашивает соответствующее ребро в чёрный цвет. Далее, если левое есть, то автомат обходит его и перекрашивает самое правое из ведущих в него рёбер в серый цвет. Если левое в графе отсутствует, то автомат идёт по циклу по правилу левой руки и ищет первое левое ветвление, в котором красит в серый цвет самое правое из рёбер, ведущих в соответствующее ветвление. Среди всех рёбер, инцидентных соответствующей вершине цикла, это ребро является вторым чёрным справа. После этого автомат возвращается по тому же ребру, уже не нанося на него краску. Автомат переходит по второму слева ребру и начинает движение по циклу по правилу левой руки в направлении, противоположном первоначальному. Автомат обойдёт цикл и все правые ветвления и установит, что обход окончен, когда окажется в вершине, которой инциденты хотя бы два чёрных ребра и ровно одно серое, причём левое ребро для автомата будет чёрным, а правое – серым.

4) Если в некоторый момент времени автомат обнаружил существование ещё одного цикла, то граф не является псевдодеревом. Автомат завершает обход.

Перейдём к обоснованию работы алгоритма.

Из Теоремы 1 следует, что автомату хватит двух красок для того, чтобы установить отсутствие циклов в графе. Далее будем предполагать, что в графе есть один цикл.

Общий вид рассматриваемого графа представлен на Рисунке 2. Будем считать, что цикл имеет длину n; v_i $(i = \overline{1,n})$ – вершины цикла, занумерованные в том порядке, в котором автомат посещает их в первый раз; $G_{i,j}$ $(i = \overline{1,n},\ j = \overline{1,2})$ – поддеревья, исходящие из вершин цикла, такие, что для $\forall i = \overline{1,n}$ $G_{i,1}$ лежит вне грани, ограниченной рассматриваемым циклом, а $G_{i,2}$ располагается в ней. Подразумевается, что на рисунке рёбра, ведущие из v_i в $G_{i,1}$ и $G_{i,2}$ $(i = \overline{1,n})$, на самом деле могут обозначать наличие более одного ребра.

Автомат установит наличие цикла, когда окажется в вершине, которой инцидентны по крайней мере 2 чёрных ребра, одно из которых он только что красил. Это соответствует переходу из вершины v_n в v_1.

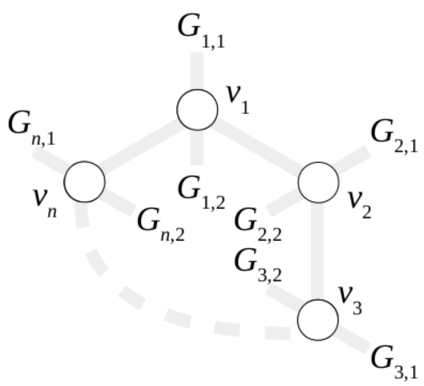

Рис. 2. Общий вид рассматриваемого графа

Возможны следующие случаи:

1) Автомат начал обход графа из $G_{1,1}$ (см. Рисунок 3а);

2) Автомат начал обход графа из $G_{1,2}$ (см. Рисунок 3б);

3) Автомат начал обход графа из вершины v_1, принадлежащей циклу. В начальный момент времени он мог либо приступить к движению

по циклу (см. Рисунок 3в), либо перейти из v_1 в $G_{1,j}$ $(j = 1, 2)$. Эти случаи сводятся к первым двум.

Порядок обхода графа до обнаружения наличия цикла представлен на Рисунке 3. В зависимости от того, из какого $G_{1,j}$, $j = \overline{1, 2}$, автомат начинал обход графа, при движении по циклу по правилу левой руки он также будет обходить либо $G_{i,1}$ $(i = \overline{2, n})$, лежащие вне грани, ограниченной рассматриваемым циклом, либо $G_{i,2}$ $(i = \overline{2, n})$, лежащие внутри этой грани. Следует отметить, что к моменту начала движения по циклу соответствующий $G_{1,j}$ $(j = \overline{1, 2})$ может быть обойдён не полностью, что отражено на Рисунках 3а и 3б.

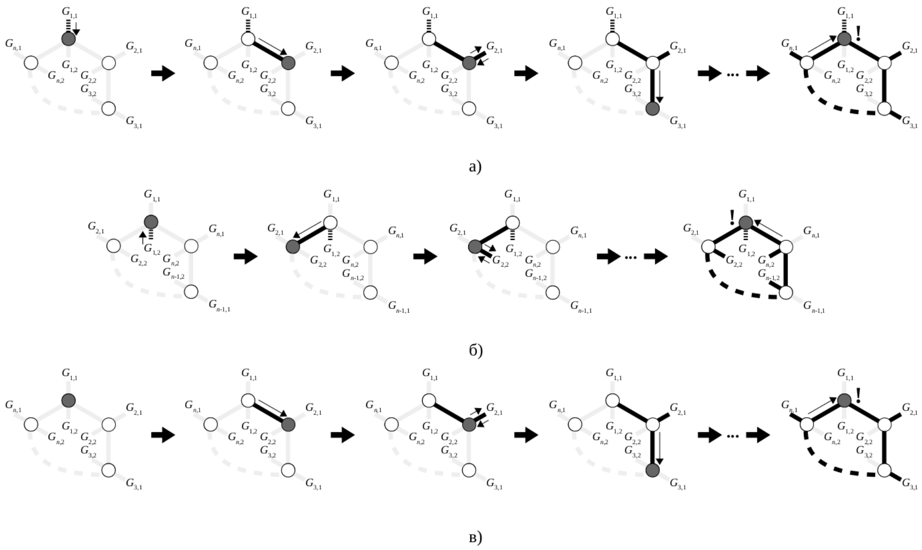

Рис. 3. Порядок обхода графа до обнаружения наличия цикла

Будем рассматривать случай, когда движение по псевдодереву начинается из $G_{1,1}$ (возможно, $G_{1,1}$ состоит лишь из вершины v_1).

Последовательность действий после обнаружения цикла отражена на Рисунке 4. Автомат перекрасит ребро, которое только что красил в чёрный цвет, в серый цвет (это ребро, соединяющее v_1 и v_n), вернётся обратно в v_1 и отдельно запомнит, увидел ли он в ней левые и правые ветвления. Далее автомат перейдёт по самому правому из чёрных рёбер в v_2 и начнёт движение по циклу по правилу левой руки в направлении, совпадающем с первоначальным, с целью установить, были ли в цикле левые (лежащие вне грани, ограниченной циклом, и раскрашенные при первом проходе по циклу) и правые (лежащие внутри грани, ограниченной циклом, и не обойдённые при первом проходе по циклу) ветвления.

Поскольку последовательности рёбер при двух проходах по циклу будут совпадать, то первым серым ребром, которое встретится автомату в ходе движения по правилу левой руки, будет $v_n v_1$. При этом автомат будет помнить, что он уже перекрашивал одно из рёбер графа в серый цвет.

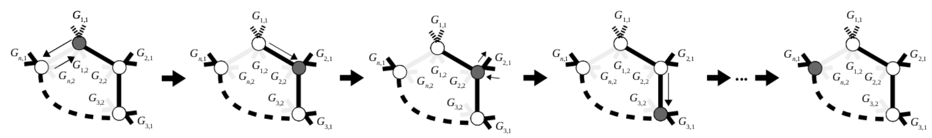

Рис. 4. Повторный обход цикла с целью выявления левых и правых ветвлений

К этому моменту автомат ещё не обошёл $G_{i,2}$, $i = \overline{1, n}$, и, возможно, не полностью обошёл $G_{1,1}$. Автомат уже обладает информацией о том, были ли в цикле левые и правые ветвления, а также о наличии в графе $G_{1,1}$ и $G_{1,2}$.

Если в ходе повторного движения по циклу автомат вообще не попадал в вершины степени больше 2, то граф является циклом и, соответственно, псевдодеревом. При этом в результате повторного обхода цикла автомат всё же попадёт в ситуацию, когда ребро справа будет белым. Это произойдёт при возвращении в вершину v_n. Автомат определит, что это не признак правого ветвления: во-первых, степень вершины равна 2; во-вторых, ему вообще не попадались вершины степени больше 2.

Если ветвления были, но среди них не было левых (в том числе отсутствовал $G_{1,1}$), то существует $i \in \overline{1, n}$: $G_{i,2} \neq \varnothing$. В таком случае автомат может осуществить следующие действия: он продолжит движение по циклу по правилу левой руки в том же направлении, которое было первоначально, но теперь будет перекрашивать рёбра цикла в серый цвет. Когда автомат перекрасит весь цикл, он увидит вокруг только серые рёбра, после чего должен будет начать движение по циклу по правилу левой руки в направлении, совпадающем с первоначальным. Однако красить что-то в чёрный цвет он начнёт только после того, как увидит слева $G_{i,2} \neq \varnothing$ (это произойдёт тогда, когда автомат окажется в вершине, степень которой больше 2). Автомат перейдёт по правилу левой руки в $G_{i,2}$ и начнёт красить рёбра в чёрный цвет. Таким образом, ситуация сведена в случаю, когда автомат начинает движение по графу из поддерева, лежащего внутри грани, ограниченной циклом, причём хотя бы одно такое поддерево существует.

Остальные ситуации можно разбить на 3 случая в зависимости от наличия в графе $G_{1,1}$ и $G_{1,2}$:

1) $G_{1,1}$ в графе отсутствует, но есть другие левые ветвления;

2) В графе есть как $G_{1,1}$, так и $G_{1,2}$;

3) В графе есть $G_{1,1}$, но нет $G_{1,2}$.

Разбор каждого из этих случаев начинается с того момента, когда автомат прошёл по циклу по второму разу по правилу левой руки в направлении, совпадающем с первоначальным, и увидел слева серое ребро (это будет ребро, соединяющее вершины v_n и v_1).

1) $G_{1,1}$ **в графе отсутствует, но есть другие левые ветвления (см. Рисунок 5).**
Автомат перекрасит увиденное им серое ребро, соединяющее вер-

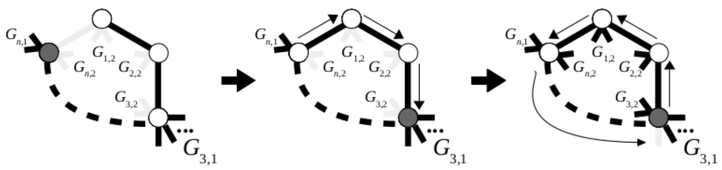

Рис. 5. Первый случай

шины v_n и v_1, в чёрный цвет, после чего продолжит движение по правилу левой руки в направлении, совпадающем с первоначальным, в поисках первого левого ветвления. Поскольку случай, когда автомат при повторном обходе цикла не обнаружил ни одного левого ветвления, уже был разобран ранее, теперь хотя бы одно левое ветвление есть. Без ограничения общности будем считать, что это $G_{3,1}$. Автомат дойдёт до вершины, которой инцидентны по крайней мере 3 чёрных ребра, одно из которых относится к циклу, второе ведёт в поддерево $G_{3,1}$, которое было обойдено ранее, а третье либо ведёт в то же поддерево, либо относится к циклу. Поскольку автомат знает цвет текущего ребра, а также цвета первого и второго слева, он сможет понять, что дошёл до v_3. Автомат раскрасит второе справа чёрное ребро, являющееся самым правым из ведущих в $G_{3,1}$, в серый цвет, после чего начнёт движение по циклу по правилу левой руки в направлении, противоположном первоначальному. При этом автомат обойдёт и $G_{3,2}$, $G_{2,2}$, $G_{1,2}$, $G_{n,2}$,..., $G_{4,2}$. Когда автомат окажется в вершине, которой инцидентно лишь одно серое ребро, являющееся при этом самым правым, и хотя бы 2 чёрных ребра, он поймёт, что вернулся в v_3. В таком случае самое левое ребро будет чёрным, а самое правое – серым. Когда автомат шёл по циклу в направлении, противоположном первоначальному, то в вершинах v_2, v_1, v_n,...,v_4 самыми правыми для него были некоторые чёрные ребра, а при обходе $G_{i,2}$ ($i \in \{1, 2, 4, ..., n\}$) самое правое могло быть серым, но тогда и левое было бы серым. Следовательно, ситуация, когда самое левое – чёрное, а самое правое – серое, означает, что автомат вернулся к ребру, которое перекрашивал в серый цвет. Более того, к этому моменту обход псевдодерева завершён.

2) **В графе есть как $G_{1,1}$, так и $G_{1,2}$ (см. Рисунок 6).**

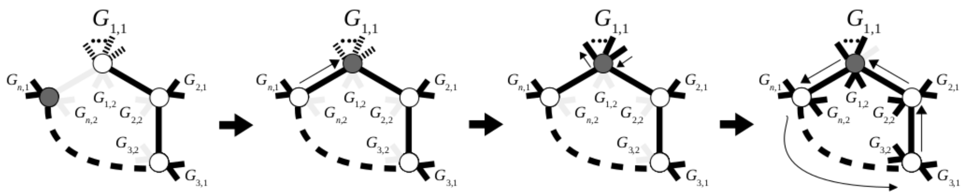

Рис. 6. Второй случай

Автомат перекрасит увиденное им серое ребро, соединяющее вершины v_n и v_1, в чёрный цвет, после чего по правилу левой руки обойдёт $G_{1,1}$. Возможно, что данное поддерево уже было частично обойдено до попадания в цикл. Когда автомат обойдёт $G_{1,1}$, он по правилу левой руки должен будет перейти из v_1 в v_2, и при этом правее этого ребра будет лежать ещё не обойдённый $G_{1,2}$. При обходе $G_{1,1}$ по правилу левой руки автомат не может попасть в ситуацию, когда левое ребро, по которому он должен пройти, имеет чёрный цвет, а следующее за ним среди рёбер, инцидентных той же вершине, имеет серый цвет, – следовательно, если автомат увидит такую пару рёбер, ему станет известно, что $G_{1,1}$ обойдён полностью, а второе из увиденных им рёбер является самым левым из рёбер, ведущих из v_1 в $G_{1,2}$. Как только автомат установит, что $G_{1,1}$ обойдён полностью, он перекрасит в серый цвет то ребро из $G_{1,1}$, по которому только что прошёл, пройдёт обратно по этому ребру в v_1 и начнёт движение по циклу по правилу левой руки в направлении, противоположном первоначальному. При этом автомат обойдёт и $G_{1,2}$, $G_{n,2}$, $G_{n-1,2}$,..., $G_{3,2}$, $G_{2,2}$. Когда автомат вернётся в вершину, которой инцидентно лишь одно серое ребро, являющееся при этом самым правым, и хотя бы 2 чёрных ребра, он установит, что вернулся в v_1. Так же, как и в первом случае, автомат завершит обход.

3) **В графе есть $G_{1,1}$, но нет $G_{1,2}$ (см. Рисунок 7).**

Как и во втором случае, автомат должен пройти по увиденному им

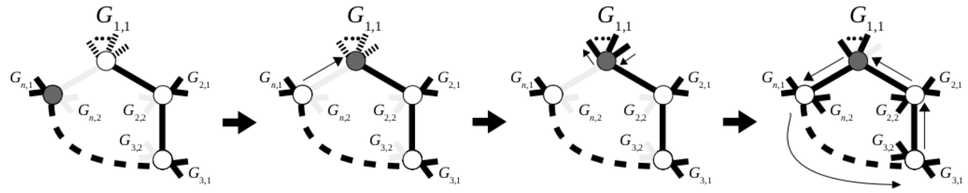

Рис. 7. Третий случай

серому ребру, соединяющему v_n и v_1, и обойти $G_{1,1}$, однако в случае отсутствия в графе $G_{1,2}$ по правилу левой руки за ребром v_1v_2 будет следовать v_1v_n. Поскольку при втором обходе цикла можно было запомнить, есть ли в графе $G_{1,1}$ и $G_{1,2}$, автомат не станет красить ребро, увиденное им серое ребро, соединяющее v_n и v_1, в чёрный цвет, в результате чего признак окончания обхода $G_{1,1}$ останется прежним. Так же, как и во втором случае, автомат перекрасит в серый цвет последнее ребро из $G_{1,1}$, по которому он проходил, после чего перейдёт по второму ребру слева (из v_1 в v_n), покрасив соответствующее ребро, и начнёт движение по циклу по правилу левой руки в направлении, противоположном первоначальному. При этом автомат обойдёт и $G_{n,2}$, $G_{n-1,2}$,..., $G_{3,2}$, $G_{2,2}$. Когда автомат попадёт в вершину, которой инцидентно лишь одно серое ребро, являющееся при этом самым правым, и хотя бы 2 чёрных ребра, он установит, что это v_1. Так же, как и в первых двух случаях, автомат завершит обход.

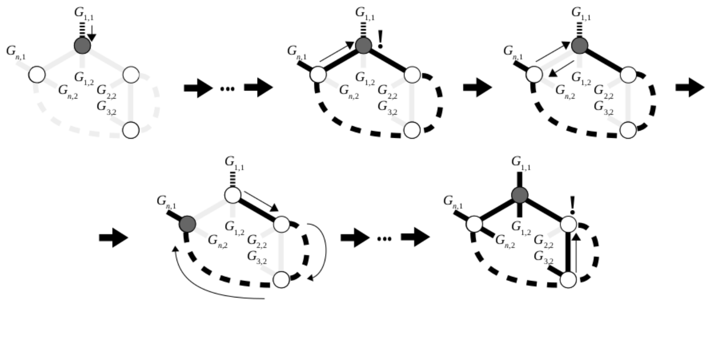

Рис. 8. Обход графа с двумя циклами, имеющими общее ребро

Если граф, обход которого осуществляет автомат, содержит более одного цикла, то при обнаружении второго цикла автомат установит, что граф не является псевдодеревом. Допустим, что в графе есть два цикла, и приведём примеры. Если они имеют общее ребро, то наличие второго

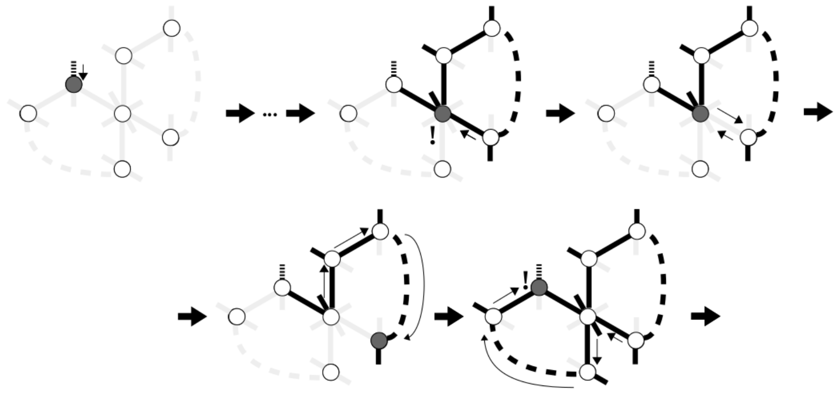

Рис. 9. Обход графа с двумя циклами, имеющими общую вершину

цикла может быть обнаружено тогда, когда автомат будет обходить поддеревья, лежащие внутри грани, ограниченной циклом (см. Рисунок 8). Если они имеют общую вершину, то наличие цикла может быть обнаружено после повторного прохода по циклу (см. Рисунок 9).

Случай, когда автомат начинает обход графа из $G_{1,2}$, разбирается аналогично.

□

5. Выводы

В данной работе представлены два алгоритма для автоматов, осуществляющих обход графов. Доказано, что автоматы, действующие согласно этим алгоритмам, смогут установить, является ли граф деревом или псевдодеревом, с использованием двух стираемых красок.

Список литературы

[1] Shannon Cl. E., "Presentation of a maze-solving machine", *Cybernetics Trans. of the 8th Conf. of the Josiah Macy Jr. Found*, 1951, 173–180.

[2] Döpp K., "Automaten in Labyrinthen I", *EIK*, **7**:2 (1971), 79–94.

[3] Döpp K., "Automaten in Labyrinthen II", *EIK*, **7**:3 (1971), 167–190.

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

[5] Budach L., "Automata and labyrinths", *Mathematische Nachrichten*, **86**:1 (1978), 195–282.

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

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

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

[9] Dantzig G., *Linear programming and extensions*, Princeton university press, 1963.

[10] Ищенко Р. А., “О разложении графов на подграфы специального вида”, *Интеллектуальные системы. Теория и приложения*, **20**:4 (2016), 184–192.

Automaton analysis of the properties of a graph to be a tree and a pseudo-tree

Demidova A.A.

In this paper, we investigate the use of automata with erasable colors to determine the properties of connected planar undirected simple graphs. The following facts are proved: two erasable colors are enough for automata to determine whether the graph is a tree or a pseudo-tree.

Keywords: Automata, graphs, trees, pseudotrees.

References

[1] Shannon Cl. E., “Presentation of a maze-solving machine”, *Cybernetics Trans. of the 8th Conf. of the Josiah Macy Jr. Found*, 1951, 173–180.

[2] Döpp K., “Automaten in Labyrinthen I”, *EIK*, **7**:2 (1971), 79–94.

[3] Döpp K., “Automaten in Labyrinthen II”, *EIK*, **7**:3 (1971), 167–190.

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

[5] Budach L., “Automata and labyrinths”, *Mathematische Nachrichten*, **86**:1 (1978), 195–282.

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

[7] 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).

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

[9] Dantzig G., *Linear programming and extensions*, Princeton university press, 1963.

[10] Ischenko R. A., “Decomposition of graphs into subgraphs of a special form”, *Intelligent Systems. Theory and Applications*, **20**:4 (2016), 184–192 (In Russian).