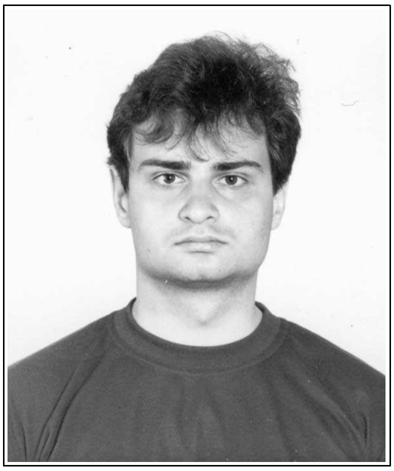

ОБ ОДНОЙ ВЗАИМОСВЯЗИ ФОРМАЛЬНЫХ ГРАММАТИК И СИСТЕМ ЛИНЕЙНЫХ ДИОФАНТОВЫХ УРАВНЕНИЙ

Д. Г. Корзун
аспирант Петрозаводского государственного университета

Научный руководитель
к.т.н., доц. Ю. А. Богоявленский

D. G. Korzun

В работе приводятся полученные нами на данный момент результаты, касающиеся изучения одной взаимосвязи между контекстно-свободными грамматиками и системами линейных диофантовых уравнений. Подобные исследования могут быть полезными при разработке новых алгоритмов для решения таких задач как целочисленная линейная оптимизация или решение систем линейных уравнений и неравенств в неотрицательных целых числах ■

A Certain Mapping Between Formal Grammars and Systems of Linear Diophantine Equations
This paper presents recent results on a certain mapping between context-free grammars and systems of linear diophantine equations. Such investigation looks to be useful for designing of new algorithms for integer linear optimizations or for solving system of linear equations and inequations in nonnegative integers ■

Введение

Постановка и анализ первых задач диофантова анализа восходят ещё к античному миру [1, 2], однако и в наше время эта область является богатым источником многочисленных интересных проблем, привлекающих внимание математиков. Изначально решения подобных задач искались среди натуральных чисел, но позднее к ним стали относить задачи поиска на более широких множествах — целых или рациональных чисел [3, 4, 5]. Последние множества являются кольцами, и это упрощает ситуацию. Например, в этом случае система линейных уравнений допускает использование алгоритма Гаусса или подобных ему методов последовательного преобразования исходной системы [6, 7]. Однако многие практические задачи [6, 8] требуют неотрицательности искомых решений. Далее мы будем рассматривать решения только на множестве $\mathbb{Z}_+ = \{0, 1, 2, \dots\}$. При этом задача нахождения решений системы линейных диофантовых уравнений (ЛДУ) является NP-полной [7]. Большинство существующих алгоритмов [9–15] заключаются в сведении исходной системы к однородной, для которой вычисляется конечный базис (так называемый базис Гильберта), после чего общее решение системы получается как неотрицательная линейная комбинация базисных.

В данной статье излагаются результаты, полученные на базе идей португальских математиков M. Filgueiras и A. Tomás, которыми была предпринята попытка [16] рассмотреть задачу анализа системы ЛДУ, применяя аппарат теории формальных языков. Доказательства приводимых результатов можно найти в [17, 18, 19]. Мы используем понятия и терминологию теории формальных языков, принятые в [20].

1. Система ЛДУ, ассоциированная с грамматикой

В данной статье под системой ЛДУ мы понимаем систему линейных уравнений с целыми коэффициентами, решения которой ищутся в множестве неотрицательных целых чисел $\mathbb{Z}_+$.

В работе [16] описывается метод построения системы ЛДУ по произвольной КС-грамматике[1] $G = (\mathrm{N}, \Sigma, \mathrm{P}, S)$ и цепочке $x \in \Sigma^*$. Такая система имеет вид

$$\begin{cases} \sum\limits_{i \in H_S} \xi_i - \sum\limits_{k=1}^{n} c_{Sk}\xi_k = 1\,, \\ \sum\limits_{i \in H_A} \xi_i - \sum\limits_{k=1}^{n} c_{Ak}\xi_k = 0\,, \quad \forall\, A \in \mathrm{N} \setminus \{S\}\,, \\ \sum\limits_{k=1}^{n} c_{ak}\xi_k = b_a\,, \quad \forall\, a \in \Sigma\,. \end{cases} \tag{1.1}$$

Здесь H_A — множество номеров тех правил грамматики, в левой части которых находится нетерминал A. Формально:

$$H_A = \{i \mid p_i = (A \to \alpha),\ p_i \in P\} \ .$$

Величина c_{Ak} — число вхождений нетерминала A в правую часть правила p_k, c_{ak} — число появлений терминала a в правой части правила p_k, b_a — число вхождений нетерминала a в цепочку x. Компонента ξ_i вектора неизвестных определяет, сколько раз должно быть задействовано правило p_i при выводе цепочки x.

В [17] этот метод получил формальное обоснование на основе балансовых соотношений в лесе возможного разбора цепочки x в грамматике G. При этом грамматика G называется *порождающей*, а построенная система $D(G,x)$ — *ассоциированной*.

Интуитивно, данные уравнения означают, что при выводе терминальной цепочки каждый, отличный от начального, нетерминал должен появиться в сентенциальных формах ровно столько раз, сколько раз он раскрывается по некоторому правилу, поскольку порождаемая терминальная цепочка не содержит нетерминалов (2-е уравнение). Начальный же символ имеет на единицу больше вхождений в сентенциальные формы, нежели число его раскрытий, т. к. с него начинается вывод (1-е уравнение). В результате последовательного применения правил грамматики при выводе в сентенциальных формах появляются терминальные символы, число появлений которых в процессе вывода должно равняться их числу в выводимой цепочке, поскольку далее в выводе терминалы раскрываться не могут (3-е уравнение).

2. Свойства ассоциированной системы

После построения ассоциированной системы весьма естественно возникает задача определения зависимости её свойств от свойств порождающей грамматики и цепочки.

2.1. Выводимые цепочки и решения

Из построения следует, что любому правильному выводу терминальной цепочки x соответствует некоторое решение системы $D(G,x)$. В принципе это означает, что решение системы ЛДУ может быть получено с использованием алгоритмов грамматического разбора цепочки x в грамматике G. В тоже время отсюда не следует, что любому решению системы обязательно соответствует некоторый правильный вывод x. Относительно произвольного решения системы возможны три случая:

1. Это решение порождается некоторым выводом цепочки x.

2. Решение соответствует выводу цепочки $x' \neq x$.

3. Решение не соответствует никакому выводу.

“Неудачными” с точки зрения применения алгоритмов разбора здесь являются последние два случая. Первый из них возникает из-за того, что вектор $\vec{\xi}$ не содержит информации о порядке применения правил при выводе. Причину, приводящую к такой ситуации, поясняют следующие теоремы [17]:

Теорема 1 *Если некоторое произвольное решение $\vec{\xi}$ системы $D(G,x)$ соответствует выводу некоторой цепочки x', то цепочки x и x' равны с точностью до перестановки символов.*

Теорема 2 *Если цепочки x и x' равны с точностью до перестановки символов, то тогда системы $D(G,x)$ и $D(G,x')$ совпадают.*

Приведённые теоремы полностью решают возникающую во втором случае проблему, которая вызывается наличием других, отличных от исходной, выводимых терминальных цепочек, также соответствующих решениям системы. Даже если такие цепочки есть, то они совпадают с исходной с точностью до порядка составляющих её терминалов.

Третий случай из вышеперечисленных является более сложным и вызывается наличием в грамматике циклов: решение соответствует выводу некоторой цепочки плюс какое-то количество циклов.

2.2. Общий вид решения

Большую роль при анализе свойств ассоциированной системы играют леса, составленные из деревьев разбора, соответствующих выводам некоторых сентенциальных форм: $A \Rightarrow u$, где $A{\in}\mathrm{N}$, $u{\in}(\mathrm{N} \cup \Sigma)^*$. Особенно важен так называемый уравновешенный лес, введённый в [16], — это лес, в котором число появлений произвольного нетерминала среди корней леса равняется числу его появлений в кроне.

Допустим, что вектор $\vec{\xi}$ является некоторым решением системы $D(G,x)$. Этот вектор определяет, сколько раз должно быть задействовано каждое правило грамматики G. И обратно, если нам дан некоторый лес деревьев разбора, то можно построить целочисленный вектор, каждая компонента которого показывает, сколько раз применяется соответствующее правило.

Фундаментальное свойство ассоциированной системы раскрывается следующей теоремой [17]:

Теорема 3 *Целочисленный неотрицательный вектор $\vec{\xi}$ является решением ассоциированной системы $D(G,x)$ тогда и только тогда, когда его можно представить в виде:*

$$\vec{\xi} = \vec{\xi}_y + \vec{\xi}_c^0 + \vec{\xi}_c^1 \ , \tag{2.1}$$

где

1. *y — терминальная цепочка, выводимая в G;*
2. *y состоит только из терминалов, имеющихся в цепочке x, и каждый такой символ встречается в y не больше, чем в x;*
3. *T_y — любое дерево разбора для цепочки y;*
4. *$\vec{\xi}_y$ — вектор, определяемый деревом T_y;*
5. *$\vec{\xi}_c^0$ — вектор, определяемый уравновешенным лесом чисто циклических деревьев[2] F_c^0;*
6. *$\vec{\xi}_c^1$ — вектор, определяемый уравновешенным лесом циклических деревьев F_c^1 таким, что его терминальная крона состоит только из тех символов цепочки x, которые не попали в цепочку y.*

Доказательство теоремы основывается на свойствах так называемого леса решений, тесно связанного с понятием уравновешенного леса.

Выскажем несколько замечаний относительно данной теоремы.

1. Всегда можно добиться, чтобы лес F_c^0 состоял только из простых чисто циклических деревьев, то есть деревьев, соответствующих выводу $A \Rightarrow^+ A$, который не может быть разложен на 2 или более чистых циклов с тем же самым нетерминалом в основе: $A \Rightarrow^+ A \Rightarrow^+ A$.
2. Если $\vec{\xi} = \vec{\xi}_y + \vec{\xi}_c^0 + \vec{\xi}_c^1$ есть решение системы $D(G,x)$ и существует чисто циклическое дерево T в грамматике G с кроной, содержащей только один нетерминал, то, добавляя в лес F_c^0 произвольное число экземпляров этого дерева, опять получаем решение данной системы. Учитывая также и предыдущее замечание, приходим к выводу, что всегда

$$\vec{\xi}_c^0 = \sum_T k_T \xi\left(\{T\}\right) \ ,$$

где суммирование выполняется по всем простым чисто циклическим деревьям грамматики G, крона которых содержит только один нетерминал, а величины k_T принимают произвольные целые неотрицательные значения. Эта сумма конечна, т. к. число простых циклических деревьев в грамматике всегда конечно.
3. Если имеются две различные тройки: $\left(T_{y_1}, F_{c1}^0, F_{c1}^1\right)$ и $\left(T_{y_2}, F_{c2}^0, F_{c2}^1\right)$, то из этого не следует, что соответствующие этим тройкам решения $\vec{\xi}^1$ и $\vec{\xi}^2$ будут также различны.
4. Учитывая замечание 2, можно сделать вывод, что уравнение (2.1) является не чем иным как одним из вариантов записи решения через базис Гильберта [7], который определяется всеми простыми чисто циклическими деревьями грамматики G.

2.3. Вопросы разрешимости

Обозначим через $\{u\}^\circ$ мультимножество[3] символов цепочки u.

Опираясь на теорему 3 об общем решении системы $D(G,x)$ и основываясь на частных свойствах некоторых классов грамматик G и цепочек x, можно изучать вопросы разрешимости $D(G,x)$. В качестве примера приведем несколько утверждений.

[2]Здесь под циклическим деревом понимается дерево, соответствующее выводу $A \Rightarrow uAv$, $A \in \mathrm{N}$, $u, v \in \Sigma^*$. При этом, если цепочка uv пустая, то дерево называется чисто циклическим.

[3]Мультимножество — это множество элементов, в котором число вхождений одного и того же элемента может быть больше одного.

Утверждение 1 *Система $D(G,x)$ разрешима тогда и только тогда, когда существует цепочка y такая, что $\{y\}^\circ \subseteq \{x\}^\circ$ и существует лес циклических деревьев F_c (возможно, пустой), терминальная крона которого в точности совпадает с $\Theta = \{x\}^\circ \setminus \{y\}^\circ$.*

Утверждение 2 *Разрешимая система $D(G,x)$ имеет бесконечное число решений тогда и только тогда, когда грамматика G содержит чистые циклы.*

Утверждение 3 *Число решений системы $D(G,x)$ конечно, если G не содержит чистых циклов.*

Утверждение 4 *Пусть $L[G]$ есть язык грамматики G. Тогда если $L[G]$ не содержит ни одной цепочки y такой, что $\{y\}^\circ \subseteq \{x\}^\circ$, то система $D(G,x)$ несовместна.*

Утверждение 5 *Если G не содержит циклов, то любому решению системы $D(G,x)$ соответствует некоторый вывод цепочки y, где y совпадает с исходной цепочкой x с точностью до порядка символов.*

Утверждение 6 *Если $L(G) = \varnothing$, то система $D(G,x)$ несовместна для любой цепочки $x \in \Sigma^*$.*

Эти утверждения являются простыми следствиями теоремы 3.

Множество решений Sol системы ЛДУ определяется через базис Гильберта:

$$Sol = H + L(B)\ , \tag{2.2}$$

где H, B — конечные множества векторов, а $L(M)$ обозначает линейную комбинацию с неотрицательными коэффициентами элементов из M. В случае ассоциированной системы все возможные леса F_c^1 вместе с соответствующими деревьями T_y определяют множество H (конечная часть множества решений), а все простые чисто циклические деревья — множество B (бесконечная часть множества решений).

3. Метод порождающей грамматики

Метод порождающей грамматики заключается в решении системы ЛДУ с использованием свойств её взаимосвязи с некоторой КС-грамматикой.

Как было описано в п. 1, по любой КС-грамматике и терминальной цепочке можно построить систему ЛДУ. Оказывается, что и для произвольно взятой системы ЛДУ можно построить такую грамматику и цепочку, которые порождают новую систему, эквивалентную в определённом смысле[4] данной [19]. Однако предлагаемые при этом методы построения требуют знания множества решений заданной системы, выраженного через базис Гильберта в виде (2.2), что не позволяет использовать их для решения системы. Там же показано, что порождающая грамматика и цепочка определяются не единственным образом, а значит в некоторых частных случаях можно попытаться найти такую порождающую пару (G,x), применение к которой теоремы 3 позволит решить поставленную задачу.

В качестве примера использования метода порождающей грамматики рассмотрим уравнение[5]

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} a_i x_i + b\ . \tag{3.1}$$

Предполагаем, что $a_i \geqslant 0$ $(i = 1, 2, \dots, n)$, $b \geqslant 0$. Также не будем рассматривать случай, когда $a_k = 1$ для некоторого k, поскольку, вычитая из обеих частей уравнения (3.1) величину x_k, приходим к новому уравнению уже с $n-1$ переменной (не содержащему переменной x_k).

Введём новую переменную y и перепишем уравнение (3.1) в виде эквивалентной системы:

$$\begin{cases} \displaystyle\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} a_i x_i + by, \\ \qquad y = 1. \end{cases} \tag{3.2}$$

Система (3.2) является ассоциированной для грамматики

$$G = (\{S, A\}, \varnothing, \mathrm{P}, S)$$

[4]В данном случае под эквивалентными системами понимаются системы, которые позволяют по решениям одной из них однозначно восстановить решения другой и наоборот.

[5]В [18] рассматривается также уравнение вида $\sum_{i=1}^{n} x_i + b = \sum_{i=1}^{n} a_i y_i$, однако этот случай более сложный. Заметим также, что оба этих случая эквивалентны уравнению $\sum_{i=1}^{n} u_i + b_1 = \sum_{j=1}^{m} a_j v_j + b_2$, однако для применения метода удобнее первая форма записи.

и пустой цепочки $x = e$. При этом множество правил P задается следующим образом:

$$\begin{array}{rl} y: & S \longrightarrow A^b \\ x_1: & A \longrightarrow A^{a_1} \\ x_2: & A \longrightarrow A^{a_2} \\ & \dots \\ x_n: & A \longrightarrow A^{a_n} \end{array} \tag{3.3}$$

Здесь в левом столбце записаны переменные из (3.2), соответствующие правилам, указанным в правом столбце.

Если $b > 0$ и $a_i > 0$ $\forall i$, то язык грамматики пуст ($L[G] = \varnothing$), поскольку из начального нетерминала S нельзя вывести ни одной терминальной цепочки (в нашем случае — пустой цепочки). Отсюда следует, что в этом случае система (3.2) неразрешима в неотрицательных целых числах, а значит, и уравнение (3.1) не имеет решений.

Если $b = 0$ и $a_i > 0$ $\forall i$, то $L[G] = \{e\}$, поскольку 1-е правило грамматики при этих условиях имеет вид $S \to e$. Из элементарных деревьев для правил $A \to A^{a_i}$ нельзя составить уравновешенного леса, ибо в этом случае обязательно $a_i > 1$ и число появлений нетерминала A в кроне всегда будет превосходить число его появлений среди корней. По теореме 3 об общем решении для ассоциированной системы любое решение системы (3.2) имеет вид

$$\vec{\xi} = (y, x_1, \dots, x_n)^\top = \vec{\xi}_0 ,$$

где $\vec{\xi}_0$ — вектор, определяемый деревом разбора пустой цепочки в грамматике G. Поскольку такой вывод всего один, а именно $S \Rightarrow e$ (применяется только 1-е правило $S \to e$), то вектор $\vec{\xi}_0$ находится единственным образом:

$$\vec{\xi}_0 = (1, 0, \dots, 0)^\top ,$$

а значит, уравнение (3.1) в этом случае имеет лишь нулевое решение $x_i = 0$, $i = 1, 2, \dots, n$.

Пусть теперь $b > 0$ и некоторые (но не все) a_i равны 0. Ясно, что в этом случае всегда можно соответствующей перенумерацией переменных получить, что $a_1 = a_2 = \ldots = a_m = 0$, $a_{m+1} > 1, \ldots, a_n > 1$ $(0 < m < n)$. Тогда множество правил порождающей грамматики G имеет вид

$$\begin{array}{rl} y & : S \longrightarrow A^b \\ x_1 & : A \longrightarrow e \\ & \dots \\ x_m & : A \longrightarrow e \\ x_{m+1} & : A \longrightarrow A^{a_{m+1}} \\ & \dots \\ x_n & : A \longrightarrow A^{a_n} \end{array} \tag{3.4}$$

Заметим, что несмотря на то, что здесь имеется по меньшей мере m одинаковых правил (соответствующих переменным $x_1, \ldots x_m$), их нельзя объединять в одно, поскольку они соответствуют различным переменным, т. е. множество правил P надо рассматривать как мультимножество.

Для вывода пустой цепочки необходимо все нетерминалы A, появившиеся в сентенциальной форме после применения первого правила грамматики, раскрыть в пустую цепочку. При развёртывании нетерминала A мы можем применить любое из правил, соответствующих переменным $x_1, \ldots, x_m$. Всего нетерминалов A в этом дереве — b штук, следовательно, число всех возможных таких деревьев равняется количеству неотрицательных целых решений уравнения

$$k_1 + k_2 + \ldots + k_m = b . \tag{3.5}$$

Обозначим число этих деревьев как M_0. Им соответствует M_0 векторов

$$\vec{\xi}_0^{\,j} = (1, k_1, \dots, k_m, 0, \dots, 0)^\top , \quad k_1 + \ldots + k_m = b , \quad j = 1, \dots, M_0 .$$

Далее построим все возможные чисто циклические деревья для этой грамматики. Заметим, что уравновешенных лесов циклических деревьев с непустой кроной здесь не может быть, ибо $\Sigma = \varnothing$. Любое простое циклическое дерево в данном случае соответствует выводу $A \Rightarrow A^{a_l} \Rightarrow^+ A$ $(m + 1 \leqslant l \leqslant n)$. В качестве первого правила может быть выбрано любое из $(n - m)$ правил, соответствующих переменным $x_{m+1}, \ldots, x_n$. В результате появляются a_l нетерминалов A (l — номер выбранного первого правила для дерева), которые затем раскрываются в пустую цепочку, кроме одного (дерево циклическое). Для этого можно воспользоваться лишь правилами, соответствующими переменным $x_1, \dots, x_m$. Таким образом, если первое правило в дереве зафиксировано (пусть, для определённости, соответствующее x_l), то при этом условии число таких чисто циклических деревьев будет равняться числу неотрицательных целых решений уравнения

$$k_1 + k_2 + \ldots + k_m = a_l - 1 \,. \tag{3.6}$$

Обозначим число этих деревьев через M_l. Им соответствуют векторы

$$\vec{\xi}_c^{lr} = (0, k_1, \dots, k_m, 0, \dots, 1, \dots, 0)^\top \,, \quad k_1 + \ldots + k_m = a_l - 1 \,,$$
$$l = m+1, \dots, n \,, \quad r = 1, \dots, M_l \,,$$

где 1 стоит на месте, соответствующем переменной x_l.

Согласно теореме 3 об общем решении ассоциированной системы, получаем формулу для общего решения системы (3.2)

$$\vec{\xi} = \vec{\xi}_0^j + \sum_{l=m+1}^{n} \sum_{r=1}^{M_l} p_{lr} \vec{\xi}_c^{lr} \,, \quad j \in \{1, 2, \dots, M_0\} \,, \quad p_{lr} \geqslant 0 \,. \tag{3.7}$$

Формула (3.7) и определяет все решения системы (3.2), а значит, и уравнения (3.1).

Пусть теперь $\vec{x}_0^j$ и $\vec{x}_c^{lr}$ — это соответственно вектора $\vec{\xi}_0^j$ и $\vec{\xi}_c^{lr}$ без первой координаты y. Тогда решение уравнения (3.1) складывается из двух компонент. Первая — это вектор $\vec{x}_0^j$ для некоторого j из $\{1, \dots, M_0\}$. Число таких векторов конечно и равняется M_0. Вторая компонента решения — $\sum_{l=m+1}^{n} \sum_{r=1}^{M_l} p_{lr} \vec{\xi}_c^{lr}$, где p_{lr} $(r = 1, \dots, M_l, l = m+1, \dots, n)$ — произвольные неотрицательные целые, а поэтому она может принимать бесконечное число значений.

Как показано в [18], искомый базис Гильберта для множества решений Sol уравнения (3.1) строится из векторов $\vec{x}_0^j$ $(j = 1, \dots, M_0)$ и $\vec{x}_c^{lr}$ $(r = 1, \dots, M_l, l = m+1, \dots, n)$:

Теорема 4 *Общее решение уравнения (3.1) определяется формулой:*

$$Sol = H + L(B) \,, \tag{3.8}$$

где $H = \{\vec{x}_0^j \mid j = 1, \dots, M_0\}$ *и* $B = \{\vec{x}_c^{lr} \mid r = 1, \dots, M_l, l = m+1, \dots, n\}$.

При этом, в силу свойств базиса Гильберта, не существует отличных от H и B множеств, которые бы удовлетворяли (3.8). Более того, элементы из H есть минимальные решения[6] уравнения (3.1), а элементы множества B — минимальные решения однородного уравнения для (3.1).

Используя стандартную комбинаторную технику, можно определить число векторов в множествах H и B.

Теорема 5 *Число минимальных решений неоднородного уравнения (3.1) определяется формулой:*

$$M_0 = |H| = C_{m+b-1}^{b} = \frac{(m+b-1)!}{b!(m-1)!} \,.$$

Теорема 6 *Число минимальных базисных решений однородного уравнения для (3.1) определяется формулой:*

$$|B| = \sum_{l=m+1}^{n} C_{m+a_l-2}^{a_l-1} = \sum_{l=m+1}^{n} \frac{(m+a_l-2)!}{(a_l-1)!(m+a_l-2)!} \,.$$

Итак, решение уравнения (3.1) свелось к решению уравнений (3.5) и (3.6), которые имеют вид:

$$\sum_{j=1}^{m} k_j = d \,. \tag{3.9}$$

Для подобного уравнения известен алгоритм линейный относительно числа решений.

Используемый здесь метод порождающей грамматики для решения диофантовых уравнений вида (3.1) в неотрицательных целых числах имеет довольно прозрачную интерпретацию с точки зрения обычной алгебраической техники: исходное уравнение расщепляется на несколько уравнений вида (3.9), каждое из которых решается по отдельности. После этого любое решение получается как некоторая сумма найденных решений расщеплённых уравнений.

[6] Решение $\vec{u}$ называется минимальным, если нельзя найти другое решение $\vec{v} \neq \vec{u}$ такое, что $\vec{u} \geqslant \vec{v}$. Вектора здесь сравниваются покомпонентно.

Заключение

В данной работе приводятся основные, полученные нами на данный момент результаты о взаимосвязи КС-грамматик и систем ЛДУ и демонстрируется, насколько успешно они могут быть использованы.

Несмотря на то, что формула (2.1) в общем случае сложна для практического применения, она может оказаться полезной в некоторых частных случаях, а также для теоретических исследований.

Метод порождающей грамматики, по мнению автора, имеет большие потенциальные возможности применительно к созданию новых алгоритмов решения ЛДУ. Необходимо дальнейшее его развитие с привлечением известных алгоритмов грамматического разбора.

Литература

1. *Башмакова И. Г.* История диофантова анализа от Диофанта до Ферма. М.: Наука, 1984. 256 с.
2. *Башмакова И. Г.* Диофант и диофантовы уравнения. М.: Наука, 1972. 68 с.
3. *Боревич З. И., Шафаревич И. Р.* Теория чисел. М.: Наука, 1972. 495 с.
4. *Серпинский В.* О решении уравнений в целых числах. М.: Гос. изд-во физ.-мат. лит., 1961. 88 с.
5. *Гельфонд А. О.* Решение уравнений в целых числах. М.: Наука, 1983. 63 с.
6. *Кофман А., Анри-Лабордер А.* Методы и модели исследования операций. М.: Мир, 1977. 432 с.
7. *Схрейвер А.* Теория линейного и целочисленного программирования. Т. 1, 2, М.: Мир, 1991. 702 с.
8. *Comon H., Dincbas M.* A Methodological View of Constraint Solving, 1995. (`ftp://ftp.lri.fr/LRI/articles/comon/casm.ps.Z`)
9. *Ajili F., Contejean E.* Avoiding Slack Variables in the Solving of Linear Diophantine Equations and Inequations // Technical Report, 1996. (`ftp://ftp.lri.fr/LRI/articles/contejean/tcs97.ps.gz`).
10. *Boudet A., Comon H.* Diophantine Equations, Presburger Arithmetic and Finite Automata // Proceedings of the 21st International Colloquim on Trees in Algebra and Programming (CAAP'96). 1996. P. 30–43.
11. *Clausen M., Fortenbacher A.* Efficient Solution of Linear Diophantine Equations // J. Symbolic Computations. V. 8 (1&2). 1989. P. 201–216.
12. *Contejean E., Devie H.* An Efficient Incremental Algorithm for Solving Systems of Linear Diophantine Equations // Information and Computation. V. 13. №. 1. 1994. P. 143–172.
13. *Huet G.* An Algorithm to Generate the Basis of Solutions to Homogeneous Linear Diophantine Equations // Informational Processing Letters. Vol. 3. №. 7. 1978. P. 144–147.
14. *Pottier L.* Minimal Solutions of Linear Diophantine Systems: Bounds and Algorithms // Proceedings of the 4th International Conference on Rewriting Techiques and Applications (RTA'91). Como (Italy), 1991. P. 162–173.
15. *Tomás A., Filgueiras M.* A New Method for Solving Linear Constraints on the Natural Numbers // Proc. of EPTA'91, 1991. P. 1–16.
16. *Filgueiras M., Tomás A.* Solving Linear Constraints on Finite Domains through Parsing // Proc. of EPTA'91, 1991. P. 32–40.
17. *Богоявленский Ю. А., Корзун Д. Г.* Общий вид решения системы линейных диофантовых уравнений, ассоциированной с контекстно-свободной грамматикой // Труды Петрозаводского гос. ун-та. Серия "Прикладная математика и информатика". Вып. 6. Петрозаводск: Изд-во ПетрГУ, 1997. С. 79–94.
18. *Корзун Д. Г.* Решение одного класса линейных диофантовых уравнений в неотрицательных целых числах методами теории формальных языков // Труды Петрозаводского гос. ун-та. Сер. "Прикладная математика и информатика". Вып. 7. Петрозаводск: Изд-во ПетрГУ, 1998. С. 101–124.
19. *Корзун Д. Г.* О существовании порождающей КС-грамматики для произвольной линейной диофантовой системы // Труды Петрозаводского гос. ун-та. Сер. "Математика". Вып. 6. Петрозаводск: Изд-во ПетрГУ, 1999. С. 172–179.
20. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т. 1. М.: Мир, 1978. 612 с.

■ Поступила в редакцию 20 августа 1999 года; в окончательном варианте 17 января 2000 года.