Оглавление

1. Графовые грамматики и машины: случай ограниченного ветвления

Напомним, что контекстно-свободной графовой грамматикой называется следующая совокупность: конечное множество нетерминалов, начальный нетерминал, конечное множество терминалов, конечное множество правил вывода. Правило вывода имет вид $\xi \longrightarrow G$, где ξ — нетерминал, а G — граф с висячими (т.е. инцидентными только одной вершине) ребрами, число которых обозначим через $V(G)$. В G, как и во всех рассматриваемых графах, могут быть кратные ребра, но нет петель. Каждая вершина графа G помечена терминалом или нетерминалом. Ребра при каждой вершине из G перенумерованы (ребро может иметь различные номера относительно концов). Независимо от этой нумерации указана нумерация висячих ребер графа G, называемая нумерацией соответствия. Применение правила $\xi \longrightarrow G$ к некоторому графу заключается в замене некоторой его вершины a, помеченой ξ и имеющей степень ветвления $V(G)$, на граф G, при этом каждое ребро l инцидентное a отождествляется с висячим ребром из G, имеющим тот же номер в нумерации соответствия, что и l относительно a. Граф A назовем выводимым в грамматике, если все его вершины помечены терминалами и он может быть выведен за конечное число шагов из одновершинного графа, помеченного начальным нетерминалом. Докажем следующую лемму.

ЛЕММА. *Для любой грамматики* Γ *существует эквивалентная ей грамматика* Γ_1 *такая, что в ней нет правил вида* $\xi \longrightarrow E$, *где* E — *одновершинный граф, вершина которого помечена нетерминалом.*

ДОКАЗАТЕЛЬСТВО. Рассмотрим ориентированный граф O, вершинами которого являются нетерминалы G, а два нетерминала ξ_1 и ξ_2 соединены направленным ребром $\overrightarrow{\xi_1\xi_2}$ если в Γ существует правило $\xi_1 \longrightarrow \xi_2$, где ξ_2 — одновершинный граф с вершиной, помеченной ξ_2. O

разбивается на частично упорядоченные классы взаимно достижимых вершин. Для каждого класса все его элементы заменим одним нетерминалом. Оставшиеся правила вида $\xi_1 \longrightarrow \xi_2$ ликвидируем, добавив для каждого правила $\xi_2 \longrightarrow G$ правило $\xi_1 \longrightarrow G$. Очевидно, что полученная грамматика эквивалентна Γ. Лемма доказана.

Из доказанной леммы следует, что можно рассматривать грамматики только с такими правилами, которые увеличивают число вершин выводимого графа. Будем считать, что все рассматриваемые далее грамматики такие.

Выводимые в одной грамматике графы могут представлять собой достаточно богатое семейство. Например, как показывает следующая теорема, любое выводимое семейство можно снабдить гамильтоновыми циклами так что новое семейство тоже будет выводимо.

ТЕОРЕМА 1. *Для любой графовой грамматики* Γ *существует грамматика* Γ_1 *такая, что для любого графа* G*, выводимого в* Γ*, существует граф* G_1*, выводимый в* Γ_1*, со следующими свойствами. Вершины* G_1 *взаимнооднозначно соответствуют вершинам* G*, включая разметку. Множество ребер* G_1 *есть объединение множества ребер, соответствующих ребрам* G *с той же нумерацией, и ребер, составляющих гамильтонов цикл в* G_1*. Ребра этого цикла имеют максимальный при каждой вершине номер в одном из направлений обхода и на 1 меньший в противоположном.*

ДОКАЗАТЕЛЬСТВО. В графах, являющихся правыми частями правил Γ с начальным нетерминалом в левой части, добавим ребра, составляющие гамильтонов цикл. В правых частях других правил добавим ребра, составляющие гамильтонову цепь, и два висячих ребра по краям этой цепи. Выбрав направление обхода, присвоим номера добавленным ребрам в соответствии с последним условием теоремы. Получим грамматику Γ_1. Справедливость утверждения теоремы легко доказывается индукцией по выводу G. Теорема 1 доказана.

В [1] доказано, что проблема существования в графе гамильтонова цикла NP-полна для произвольного графа.

А.О.Слисенко в [2] доказал, что для графов, выводимых в одной грамматике, эта проблема решается за полиноминальное от размера графа время и степень полинома зависит от грамматики. Наша цель — обобщить идеи, использованные А.О.Слисенко.

Мы обобщаем проблему существования гамильтонова цикла следующим образом. Рассматриваем конечные связные графы ограниченной степени ветвления, в которых для каждой вершины задана нумерация инцидентных ей ребер. В каждой вершине графа стоит буква из некоторого конечного алфавита (разные вершины могут быть одинаково помечены). Одна из вершин выделена в качестве начальной. Мы будем рассматривать машины Тьюринга специального вида, работающие на таких графах. Опишем детерминированную машину, работающую на графах со степенью ветвления $\leq N$, у которой число посещений каждой вершины $\leq K$. Она задается набором $\langle A, Q, q_0, F \rangle$, где A — конечный алфавит, включачащюий в себя алфавит разметки графов и дополнительный рабочий алфавит, Q — конечное множество состояний машины, q_0 — выделенное начальное состояние, F — программа. Программа F это функция. Область ее определения это множество пятерок $\langle q, a, n, m, k \rangle$, где $q \in Q$ — текущее состояние машины, $a \in A$ — буква, которой помечена текущая вершина, $n \leq N$ — степень ветвления текущей вершины, $m \leq n$ — номер относительно текущей вершины того ребра, по которому машина последний раз пришла в эту вершину (перед началом работы $m = 0$), $k \leq K + 1$ — число посещений текущей вершины с учетом текущего посещения. Если $k \leq K$, а q — незаключительное состояние, то значением F является набор $\langle q', a', n' \rangle$. Здесь $q' \in Q$ — новое состояние машины, $a' \in A$ — новая пометка текущей вершины, $n' \leq n$ — номер ребра, по которому машина сдвигается из текущей вершины. Если $k = K + 1$, значением является безрезультатная остановка. Если q — заключительное состояние, $k \leq K$, то машина допускает граф. В начале работы машина находится в начальной вершине и в начальном состоянии. Потом она естественным образом действует по своей программе и заканчивает работу.

ТЕОРЕМА 2. *Для любого связного графа существует машина, которая, работая на нем, обходит все его вершины и кончает работу в начальной вершине.*

ДОКАЗАТАЛЬСТВО. Опишем машину $\mathcal{M}$, которая, обходя все вершины графа, одновременно строит его остовное дерево. Пусть S_0 — степень ветвления при начальной вершине a_0. Работа машины $\mathcal{M}$ состоит из S_0 этапов. На n-ом этапе она обходит все еще не посещенные вершины, соединимые с a_0 путем, проходящим через ребро с номером n и не проходящим по уже посещенным вершинам. Обозначим множество таких вершин через C. Перед началом n-ого этапа a_0 помечается числом n, чтобы помнить номер этапа. $\mathcal{M}$ смещается из a_0 по n-му ребру и попадает в вершину a_1. Если в a_1 $\mathcal{M}$ уже была, то a_1 помечена некоторым числом. В этом случае $\mathcal{M}$ смещается обратно в a_0 и n-ый этап заканчивается. В противном случае $\mathcal{M}$ запоминает в разметке a_1 номер ребра $a_1 a_0$ относительно a_1 и обходит множество $C \setminus \{a_1\}$ вершин, соединимых с a_1 путями, не проходящими через уже посещенные вершины. Для этого рекурсивно запускается та же процедура, что и для a_0. Закончив обход, $\mathcal{M}$ возвращается в a_0 и n-ый этап заканчивается. В силу связности графа любая его вершина соединима с начальной и будет посещена на некотором этапе. Поскольку из каждой вершины рекурсивная процедура запускается один раз, то каждое ребро будет пройдено не более четырех раз. Поэтому число посещений любой вершины ограничено. Теорема 2 доказана.

Каждой машине соответствует допускаемое ей множество графов. Назовем множества, которые распознаются некоторой машиной, распознаваемыми.

ТЕОРЕМА 3. *Дополнение распознаваемого множества до множества всех графов с той же максимальной степенью ветвления и с разметкой в том же алфавите распознаваемо. Объединение и пересечение двух распознаваемых множеств, алфавиты разметки которых совпадают, являются распознаваемыми.*

ДОКАЗАТЕЛЬСТВО. Машина, распознающая дополнение, получается из машины, распознающей само множество, переменой местами остановки в заключительном состоянии и безрезультатной остановки. В силу равенства $M_1 \bigcap M_2 = \overline{\overline{M_1} \bigcup \overline{M_2}}$ утверждение для пересечения следует из утверждений для дополнения и объединения. Опишем машину $\mathcal{M}$, распознающую $M_1 \bigcup M_2$, где M_1 и M_2 распознаются машинами $\mathcal{M}_1$ и $\mathcal{M}_2$.

Сначала $\mathcal{M}$ действует в соответствии с программой машины $\mathcal{M}_1$, модифицированной так, чтобы запомнить в разметке первоначальную разметку графа и начальную вершину. Если $\mathcal{M}_1$ не допускает граф, $\mathcal{M}$ начинает обходить все вершины графа, пока не найдет начальную. После этого $\mathcal{M}$ действует в сответствии с программой $\mathcal{M}_2$. Теорема 3 доказана.

Пусть M — произвольное множество графов, алфавит разметки которых есть прямое произведение алфавитов A_1 и A_2. Проекцией множества M на A_1 назовем множество графов, размеченных алфавитом A_1, для которых можно так добавить разметку алфавитом A_2, что получившийся граф принадлежит M. Обобщением M на A_1 назовем множесто графов, размеченных в A_1, для которых при любой разметке алфавитом A_2 полученный граф принадлежит M. Будем рассматривать иерархию классов предикатов на графах наподобие иерархии Мейера-Стокмейера. Базовыми предикатами являются распознаваемые множества. Обозначим этот класс через $\sigma_0 = \pi_0$. Определим по индукции другие классы иерархии. Класс σ_{n+1} состоит из проекций множеств класса π_n. Класс π_{n+1} состоит из обобщений множеств класса σ_n. Каждому множеству графов из класса σ_n, размеченному алфавитом A_1, соответствует формула

$$\exists \mathcal{A}_{n+1} \, \forall \mathcal{A}_n \, \exists \mathcal{A}_{n-1} \ldots (\exists/\forall) \mathcal{A}_2 \mathcal{M}(\mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n).$$

Здесь $\mathcal{A}_1, \dots \mathcal{A}_n$ — разметки в алфавитах A_1,A_2,...A_n, а $\mathcal{M}$ — машина, работающая на графах, размеченных алфавитом $A_1 \times A_2 \times \ldots \times A_n$. Аналогично, каждому множеству класса π_n соответствует формула

$$\forall \mathcal{A}_{n+1} \, \exists \mathcal{A}_n \, \forall \mathcal{A}_{n-1} \dots (\exists/\forall) \mathcal{A}_2 \mathcal{M}(\mathcal{A}_1 \times \ldots \times \mathcal{A}_n).$$

ТЕОРЕМА 4. *Пусть M_1 и M_2 — два множества из одного любого класса иерархии, алфавиты разметки которых совпадают. Тогда множества $M_1 \bigcup M_2$ и $M_1 \bigcap M_2$ принадлежат тому же классу. Проекция любого множества из σ_n лежит в σ_n. Обобщение любого множества из π_n лежит в π_n.*

ДОКАЗАТЕЛЬСТВО. Пусть множества M_1 и M_2 задаются формулами Φ_1 и Φ_2. Опишем формулу Φ, задающую $M_1 \bigcup M_2$ или $M_1 \bigcap M_2$. Машина $\mathcal{M}$, стоящая в ее бескванторной части работает на графе, размеченном алфавитом, представляющем собой прямое произведение, первая компонента которого — алфавит разметки множеств M_1 и M_2, а другие компоненты — прямые произведения двух алфавитов, стоящих на одинаковых местах в Φ_1 и Φ_2. $\mathcal{M}$, также как в доказательстве теоремы 3 последовательно запускает машины из Φ_1 и Φ_2, каждую на своей "половине" разметки. Алфавиты, соответствующие кванторам в кванторной приставке формулы Φ, являются прямыми произведениями алфавитов, соответствующих тем же кванторам в Φ_1 и Φ_2. Вторая часть теоремы очевидна, так как два подряд идущих одинаковых квантора можно слить в один, беря прямое произведение соответствующих алфавитов. Теорема 4 доказана.

ТЕОРЕМА 5. *Семейство графов, обладающих гамильтоновым циклом и размеченных в конечном алфавите, принадлежит σ_1.*

ДОКАЗАТЕЛЬСТВО. У графа с выделенным гамильтоновым циклом и направлением обхода по нему каждую вершину разметим номером ребра, по которому из нее выходит цикл. Машина, пользуясь этой разметкой, проходит по предполагаемому циклу, отмечая посещенные вершины, пока не придет в начальную. Затем она обходит весь граф и проверяет, что все вершины отмечены. Теорема 5 доказана.

А.О.Слисенко доказал следующую теорему.

ТЕОРЕМА 6. *Существует алгоритм, который по размеченному связному графу G и грамматике выясняет, выводим ли G в этой грамматике, и если выводим, строит его вывод. При фиксированной грамматике Γ время работы этого алгоритма полиномиально зависит от размера G.*

ДОКАЗАТЕЛЬСТВО. В процессе вывода любая нетерминальная вершина промежуточного графа соответствует подграфу в окончательном графе, выведенному из нее. Число граничных ребер этого подграфа ограничено ветвлением грамматики. Пусть это число $\leq m$. Опишем требуемый алгоритм. Перебираем в G множества из $\leq m$ предполагаемых граничными ребер с указанием, какой конец каждого ребра внутренний и какой внешний. Для каждого такого множества M рассмотрим множество G_1 вершин G, соединимых с внутренней вершиной из M путем, не проходящим через внешние вершины, и множество G_2 вершин, соединимых с внешней вершиной путем, не проходящим через внутренние. В силу связности G, $G_1 \bigcup G_2 = G$. Если $G_1 \bigcap G_2 = \emptyset$, то у подграфа G_1 множество граничных ребер есть M и подграф с таким свойством единственен. Если $G_1 \bigcap G_2 \neq \emptyset$ то подграфа с этим свойством не существует. Таким образом выделим множество P всех подграфов с числом граничных ребер $\leq m$. Будем составлять таблицу, показывающую какие подграфы из P с висячими граничными ребрами выводимы из каких нетерминалов грамматики Γ. Будем указывать в таблице также соответствие между номерами ребер при нетерминале и висячими ребрами выводимого из этого нетерминала графа. Заполнение таблицы начинаем с наименьших подграфов, выводящихся за один шаг. Пусть мы указали в таблице все подграфы, которые могут быть выведены за $\leq n$ шагов. Тогда для каждого $p \in P$ и каждого правила $\eta \longrightarrow B$ из Γ естественным образом проверяется, имеется ли вывод p из η глубины $\leq n+1$ с применением правила $\eta \longrightarrow B$ на первом шаге. Таким образом добавим в таблицу все подграфы, выводимые за $n+1$ шагов. Так мы за полиномиальное от размера G время заполним всю таблицу и узнаем выводим ли G из начального нетерминала. Теорема 6 доказана.

Существенность условия связности показывает следующая теорема.

ТЕОРЕМА 7. *Для несвязных графов проблема выводимости размеченного графа в некоторой фиксированной грамматике NP-полна от размера графа.*

ДОКАЗАТЕЛЬСТВО. Пусть есть ленточная недетерминированная машина Тьюринга $\mathcal{M}$, распознающая NP-полный предикат и работающая время $p(|x|)$, где p — полином, x — входное слово. Можно, очевидно, считать, что для любого допускаемого слова x существует допускающий протокол длины ровно $p(|x|)$. Этот протокол будем представлять в виде матрицы B размера $p(|x|) \times p(|x|)$, на i-той строке которой записана конфигурация $\mathcal{M}$ в i-ый момент времени. Пусть L — алфавит записи конфигураций. Опишем некоторую процедуру заполнения матрицы B. Пусть для каждой тройки (i, j, b), где $1 < i \leq p(|x|)$, $j \leq p(|x|)$, $b \in L$, существует лампочка, которую можно зажигать. Для каждой тройки $(1, j, b_j)$, где $j \leq p(|x|)$, b_j — j-ый элемент начальной конфигурации, тоже есть лампочка. Будем двигаться по строкам матрицы B, заполняя ее элементами b_{ij} в порядке лексикографического возрастания пары (i, j).

На каждом шаге мы сначала произвольно угадываем значение элемента b_{ij}. Пусть b — угаданное значение. Зажигаем лампочку (i, j, b). Если $i < p(|x|)$ и $j > 1$, угадываем значение b_1 элемента $b_{i+1,j-1}$ из тех, которые возможны при условии, что $\mathcal{M}$ совершила один шаг работы в соответствии с программой, с учетом четырех элементов, угаданных на двух предыдущих шагах. После этого зажигаем все $S-1$ лампочек $(i+1, j-1, b_2)$, где $S = |L|$, $b_2 \neq b_1$. Если $j = p(|x|)$, то после этого угадываем и возможное значение b_3 элемента $b_{i+1,p(|x|)}$ и зажигаем $S-1$ лампочек $(i+1, p(|x|), b_4)$, $b_4 \neq b_3$. Покажем, что $\mathcal{M}$ допускает x тогда и только тогда, когда наша процедура может отработать так, чтобы зажечь каждую лампочку ровно один раз, и в последней конфигурации $\mathcal{M}$ находится в заключительном состоянии. Если допускающий протокол существует, то наша процедура угадывает его и при этом для каждой пары (i, j) зажигает лампочку (i, j, b), где b — значение b_{ij}, проходя по i-ой строке, а лампочки (i, j, c), $c \neq b$, проходя по предыдущей. Обратно, пусть $\mathcal{M}$ зажигает каждую лампочку ровно один раз. Тогда, проходя по i-ой строке, процедура должна угадать именно те значения b_{ij}, которые были угаданы при проходе по $(i-1)$-ой строке. Значит каждая последующая конфигурация получается из предыдущей совершением одного шага работы $\mathcal{M}$. Промоделируем описанную процедуру выводом специального графа G в некоторой грамматике Γ. Опишем граф G. Он состоит из одной основной компоненты K и многих вспомогательных. Все компоненты представляют собой размеченные цепи. Каждая буква из L является терминалом. Занумеруем пары (i, j) в лексикографическом порядке. K отражает весь ход нашей процедуры. Она состоит из $p^2(|x|)$ крупных подотрезков, которые взаимнооднозначно соответствуют парам (i, j) и располагаются по возрастанию их номеров. Номер (i, j) обозначим через $\mathrm{NOM}(i, j)$. Каждый крупный подотрезок состоит из мелких. Первый мелкий подотрезок представляет собой цепь из $\mathrm{NOM}(i, j)$ вершин, помеченных терминалом t, после которых идет вершина, помеченная терминалом r_1. Если $j \neq 1$ и $i \neq p(|x|)$, то далее следуют еще $S-1$ мелких подотрезков. Они состоят из $\mathrm{NOM}(i+1, j-1)$ терминалов t, за которыми следует терминал r_2. Если $j = p(|x|)$, то после этого блока подотрезков следует еще $S-1$ мелких подотрезков, состоящих из $\mathrm{NOM}(i+1, p(|x|))$ терминалов t и терминала r_3. В конце K стоит стоит терминал r_0. Остальные компоненты взаимнооднозначно соответствуют лампочкам. Компонента, соответствующая лампочке (i, j, b), это цепь, начинающаяся с терминала b, за которым идут $\mathrm{NOM}(i, j)$ терминалов d, а после них стоит терминал d_0. Главная идея в грамматике Γ состоит в том, чтобы в Γ можно было выводить основную компоненту и одновременно "отслаивать" от нее лампочки. Выведение лампочки соответствует ее зажиганию. В процессе вывода K выведенная часть ее указывает, какой шаг процедуры моделируется. Если выводится крупный подотрезок, соответствуюший (i,j), то это означает, что процедура совершает действия, находясь в точке b_{ij}. Вывод первого мелкого подотрезка соответствует зажиганию лампочки в (i,j), а вывод последующих $S-1$ мелких подотрезков — зажиганию лампочек в $(i+1,j-1)$. Опишем Г подробнее. Правила ее делятся на три вида: правила начала отщепления лампочки, продолжения и конца. Правила начала имеют вид: $\xi \longrightarrow C_1$, где C_1 — двухвершинный граф, представляющий собой нетерминал η, смежный с вершиной, помеченной терминалом $b \in L$. Из η выходит одно висячее ребро, которое в выводе становится ребром основной компоненты, а b это пометка первой вершины лампочки, зависящая от ξ. η зависит от ξ и от b. Правила продолжения имеют вид: $\eta \longrightarrow C_2$, где C_2 — трехвершинный граф. Он представляет собой нетерминал η, смежный с двумя терминальными вершинами, помеченными t и d. Из них выходит по одному висячему ребру, которые в выводе становятся ребрами основной компоненты и лампочки соответственно. Правила конца вывода лампочки имеют вид: $\eta \longrightarrow C_3$, где C_3 — трехвершинный граф. Он состоит из двух связных компонент. Первая компонента это терминальная вершина d_0, из которой исходит висячее ребро. В процессе вывода d_0 — последняя вершина отделившейся лампочки. Вторая компонента это нетерминал ξ, соединенный с терминальной вершиной r_i, где $i \in \{1,2,3\}$. Из r_i выходит висячее ребро, которое в выводе присоединяется к основной компоненте. ξ и i здесь зависят от η.

Опишем, какую информацию должен “помнить” текущий нетерминал η. η “помнит” элементы, угаданные на текущем шаге и двух предыдущих шагах по текущей строке, в частности отсутствие этих элементов, если процедура находится в начале строки, или отсутствие следующей строки, если текущая строка последняя. В η содержится также информация, является ли текущий элемент последним элементом строки. η “помнит”, зажигает ли процедура лампочку в текущей строке или в последующей. Например, если в текущей, то элемент b в правиле начала вывода лампочки может быть любым (угадывание произвольно), а если в последующей, то b зависит от элементов, угаданных на данном шаге и двух предыдущих. В этом случае η “помнит” какие из $S-1$ лампочек зажигались, а какие еще предстоит зажечь. Если текущая строка последняя и было угадано состояние машины, η “помнит” заключительное ли оно. Символ r_0 может появиться только если состояние заключительное. Соответствие между ξ и η в правилах начала и конца обеспечивает восстановление всей информации, за исключением той догадки, совершен ли переход на последний элемент строки или на начало последней строки. Пусть существует допускающий способ работы процедуры. Тогда существует и вывод G в Γ, так как единственность зажигания каждой лампочки позволит вывести соответствующую ей компоненту ровно один раз. Пусть, наоборот, существует вывод G в Γ. Из устройства Γ следует, что появление символа r_i в K может произойти только одновременно с окончанием вывода лампочки, и длина ее равна длине соответствующего отрезка в K. Если бы в процессе вывода неправильно был бы угадан конец строки, то символ r_3 появился бы не на том месте, что в основной компоненте. Из всего этого следует, что если G выведен, то порядок вывода лампочек соответствует порядку их зажигания в процедуре. То, что каждая лампочка была выведена ровно один раз, означает, что процедура допускающая. Теорема 7 доказана.

NP-полнота сохранится и в случае, если потребовать, чтобы граф был неразмеченным, а Γ представляла бы собой просто набор графов с висячими ребрами и начальный граф. При этом любой граф можно подставить вместо любой вершины, у которой число инцидентных ребер равно числу висячих ребер графа, соответствие между ребрами любое. Дадим набросок доказательства NP-полноты в этом случае. Смоделируем разметку вершин графа с помощью отростков длины 1. Каждому нетерминалу и терминалу поставим в соответствие свое натуральное число, причем любому терминалу большее, чем любому нетерминалу. Модифицируем граф G, добавив к каждой вершине столько отростков, чтобы число инцидентных ей ребер было равно числу, соответствующему ее пометке. В Γ оставим лишь графы правой части, модифицированные следующим образом. В нетерминальные и те терминальные вершины, из которых не выходит висячих ребер, добавим столько отростков, чтобы общее количество инцидентных им ребер соответствовало их пометкам. В те терминальные вершины, из которых выходят висячие ребра, добавим частично отростки и частично висячие ребра, так чтобы общее количество висячих ребер было равно числу, соответствующему нетерминалу в левой части и чтобы число инцидентных каждой вершине ребер соответствовало ее пометке.

В процессе вывода заменить на граф можно только одну вершину. Очевидно, что к выводу G может привести лишь такая замена, при которой два ребра, которые соответствуют направлению на лампочку и основную компоненту, отождествляются с соответствующими висячими ребрами графа. Дальнейшее доказательство следует доказательству теоремы 7.

Скажем, что формула Φ из иерархии истинна на графе G, если G принадлежит задаваемому Φ множеству. Следующая теорема обобщает теорему А.О.Слисенко о гамильтоновом цикле, упомянутую во введении.

ТЕОРЕМА 8. *Существует алгоритм, который по графу G и формуле Φ из описанной выше иерархии выясняет истинна ли Φ на G. При фиксированной формуле для графов, выводимых в фиксированной грамматике, время алгоритма полиномиально зависит от размера G.*

ДОКАЗАТЕЛЬСТВО. Сначала заметим, что для любой грамматики Γ существует эквивалентная ей грамматика Γ_1 такая, что в правой части любого правила Γ_1 стоит граф из ≤ 2 вершин. Чтобы получить Γ_1, каждое правило Γ расщепим на несколько последовательных правил, отщепляя от графа в правой части по одной вершине, а оставшуюся часть каждый раз обозначая новым нетерминалом. Очевидно, что размер Γ_1 полиномиален от размера Γ. Легко видеть, что для любого n и любого конечного алфавита A существует грамматика $\Gamma(A, n)$, в которой выводимы все графы, размеченные в A, выводимые в какой-либо грамматике с ≤ 2 вершинами в правой части каждого правила и $\leq n$ ребрами при каждой из этих вершин. Эта грамматика получается объединением всех таких правил с терминалами из A и одним, кроме начального, нетерминалом.

ЛЕММА. *Граф G выводим в $\Gamma(A, n)$ тогда и только тогда, когда в ней выводима любая его связная компонента.*

ДОКАЗАТЕЛЬСТВО. Пусть любая компонента выводима, а всего компонент K. Тогда сначала выведем граф из K изолированных нетерминальных вершин, отщепляя их по одной. Затем из этих вершин выведем по компоненте. Пусть, наоборот, G выводим и мы хотим доказать выводимость компоненты K. Возьмем вывод G. Начиная с начала вывода, преобразуем его следующим образом. Пусть на очередном шаге применяется правило $\eta \longrightarrow B$. Каждую терминальную вершину из B, которая в конце не принадлежит K и каждую нетерминальную вершину, из которой в конце выведется подграф, не пересекающийся с K, удалим из B. Удалим из B также все ребра, включая висячие, которые в окончательном графе не принадлежат K. Поскольку по предположению индукции все ребра текущего графа принадлежат K, новое правило $\eta \longrightarrow B'$ применимо. Применим его. Очевидно, что продолжая так ограничивать первоначальный вывод на K, получим вывод K в $\Gamma(A, n)$. Лемма доказана.

Из доказанной леммы следует, что перебирая грамматики $\Gamma(A, n)$ по возрастанию n мы найдем, используя алгоритм из теоремы 6, грамматику Γ, в которой выводим G, и построим его вывод. Этот вывод мы будем представлять в виде бинарного дерева D с размеченными вершинами. Каждая вершина дерева D соответствует вершине промежуточного графа в выводе, узел — нетерминальной, а лист — терминальной. Каждый узел помечен примененным в нем правилом вывода, каждой вершине правой части этого правила соответствует вершина дерева, смежная с данным узлом в направлении от корня. В пометке узла также ставится символ u. Лист помечен той же пометкой, что и соответствующая ему вершина графа G, указывается также степень этой вершины. Формуле Φ сопоставим детерминированный автомат A, работающий на деревьях. A будет допускать дерево вывода тогда и только тогда, когда на выведенном графе истинна Φ. Пусть $\mathcal{M}$ - машина, стоящая в бескванторной части Φ, а P — подграф графа G, имеющий не более n граничных ребер (n — параметр Γ). Рассмотрим подграф P изолированно от остальной части G вместе с граничными ребрами, которые перенумерованы. Пусть есть следующий способ работы $\mathcal{M}$ на P. $\mathcal{M}$, работая на P, может выходить из P и входить в P. Входить в P $\mathcal{M}$ может по произвольному ребру в произвольном состоянии. Войдя в подграф P, разметка которого определяется всей предыдущей работой $\mathcal{M}$, она работает в соответствии со своей программой до очередного выхода или до остановки в заключительном состоянии. Выходы и входы чередуются. Если начальная вершина находится в P, $\mathcal{M}$ начинает работу в ней, иначе входит в P. Посещаемость каждой вершины P ограничена, поэтому число входов и выходов тоже ограничено. Если $\mathcal{M}$ заканчивает работу в P, то она должна остановиться в заключительном состоянии, а если $\mathcal{M}$ последний раз вышла из P (в любом состоянии), то может больше не входить, в частности возможна пустая работа $\mathcal{M}$, когда она в P вообще не входила. Каждому такому способу работы соответствует последовательность S номеров ребер, по которым $\mathcal{M}$ входила в P и выходила из него, с приписанной каждому номеру информацией: вход или выход совершает $\mathcal{M}$; состояние, в которое переходит $\mathcal{M}$. Длина S не превышает $2nK$, где K — максимальное число посещений одной вершины. S назовем реализуемым следом машины $\mathcal{M}$ на P, а последовательности вида S и длины $\leq 2nK$ — следами. Таким образом подграфу P соответствует множество реализуемых следов, не зависящее от его внешности. Опишем детерминированный автомат A_1, который допускает дерево вывода тогда и только тогда, когда $\mathcal{M}$ допускает выведенный граф. Неначальными состояниями автомата A_1 будут множества следов. A_1 будет двигаться от листьев к корню дерева. В листьях он находится в одном и том же начальном состоянии. В каждом узле состоянием A_1 будет множество реализуемых следов машины $\mathcal{M}$ для подграфа, выведенного из данного узла. Состояние A_1 в очередной вершине p определяется по состояниям в двух смежных с ней в направлении от корня вершинах p_1 и p_2 и разметке вершин p, p_1, p_2. Покажем как это делается. Для всех одновершинных подграфов, соответствующих листьям, множество реализуемых следов определяется непосредственно по пометке листа. Пусть

вершинам p,p_1,p_2 соответствуют подграфы P,P_1,P_2. Построим множество реализуемых следов для P по множествам реализуемых следов для P_1 и P_2, а также информации о том, какие граничные к P_i ребра ведут в P_j, а какие в $G \setminus P$. Эту информацию дает правило вывода, которым помечена p. Назовем подробным следом для P такую последовательность S', в которой указаны все переходы между P_i и $G \setminus P$ и между P_1 и P_2 с информацией: направление перехода, номера ребра относительно P_i и P или P_1 и P_2, состояние после перехода. S' реализуем, если существует способ работы $\mathcal{M}$ на P, такой, что последовательность всех переходов $\mathcal{M}$ между P_1, P_2, $G \setminus P$ соответствует S'. Назовем проекцией S' на P_i след для P_i, получающийся вычленением из S' переходов между P_i и $G \setminus P_i$. Очевидно, что длина S' не превышает удвоенной длины обычного следа и что обычный реализуемый след для P является проекцией на P некоторого реализуемого подробного следа. Поэтому достаточно узнать, какие подробные следы реализуемы. Но очевидно:
(S' реализуем) $\Longleftrightarrow$ (проекции S' на P_1 и P_2 реализуемы).

A_1 допускает D, если в корне его состояние — множество из одного пустого следа. В [3] доказано, что проекция и дополнение множества деревьев, распознаваемого детерминированным автоматом, тоже распознаются детерминированными автоматами. Операция обобщения выражается через проекцию и дополнение: это дополнение до проекции дополнения. Операция проекции на графах означает проекцию на деревьях их вывода по разметке в листьях, но это несущественно, так как в узлах, помеченных символом u, автомат может не обращать внимание на "лишние" компоненты. Таким образом, производя согласно формуле Φ операции проекции и обобщения, построим искомый автомат A. Запустив его на D, узнаем истинна ли Φ. Теорема 8 доказана.

2. Связь с решетками

В этом разделе мы попытаемся оценить ту же проблему с другой стороны. С.Ф.Сопрунов доказал в [4], что для любой грамматики существует n такое, что граф, представляющий собой плоскую квадратную решетку размера $n \times n$, невыводим в этой грамматике. Это следует также из совокупности результатов теоремы 8 и следующей теоремы.

ТЕОРЕМА 9. *Проблема истинности некоторой фиксированной σ_1 формулы на размеченной квадратной решетке NP-полна от размера решетки.*

ДОКАЗАТЕЛЬСТВО. Выше мы рассматривали детерминированную машину, работающую на графах, но можно рассматривать и недетерминированную, программа которой не является однозначной функцией. Заметим, что класс σ_1 это множества, распознаваемые такой недетерминированной машиной с ограниченной посещаемостью вершины. Возьмем обычную ленточную недетерминированную машину Тьюринга $\mathcal{M}$, работающую полиномиальное время и распознающую NP-полный предикат. Промоделируем работу $\mathcal{M}$ работой недетерминированной машины $\mathcal{M}_1$ на данной решетке. При этом на каждой горизонтали в разметке будет записана конфигурация $\mathcal{M}$ в один момент времени. Сначала на нижней горизонтали размечена конфигурация $\mathcal{M}$ в начальный момент времени. Каждый шаг $\mathcal{M}$ моделируется следующим образом. $\mathcal{M}_1$ переходит вверх на следующую горизонталь и, проходя с одного края до другого, размечает ее вершины. Из каждой вершины $\mathcal{M}_1$ идет вниз, определяет по нижней вершине и двум ее соседям какая буква должна стоять в данной вершине. Очевидно, что $\mathcal{M}_1$ проходит по каждому ребру не более четырех раз. Теорема 9 доказана.

Некоторые основания считать семейство решеток простейшим семейством, для которого не существует полиномиального алгоритма, выясняющего истинность формулы из иерархии, дает теорема, доказанная Н.Робертсоном и П.Д.Сеймуром в [5]. Из нее следует, что для любого n существует грамматика такая, что если граф не имеет миноров (см. [5]), изоморфных решетке размера $n \times n$, то он выводим в этой грамматике.

3. Графовые машины: случай неограниченного ветвления

Очевидно, что графы, выводимые в одной грамматике, имеют ограниченную степень ветвления. В [7] К. Ю. Горбунов предложил понятие делимости, обобщающее выводимость на графы неограниченной степени ветвления. Будем рассматривать процессы деления конечного графа на подграфы: каждый не одновершинный подграф, возникший в процессе деления на очередном шаге делится на два подграфа, пока все подграфы не станут одновершинными. Скажем, что подграф B графа G отделим от своего дополнения $\leq n$ вершинами, если можно так выделить $\leq n$ вершин графа G, что любое граничное к B ребро инцидентно хотя бы одной из выделенных вершин. Вершины из B будем называть внутренними, из $G \setminus B$ — внешними.

ОПРЕДЕЛЕНИЕ. Граф называется n-делимым, если существует процесс его деления, в котором каждый возникающий подграф отделим от своего дополнения $\leq n$ вершинами.

Под n-делимостью подграфа будем понимать его n-делимость как графа с учетом граничных ребер. Теорема 6, доказанная А.О. Слисенко, обобщается, как показал К.Ю. Горбунов, на понятие делимости следующей теоремой.

ТЕОРЕМА 10. *Существует алгоритм, который по графу G и числу n выясняет, является ли G n-делимым, и если является, то строит процесс его деления. При фиксированном n время работы этого алгоритма полиномиально зависит от размера G.*

ДОКАЗАТЕЛЬСТВО. Пусть B — подграф графа G, отделенный от своего дополнения $\leq n$ отмеченными вершинами. Назовем вершину $g \in G$ *богатой для* B, если ей инцидентно $> n$ граничных к B ребер (кратные ребра считаются одним ребром). Деление B на две части, одна из которых одновершинная, назовем отделением соответствующей вершины.

ЛЕММА. *После отделения любой отмеченной внутренней вершины g подграф $B \setminus \{g\}$ будет отделим от дополнения $\leq n$ вершинами. Если b — богатая внутренняя вершина и B n-делим, то $B \setminus \{b\}$ n-делим за не большее, чем для B, число шагов деления. Если B несвязный подграф, то (B n-делим)$\Longleftrightarrow$(любая его связная компонента n- делима).*

ДОКАЗАТЕЛЬСТВО. Первое предложение леммы очевидно. Докажем второе. Процесс деления B индуцирует некоторый процесс деления подграфа $B \setminus \{b\}$. Пусть C — какой-либо подграф, возникший на промежуточном этапе второго процесса, а C' — соответствующий ему подграф процесса деления B. Достаточно показать, что если C' отделим от дополнения $\leq n$ вершинами, то и C тоже. Если $C' = C$ то все доказано. Пусть $C' = C + \{b\}$. Так как b — богатая внутренняя вершина для B, то она богатая и для C'. Поскольку отмеченных для C' вершин $\leq n$, то все концы граничных к C' ребер, инцидентных b не могут быть отмечены для C'. Поэтому b отмечена для C'. Но C получается из C' отделением b и по первому предложению леммы отделим от своего дополнения $\leq n$ вершинами. Последнее предложение леммы следует из того, что процесс деления B индуцирует процесс деления на любой его компоненте. Лемма доказана.

Пусть подграф B n-делим и отмечены $\leq n$ вершин, отделяющих его от дополнения. Покажем, что можно так отделить от B часть внутренних отмеченных вершин, чтобы после каждого отделения оставшаяся часть была бы n-делима, а в конце она не имела бы богатых внутренних вершин. Отделяем от B одну из его богатых, а значит и отмеченных внутренних вершин. Оставшийся подграф B_1 по лемме n-делим. Богатая для B_1 внутреняя вершина b_1 могла не быть богатой для B. Но каждое инцидентное ей граничное к B_1 ребро либо граничное и к B, либо ведет в отмеченную для B отделенную внутреннюю вершину. Поэтому b_1 отмечена для B, иначе бы больше n смежных с ней вершин были бы отмечены для B, что невозможно. Отделяя b_1 от B_1 получим B_2 и так далее, пока у B_i не останется богатых внутренних вершин, что произойдет за $\leq n$ шагов. Таким образом, если G n-делим, то существует такой процесс его деления, при котором для любого возникающего в этом процессе подграфа B сначала отделяются $\leq n$ вершин, а затем оставшийся подграф B', не имеющий богатых внутренних вершин, делится на связные компоненты. Очевидно, что для B' число внешних граничных неотмеченных вершин не более n^2. Будем называть такие вершины присоединенными к B'. Будем рассматривать совокупности: множество M из $\leq n$ вершин графа G с указанием какие из них внешние, а какие внутренние; множество P из $\leq n^2$ присоединенных внешних вершин; связную компоненту подграфа $G \setminus P$, содержащую все внутренние вершины и отделенную от своего дополнения множеством M. Будем по шагам составлять список всех таких совокупностей, в которых связная компонента n-делима. После m-го шага в нашем списке будут все совокупности с компонентой, n-делимой не более чем за m делений и, возможно, некоторые с компонентой, n-делимой более чем за m делений. На $(m+1)$-ом шаге мы перебираем все совокупности и для каждой совокупности S, не входящей в наш список, делаем следующее. Угадываем два множества из $\leq n$ вершин M_1 и M_2 с указанием, какие вершины в каждом из них внутренние и какие внешние. Эти множества есть предполагаемые отделяющие множества для двух частей K_1 и K_2, на которые мы хотим разбить компоненту K из S и которые предполагаем n-делимыми за $\leq m$ делений. Затем для каждого M_i угадываем множества $O_1 \subseteq M_1$ и $O_2 \subseteq M_2$ тех внутренних вершин, которые можно отделить и перевести во внешние, чтобы подграфы $K_1 \setminus O_1$ и $K_2 \setminus O_2$ остались бы n-делимыми и не имели бы богатых внутренних вершин. Угадываем множества P_1 и P_2 присоединенных для этих подграфов вершин. Пусть V_i $(i = 1, 2)$ есть объединенине множества внешних вершин из M_i и множества $O_i \bigcup P_i$. Тогда, если все наши угадывания верны, подграфы $K_1 \setminus O_1$ и $K_2 \setminus O_2$ должны быть объединением некоторых связных компонент подграфов $G \setminus V_1$ и $G \setminus V_2$ соответственно. Назовем компоненты $G \setminus V_1$ первыми, а $G \setminus V_2$ вторыми. Попытаемся разбить подграф $K \setminus (O_1 \bigcup O_2)$ на две части так, чтобы одна часть была объединением некоторых из первых компонент, а вторая — вторых. Рассматриваем лишь те компоненты, которые лежат внутри $K \setminus (O_1 \bigcup O_2)$. Понятно, что к K_i должны быть приписаны i-ые компоненты, удовлетворяющие любому из следующих трех условий.

1. Содержащие внутреннюю вершину из M_i.
2. Содержащие вершину из V_j при $j \neq i$.

3. Имеющие граничное ребро, не инцидентное ни одной вершине из M_j при $j \neq i$.

Легко видеть по условию 2, что любая оставшаяся неприписанной i-ая компонента либо строго вложена в некоторую j-ую компоненту ($j \neq i$), либо совпадает с ней. В первом случае i-ая компонента никуда не приписывается, так как j-ая уже приписана по условию 2. Во втором случае, если j-ая компонента уже приписана, то i-ая не приписывается, а если не приписана, то по условиям 1 и 3 "спорная" компонента не содержит внутренних вершин из M_1 и M_2, а любой внешний конец граничного к ней ребра принадлежит и M_1 и M_2. Приписываем ее в произвольную часть. После этого проверяем, что любая вершина из $K \setminus (O_1 \bigcup O_2)$ принадлежит ровно одной части. Легко видеть, что истинность этого условия не зависит от того, куда мы приписали "спорные" компоненты. То, что множества M_1 и M_2 являются отделяющими для построенных K_1 и K_2, следует из проверенного условия и условия 3. Если K n-делим за $m+1$ шагов и все наши угадывания были верными, то все выбранные компоненты должны уже быть в нашем списке. Пусть они там есть. Покажем, что тогда K n-делим, возможно за $> m+1$ шагов. Опишем процесс деления K. Делим K на K_1 и K_2. Отделяем от K_i по одной все вершины из O_i, при этом по лемме граница остается $\leq n$. Подграфы $K_1 \setminus O_1$ и $K_2 \setminus O_2$ делим на связные компоненты. Эти компоненты, как мы убедились ранее, есть в списке, значит они n-делимы. Их делением процесс кончается. Поскольку количество совокупностей и способов угадывания ограничено полиномом от размеров G при фиксированном n, то описанный алгоритм искомый. Теорема 10 доказана.

Рассмотренные ранее детерминированные машины на графах ограниченной степени ветвления могут быть обобщены недетерминированными машинами, работающими на графах неограниченной степени ветвления. Вершины графов, как и прежде, размечены буквами некоторого алфавита. Ребра тоже размечены буквами некоторого алфавита, а не перенумерованы, как было раньше. Разные ребра могут иметь одинаковые пометки. В графе выделена одна начальная вершина. Опишем недетерминированную машину, работающую на таких графах с числом посещений каждой вершины $\leq K$. Она задается набором $\langle A, B, Q, q_0, F\rangle$, компоненты A, Q, q_0 которого те же, что и в детерминированном случае. B это алфавит разметки ребер графа. Программа F это функция. Область ее определения — множество наборов $\langle q, a, k\rangle$, где $q \in Q$ — текущее состояние машины, а $a \in A$ — буква, которой помечена текущая вершина, $k \leq K+1$ — число посещений текущей вершины с учетом текущего. Если $k \leq K$, а q — незаключительное состояние, то значением F является множество, элементы которого — наборы $\langle q', a', (b, b')\rangle$.

Здесь q' — новое состояние машины, $a' \in A$ — новая пометка текущей вершины, $b \in B$ — пометка ребра, по которому машина сдвигается из текущей вершины, $b' \in B$ — новая пометка этого ребра. Если $k = K+1$, значением является безрезультатная остановка. Если q — заключительное состояние, $k \leq K$, то машина допускает граф. В начале работы машина находится в начальной вершине и в начальном состоянии. Потом она недетерминированно действует по своей программе, причем на каждом шаге недетерминированно выбирает ребро с данной пометкой, по которому сдвигается. Каждая машина определяет множество графов, на которых существует допускающий их способ работы. Назовем такие множества распознаваемыми.

ТЕОРЕМА 11. *Объединение и пересечение двух распознаваемых множеств являются распознаваемыми.*

ДОКАЗАТЕЛЬСТВО. Объединение, очевидно, допускается машиной, которая является формальным объединением машин, где два начальных состояния “склеены” в одно. Чтобы доказать утверждение для пересечения, докажем следующую лемму.

ЛЕММА. *Для любой машины $\mathcal{M}$ существует машина $\mathcal{M}'$, которая распознает то же множество графов, что и $\mathcal{M}$, и заканчивает работу в начальной вершине.*

ДОКАЗАТЕЛЬСТВО. Модифицируем $\mathcal{M}$ так, чтобы она имела возможность вернуться из текушей вершины в начальную. Это делается следующим образом. В процессе работы каждой вершине a, посещенной машиной, присваивается четность противоположная той, которую имеет вершина, из которой машина впервые пришла в a. Нечетные вершины помечаются специальным символом α_1, а четные — α_2. При каждом переходе из вершины a_1 в a_2 ребро l, по которому совершен переход, помечается кроме основного еще специальным символом β'. Если a_2 уже была посещена (в этом случае она помечена α_1 или α_2), то $\mathcal{M}'$ за два шага туда и обратно стирает символ β' с l (никакое другое ребро в этот момент символом β' не помечено). Иначе, $\mathcal{M}'$ за те же два шага заменяет β' на β_1, если a_1 четная или на β_2, если нечетная. Чтобы вернуться в начальную вершину, $\mathcal{M}'$ должна пройти по цепи из ребер с чередующимися пометками β_1 и β_2. Лемма доказана.

Машина, распознающая пересечение множеств распознаваемых машинами $\mathcal{M}_1$ и $\mathcal{M}_2$, строится так. Машина $\mathcal{M}_1$ переделывается в $\mathcal{M}'$, которая работает в соответствии с программой $\mathcal{M}_1$ и, если $\mathcal{M}_1$ допускает граф, возвращается в начальную вершину. Информацию о первоначальной разметке $\mathcal{M}'$ сохраняет. К $\mathcal{M}'$ последовательно подсоединяется $\mathcal{M}_2$. Теорема 11 доказана.

Определим классы σ_n и π_n для случая неограниченной степени ветвления и $n \geq 1$. Класс σ_1 это класс всех распознаваемых множеств. Класс π_1 это класс множеств, дополнение до которых является распознаваемым множеством. Класс σ_{n+1} состоит из проекций множеств класса π_n. Класс π_{n+1} состоит из обобщений множеств класса σ_n. Здесь операции проекции и обобщения производятся по разметкам и вершин и ребер. Теорема 4 справедлива и в недетерминированном случае. Отличие ее доказательства лишь в том, что в бескванторной части формулы фигурирует (возможно, под знаком дополнения) недетерминированная машина. Теорема 8 обобщается следующей теоремой.

ТЕОРЕМА 12. *Существует алгоритм, который по графу G и формуле Φ из недетерминированной иерархии выясняет истинна ли Φ на G. При фиксированной Φ и фиксированном n для n-делимых графов время работы этого алгоритма полиномиально зависит от размера G.*

ДОКАЗАТЕЛЬСТВО. Перебирая по возрастанию натуральные числа, мы найдем, используя алгоритм из теоремы 10, минимальное n такое, что G n-делим, и построим процесс его деления. Докажем следующую лемму, которую сообщил автору К.Ю. Горбунов.

ЛЕММА 1. *Если граф G n-делим и есть процесс его деления, то можно для каждого возникающего подграфа P так отметить $\leq n$ отделяющих вершин, чтобы при его делении на P_1 и P_2 выполнялись следующие условия:*

1. *Если вершина $a \notin P$ и a отмечена для P_i, то a отмечена для P.*
2. *Если $a \in P_i$ и a отмечена для P, то a отмечена для P_i.*
3. *Если $a \in P_i$ и a отмечена для P_j $(j \neq i)$, то a отмечена для P_i.*

ДОКАЗАТЕЛЬСТВО. Для каждого P пусть $n_1(P) \leq n$ — минимальное число такое, что существует система из n_1 вершин, отделяющая P от $G \setminus P$. Среди всех отделяющих систем мощности n_1 отметим систему с минимальным числом внешних вершин.

Докажем утверждение 1. Рассмотрим множество M вершин из $G \setminus P$, отмеченных для P_i, но не для P. Пусть вопреки утверждению $M \neq \emptyset$. Пусть C — множество вершин из P_i, соединенных ребром с вершиной из M и не отмеченных для P_i. Предположим, что $|C| > |M|$. Очевидно, все вершины из C отмечены для P, а любое граничное к P ребро инцидентное вершине из C ведет либо в M, либо в вершину, отмеченную для P. Поэтому, если заменить для P множество C отмеченных вершин на M, получится система, отделяющая P от $G \setminus P$ и меньшая по мощности. Это противоречит минимальности первоначальной системы. Предположим, что $|C| \leq |M|$. Любая вершина из P_i, смежная с вершиной из M либо лежит в C, либо отмечена для P_i. Поэтому, если заменить для P_i множество M отмеченных вершин на C, получится система, отделяющая P_i от $G \setminus P_i$. Она будет не больше первоначальной, а внешних к P_i вершин будет меньше. Противоречие.

Докажем утверждение 2. Рассмотрим множество M вершин из P_i, отмеченных для P, но не для P_i. Пусть $M \neq \emptyset$. Пусть C – - множество вершин из $G \setminus P$, соединенных ребром с M и не отмеченных для P. Предположим, что $|M| \leq |C|$. Очевидно, C отмечено для P_i. Заменим для P_i C на M. Получим систему отделяющих вершин для P_i с меньшим количеством внешних вершин, что противоречит минимальности системы. Предположим, что $|M| > |C|$. Заменим для P M на C. Получим систему отделяющих для P вершин, меньше предыдущей. Противоречие.

Доказательство утверждения 3 получается из доказательства утверждения 2 заменой $G \setminus P$ на P_j и P на P_j. Лемма 1 доказана.

Будем считать, что для всех рассматриваемых далее подграфов система отделяющих вершин берется такой, как при доказательстве леммы 1.

Процесс деления G будем представлять в виде бинарного дерева с размеченными вершинами. Каждой вершине p этого дерева соответствует подграф P, возникший в процессе деления. Если p узел, то двум его "сыновьям" p_1 и p_2 соответствуют два подграфа P_1 и P_2, на которые делится P. Пусть $V(p)$ — множество вершин G, отмеченных для хотя бы одного из подграфов P, P_1, P_2. Занумеруем $V(p)$. В разметке p указываются номера элементов множества $V(p)$ и для каждого номера говорится, для какого подграфа соответствующая вершина g отмечена, лежит ли она в P, P_1, P_2, принадлежит ли она $V(p_1)$, где p_1 — "отец" p и, если да, то какой номер она там имеет. Если $g \in V(p)$, то для каждой буквы $b \in B$ указываается $\min(2K, m)$, где K — максимальное число посещений одной вершины, а m — число граничных к P ребер, помеченных b и инцидентных g. Указывается также наличие всех ребер, оба конца которых являются отмеченными (возможно, для разных подграфов) и пометки этих ребер. Будем называть такие ребра отмеченными. В пометке узла также ставится символ u. В пометке листа указывается пометка соответствующей ему вершины и то, начальная ли она. Формуле Φ сопоставим детерминированный автомат A, работающий на деревьях. A будет допускать дерево деления тогда и только тогда, когда на первоначальном графе истинна Φ. Пусть $\mathcal{M}$ — недетерминированная машина, фигурирующая в бескванторной части Φ, а P — подграф графа G, отделенный от своего дополнения $\leq n$ отмеченными перенумерованными вершинами. Рассмотрим подграф P изолированно от остальной части G вместе с граничными ребрами и отмеченными вершинами. Пусть есть следующий способ работы $\mathcal{M}$ на P. $\mathcal{M}$, работая на P, может выходить из P и входить в P. Перед входом в P M может находиться в произвольном состоянии и входить по произвольному ребру, пометка которого соответствует шагу программы. Войдя в подграф P, у которого разметка вершин и ребер (в том числе граничных) определяется всей предыдущей работой $\mathcal{M}$, она недетерминированно работает в соответствии со своей программой до очередного выхода или до остановки в заключительном состоянии. Выходы и входы чередуются. Если начальная вершина находится в P, $\mathcal{M}$ начинает работу в ней, иначе входит в P. Посещаемость каждой вершины в P ограничена, поэтому число входов и выходов тоже ограничено. Если $\mathcal{M}$ заканчивает работу в P, то она должна остановиться в заключительном состоянии, а если $\mathcal{M}$ последний раз вышла из P (в любом состоянии), то может больше не входить, в частности возможна пустая работа $\mathcal{M}$, когда она в P вообще не входила. Каждому такому способу работы соответствует последовательность S номеров отмеченных вершин, которым инцидентны ребра переходов. Если ребро, по которому $\mathcal{M}$ вошла или вышла, инцидентно двум отмеченным вершинам, то вместо одного номера в S указывается пара номеров. Каждому номеру приписывается информация: вход или выход совершает $\mathcal{M}$; состояние, в которое переходит $\mathcal{M}$; пометки ребра, по которому переходит $\mathcal{M}$ до перехода и после перехода. Длина S ограничена $2nK$, где K — максимальная посещаемость вершины. S назовем реализуемым следом машины $\mathcal{M}$ на P, а последовательности вида S и длины $\leq 2nK$ — следами. Способ работы $\mathcal{M}$ будем называть реализацией следа S. Таким образом подграфу P соответствует множество реализуемых следов, зависящее только от P, его граничных ребер и отмеченных вершин. Опишем детерминированный автомат A_1, который допускает дерево деления тогда и только тогда, когда $\mathcal{M}$ допускает первоначальный граф. Неначальными состояниями A_1 будут множества следов. A_1 будет двигаться от листьев к корню дерева. В листьях он находится в одном и том же начальном состоянии. В каждом узле состоянием A_1 будет множество реализуемых следов $\mathcal{M}$ для подграфа, соответствующего этому узлу. Состояние A_1 в очередной вершине p определяется по состояниям в двух ее "сыновьях" p_1 и p_2 и разметке вершин p, p_1, p_2. Покажем как это делается. Для всех одновершинных подграфов, являющихся листьями, множество реализуемых следов определяется непосредственно по разметке. Пусть вершинам p, p_1, p_2 соответствуют подграфы P, P_1, P_2. Построим множество реализуемых следов для P по множествам реализуемых следов для P_1 и P_2 и разметке p. Назовем подробным следом для P такую последовательность S', в которой указаны все переходы между P_i и $G \setminus P$ и между P_1 и P_2 с информацией: направление перехода, номера отмеченных вершин для P_i и P или P_1 и P_2, которым инцидентно ребро перехода, пометки этого ребра до и после перехода, состояние $\mathcal{M}$ после перехода. S'

реализуем, если существует способ работы $\mathcal{M}$ на P такой, что последовательность всех переходов $\mathcal{M}$ между P_1, P_2, $G \setminus P$ соответствует S'. Очевидно, что длина S' не превышает удвоенной длины обычного следа и что обычный реализуемый след для P является проекцией на P некоторого реализуемого подробного следа. Поэтому достаточно узнать, какие подробные следы реализуемы. Покажем, что для существования реализации S' необходимо и достаточно, чтобы S' удовлетворял следующим трем условиям.

1. Проекции S' на P_1 и P_2 реализуемы.

2. Пометка каждого отмеченного ребра l перед очередным прохождением по нему совпадает с пометкой l после последнего прохождения по нему, а перед первым прохождением — с начальной пометкой.

3. Перед каждым переходом в S' между P и $G \setminus P$ по ребру, инцидентному отмеченной для P внутренней вершине g, существует > 0 граничных к P ребер, инцидентных a и имеющих пометку, указанную в данном переходе.

Необходимость данных условий очевидна. Докажем достаточность. Обозначим проекции S' на P_1 и P_2 через S_1 и S_2. Пусть мы построили начало реализации следа S' и проекции этого начала на P_1 и P_2 совпадают с началами некоторых реализаций следов S_1 и S_2. Пусть нам предстоит сделать очередной переход, которому в S' должен соответствовать очередной член s. Продолжим реализацию S' следующим образом. Возможны следующие два случая.

СЛУЧАЙ 1. Переход из P_i в P_j. Переберем возможные подслучаи.

а) Для P_i и P_j в s указаны внутренние отмеченные вершины. Тогда переход совершается по отмеченному ребру l, пометка которого совпадает согласно условию 2 с пометкой в s. Очевидно, что дальнейшая реализуемость S_i и S_j не зависит от того, по какому ребру с данной пометкой, инцидентному данной внутренней отмеченной вершине, машина $\mathcal{M}$ выйдет или войдет. Поэтому, совершив переход по l, $\mathcal{M}$ совершит выход из P_i и вход в P_j в соответствии с реализациями S_i и S_j. Затем $\mathcal{M}$ проработает в P_j и продолжит реализацию S_j до следующего перехода.

б) Для P_i отмечена только внутренняя вершина, для P_j — только внешняя. $\mathcal{M}$ входит в P_j, продолжая реализацию S_j до следующего перехода.

в) Для P_i отмечена только внешняя вершина, для P_j — только внутренняя. Машина входит в P_j, продолжая реализацию S_i, и работает в P_j в соответствии с реализацией S_j.

г) Для P_1 и P_2 в s указаны внешние отмеченные вершины. Тогда по утверждению 3 леммы 1 эти вершины отмечены и как внутренние. Данный случай свелся к случаю а).

СЛУЧАЙ 2. Переход между P_i и $G \setminus P$.

а) Для P_i отмечена внутренняя вершина, для P — внешняя. Совершаем переход по отмеченному ребру, имеющему в силу условия 2 нужную пометку. Если совершен вход в P_i, $\mathcal{M}$ продолжает работу в соответствии с реализацией S_i.

б) И для P_i и для P отмечена только внутренняя вершина. По условию 3 существует граничное к P ребро, инцидентное этой вершине и имеющее нужную пометку. По нему и совершаем переход.

в) И для P_i и для P отмечена только внешняя вершина. Переход совершается в соответствии с реализацией S_i.

г) Для P_i отмечена внешняя вершина, для P — внутренняя. Тогда по утверждению 1 леммы 1 внешняя отмеченная для P_i вершина отмечена и для P. Внутренняя для P вершина по утверждению 2 леммы 1 отмечена и для P_i. Данный случай свелся к случаю а).

Таким образом мы продолжим начало реализации S' на один член и снова проекции этого начала совпадают с началами реализаций S_1 и S_2. Значит, мы можем по шагам реализовать S'. Достаточность доказана.

Очевидно, что истинность приведенных трех условий проверяется по S'. Возможность найти множество реализуемых для P следов доказана. Автомат A_1 допускает дерево, если в корне его состояние — непустое множество следов.

Докажем возможность построения искомого автомата A. Выразим формулой в монадической теории дерева тот факт, что его разметка, несущая информацию о реберной разметке графа, корректна. Каждой вершине g из G поставим в соответствие множество $M(g)$ вершин D таких, что g отмечена или для подграфа, соответствующего данной вершине, или для хотя бы одного из двух его сыновей, на которые он делится. Таким образом $M(g) = \{p \mid g \in V(p)\}$.

ЛЕММА 2. *Для любой вершины g графа G множество $M(g)$ связно в D. Для любого ребра (g_1, g_2) из G $M(g_1) \bigcap M(g_2) \neq \emptyset$.*

ДОКАЗАТЕЛЬСТВО. Вершине g соответствует в D путь $l(g)$ от корня к листу, вдоль которого g — внутренняя вершина. В силу минимальности семейства отмеченных вершин, g отмечена в листе. Следовательно, в силу утверждения 2 леммы 1, вершины из $l(g)$, в которых g отмечена, составляют непустой отрезок от некоторой вершины до листа. Пусть $p \in M(g)$ и g — внешняя отмеченная вершина для подграфа p. Из утверждения 1 леммы 1 следует, что на всем пути от p до $l(g)$ g остается внешней отмеченной вершиной. Следовательно, весь этот путь, включая его конец $p_1 \in l(g)$, принадлежит $M(g)$. Из утверждения 3 леммы 1 следует, что для сына подграфа p_1, принадлежащего $l(g)$, g — внутренняя отмеченная вершина. Следовательно, этот сын лежит на вышеупомянутом отрезке и связность $M(g)$ доказана. Докажем второе утверждение леммы. Пусть p_1 — наиболее удаленная от корня вершина, лежащая и на $l(g_1)$ и на $l(g_2)$. Очевидно, для любого ее сына отмечена либо g_1 либо g_2. Если и g_1 и g_2 отмечены для одного из сыновей, то $p \in M(g_1) \bigcap M(g_2)$. Иначе, пусть например g_1 в сыновьях не отмечена. Пусть p_2 — наименее удаленная от корня вершина на $l(g_1)$, для которой отмечена g_1. Тогда для отца вершины p_2 отмечена g_2 и он, очевидно, принадлежит $M(g_1) \bigcap M(g_2)$. Лемма 2 доказана.

Каждой вершине $g \in G$ сопоставим следующую разметку R_g дерева D. В каждой вершине дерева указано целое неотрицательное число. Оно отлично от нуля в точности на $M(g)$ и равно в $p \in D$ номеру g в $V(p)$. Напомним, что основная разметка D задает для каждой пары смежных вершин p_1,p_2 из D соответствие между номерами вершин из $V(p_1)$ и $V(p_2)$.

ЛЕММА 3. (*Числовая разметка R дерева D с непустой связной ненулевой частью есть R_g для некоторой $g \in G$*)$\Longleftrightarrow$(*Для любых смежных p_1,p_2 из D, номера R в p_1,p_2 согласованы с основной разметкой D*).

ДОКАЗАТЕЛЬСТВО. Утверждение $\Longrightarrow$ очевидно. Докажем обратное. Возьмем вершину $p \in D$, где $R(p) \neq 0$. Номеру $R(p)$ соответствует некоторая $g \in G$. Так как $M(g)$ связно в D, то в силу согласованности R совпадает с R_g на $M(g)$, а в силу связности ненулевой части — и на $D \setminus M(g)$. Лемма 3 доказана.

Таким образом, вершины графа G можно моделировать разметками дерева D вышеописанного вида. Ребра G можно по лемме 2 моделировать парами (R_1, R_2) таких разметок, где R_1 и R_2 имеют непустое ненулевое пересечение. Связность ненулевой части R можно выразить так: для любой разметки R' такой, что для любой $p \in D$ если $R'(p) \neq 0$, то $R'(p) = R(p)$, выполняется:

$$(\forall(p_1, p_2)(R'(p_1) \neq 0 \ \& \ R(p_2) \neq 0 \Rightarrow R'(p_2) \neq 0)) \Longrightarrow (R' = R)$$

Здесь (p_1, p_2) — ребро из D. Понятие “вершина $p \in D$” моделируется разметкой D единицей в p и нулями в $D \setminus p$, а “ребро (p_1, p_2)” —— разметкой с единицами в p_1 и p_2. Корректность разметки отмеченных ребер на D заключается в том, чтобы одни и те же ребра имели одинаковые пометки. В силу связности $M(g_1) \bigcap M(g_2)$, где (g_1, g_2) — ребро, эта корректность выражается в согласованности с числовой разметкой D и проверяется автоматом очевидным образом. Корректная разметка в D отмеченных ребер однозначно определяет разметку ребер в G. Поэтому осталось выразить согласованность с ней той части разметки D, которая указывает число ребер с данной пометкой, инцидентных данной отмеченной вершине. Эта согласованность имеет место, если для любой

$p \in D$ и любой вершины g из $V(p)$ выполняется: $\mathop{\&}\limits_{i=1}^{N} \mathop{\&}\limits_{b \in B}$(число граничных ребер с пометкой b инцидентных g равно i)$\Longleftrightarrow$(в p при номере вершины g стоит i). Здесь $N = \min(2K, m)$, при $i = N$ в левой части эквивалентности подразумевается "не меньше i". Принадлежность g к $V(p)$ означает, что разметка R_g не нулевая в p, а существование i различных граничных ребер инцидентных g с пометкой b означает, что существует i различных вершинных разметок R с непустыми ненулевыми пересечениями $R \bigcap R_g$ и пометкой b в них таких, что (вершина с номером $R(p)$ указана в разметке D как внешняя к p)$\Longleftrightarrow$(g указана как внутренняя к p).

Опишем порядок построения автомата A. В кванторной приставке формулы Φ выражаем понятие "реберная разметка графа G"(это означает корректность разметки на D) вышеописанным способом. Получаем формулу Φ' на D. В силу того, что проекция и дополнение автоматного множества деревьев являются автоматными, и коньюнкция двух автоматных множеств автоматна, можно построить по автомату A_1 искомый автомат A, проверяющий истинность Φ' на D, что равносильно истинности Φ на G. Теорема 12 доказана.

Пусть мы ограничиваем не число посещений машиной каждой вершины, а лишь число прохождений по каждому ребру. Покажем, что в этом случае существует формула из σ_1, проблема истинности которой на 2-делимых графах NP-полна от размера графа. Возьмем обычную ленточную недетерминированную машину Тьюринга $\mathcal{M}$, задающую NP-полный предикат, время работы которой на отрезке ленты равно $p(|x|)$, где p — полином, x — входное слово. Этот отрезок будем представлять размеченной цепью. Соединим каждые две соседние вершины этой цепи $p(|x|)$ ребрами. В разметке вершин укажем через одну от края четные и нечетные вершины. В разметке ребер тоже укажем чередующиеся четные и нечетные пучки кратных ребер, а также каждое ребро пометим числом 0. Смоделируем $\mathcal{M}$ работой машины $\mathcal{M}'$ на полученном графе с проходом по каждому ребру не более одного раза. Каждый раз $\mathcal{M}'$ идет по ребру, помеченному 0, и меняет эту пометку на 1. Если текущая вершина четная, то, например, при команде "направо" $\mathcal{M}'$ идет по четному ребру, а если нечетная, то наоборот.

4. Гиперграфы

Напомним, что гиперграфом называется множество вершин и гиперребер, где гиперребро — произвольное подмножество множества вершин. Будем рассматривать гиперграфы с размеченными вершинами.

Подгиперграфом P гиперграфа G назовем подмножество вершин G и те гиперребра, которые целиком лежат в этом подмножестве. Гиперребра, которые частично в нем лежат, назовем граничными к P. Под размером гиперграфа будем понимать сумму числа его вершин и числа гиперребер. Процесс деления множества вершин гиперграфа определяется так же, как для графа.

ОПРЕДЕЛЕНИЕ. Гиперграф называется n-делимым, если существует такой процесс его деления, при котором каждый возникающий подгиперграф имеет не более n граничных гиперребер.

ТЕОРЕМА 13. *Существует алгоритм, который по гиперграфу G и числу n выясняет, является ли G n-делимым и, если является, строит процесс его деления. При фиксированном n время работы этого алгоритма полиномиально зависит от размера G.*

ДОКАЗАТЕЛЬСТВО. Назовем подгиперграф P гиперграфа G связным, если для любых двух его вершин $\mathbf{p}_1$ и $\mathbf{p}_2$ существует последовательность вершин $\mathbf{p}_1$, $\mathbf{x}_1$, $\mathbf{x}_2, \ldots \mathbf{x}_k$, $\mathbf{p}_2$ такая, что любые две соседние в ней вершины принадлежат одному гиперребру, лежащему целиком внутри P. Очевидно, если P несвязен, он однозначно разбивается на связные компоненты. Под n-делимостью P будем понимать n-делимость P как гиперграфа, но с учетом граничных гиперребер.

ЛЕММА. *(P n-делим)$\Longleftrightarrow$(любая его связная компонента n-делима).*

ДОКАЗАТЕЛЬСТВО. Пусть P n-делим, K — его связная компонента. Рассмотрим пересечение подгиперграфа P', возникшего в процессе деления P, с K. Так как K — компонента, не существует гиперребра граничного к K и целиком лежащего в P'. Поэтому, число граничных гиперребер у $P' \bigcap K$ не больше, чем у P'. Значит процесс деления P индуцирует процесс деления K. Обратно, если любая связная компонента P n-делима, то сначала делим P на компоненты, отщепляя их по одной. При этом число граничных гиперребер не увеличивается. Лемма доказана.

Очевидно, число связных подгиперграфов гиперграфа G с $\leq n$ граничными гиперребрами полиномиально от размера G. Будем составлять список тех из этих подгиперграфов, которые n-делимы. Пусть перед очередным шагом в списке есть все подгиперграфы, n-делимые за $\leq m$ делений. На очередном шаге для каждого кандидата K на то, чтобы быть помещенным в список, делаем следующее. Угадываем множество R, $|R| \leq n$ гиперребер, целиком лежащих в K и граничных к двум частям K_1 и K_2 на которые по предположению можно разделить K. Угадаем также разбиение граничных к K гиперребер на три множества: граничные только к K_1, только к K_2 и к обеим частям. Удалением множества R, K разбивается на связные компоненты, которые назовем мелкими компонентами. Каждая мелкая компонента K' должна, очевидно, быть компонентой одной из K_i. Поэтому, если K n-делим и все наши угадывания верны, то среди граничных к K' гиперребер не могут быть одновременно гиперребра граничные только к K_1 и только к K_2. Проверим это условие. Пусть оно выполняется. Связные компоненты подгиперграфов K_1 и K_2 должны быть среди мелких. Поэтому из леммы следует, что все мелкие компоненты должны находиться в нашем списке. Пусть и это условие выполняется. Покажем, что тогда K n-делим и, следовательно, его можно занести в список. Определим K_1 и K_2 следующим образом. Мелкие компоненты, все граничные гиперребра которых являются по предположению граничными к K_i, отнесем в K_i (те, для которых это условие выполнено и для i=1 и для i=2, отнесем в произвольную часть). Очевидно, что граничных гиперребер у K_i не более n. Разделим K на K_1 и K_2. Затем делим K_1 и K_2 на компоненты, отщепляя их по одной. Полученные мелкие компоненты делим дальше. Таким образом заполним весь список. Очевидно: (G n-делим)$\Longleftrightarrow$(любая его связная компонента n-делима). Теорема 13 доказана.

Подобно тому, как для n-делимых графов эффективно разрешима проблема допускания их машиной, для n-делимых гиперграфов тоже есть подобная проблема. Пусть каждому гиперребру l гиперграфа G соответствует конечная коммутативная полугруппа D_l и в ней есть подмножество D'. Пусть для каждого гиперребра l существует функция F_l из алфавита пометок вершин A в D_l. Тогда каждой вершине из l с пометкой a соответствует элемент $F_l(a) \in D_l$. Если для каждого гиперребра l сумма элементов, соответствующих всем его вершинам принадлежит D', то будем говорить, что есть допустимая разметка гиперграфа G.

ТЕОРЕМА 14. *Существует алгоритм, который по гиперграфу G и конечным коммутативным полугруппам D_l для каждого гиперребра l вместе с функциями F_l из алфавита разметки A в D_l и подмножествами $D' \subseteq D_l$ выясняет, существует ли допустимая разметка G или не существует. При фиксированном n для n-делимых гиперграфов время алгоритма полиномиально зависит от размера G и от $\max|D_l|$.*

ДОКАЗАТЕЛЬСТВО. Пользуясь алгоритмом из теоремы 13 находим минимальное n такое, что G n-делим, и строим процесс его деления. Этот процесс будем представлять в виде дерева деления. Пусть P — некоторый подгиперграф, возникший в процессе деления. Пусть F — функция, сопоставляющая каждому граничному к P гиперребру элемент из D_l. Назовем такую функцию граничной. Скажем, что F реализуема на P, если существует разметка вершин P такая, что сумма элементов D_l, соответствующих вершинам любого гиперребра l, лежащего целиком внутри P, принадлежит D', а для любого граничного к

P гиперребра l сумма элементов D_l, соответствующих его вершинам, лежащим в P, есть $F(l)$. Каждому подгиперграфу P в дереве деления поставим в соответствие множество реализуемых на нем граничных функций, а если у P нет граничных гиперребер, то также указание на то, существует ли допустимая разметка P. Идя по дереву деления от листьев к корню, будем вычислять эту информацию. Для одновершинных подгиперрафов в листьях она определяется непосредственно. Пусть мы хотим узнать множество реализуемых граничных функций для P, зная его для P_1 и P_2, где $P = P_1 \bigcup P_2$. Перебираем пары F_1, F_2 реализуемых функций для P_1 и P_2. Пара F_1, F_2 определяет некоторую реализуемую граничную функцию F для P тогда и только тогда, когда для любого гиперребра l, лежащего целиком в P, но не лежащего целиком ни в P_1, ни в P_2: $F_1(l) + F_2(l) \in D'$. В этом случае, для любого граничного к P гиперребра l: $F(l) = F_1(l) + F_2(l)$, где слагаемое $F_i(l)$ отсутствует, если P_i не пересекается с l. Обратно, любой реализации функции F для P соответствуют некоторые F_1,F_2. Таким образом, дойдя до корня дерева, ответим на вопрос о существовании допустимой разметки для G. Теорема 14 доказана.