4 Conclusion

Our approach has the following advantages:

1. It allows to reduce the labour intensity of other application component development of our information management system.
2. It allows to apply a powerful mechanism of the formal grammars to solve our task.
3. It separates the description of a structure from realization details and tools.
4. We can apply this method to other objects (not only the documentation set of a department). So we may consider this way as universal in that context.

The disadvantages are as follows:

1. Labouriousness in making grammars.
2. Specific requirements to developers such as skills in formal language theory.

Bibliography

[1] F. Bauer, G. Gooze, *Informatika. Vvodny kurs.* In Russian; translated from German. Vol. 2, Moscow: Mir, 1990.

[2] T. Kasai. *An hierarchy between context-free and context-sensitive languages.* Journal of Computer and System Sciences 4 (1970), pp. 492–508.

[3] D. J. Rosenkrantz. *Programmed grammars and classes of formal languages.* J. Assoc. Comput. Mach., 16 (1969), pp. 107–131.

[4] F. Boumphrey, O. Direnzo, J. Duckett. *XML. Novye perspektivy WWW.* Moscow: DMK, 2000. 688 p. (in Russian)

Generating Homogeneous Systems of Equations for Testing and Experimental Analysis of Linear Diophantine Solvers

Kirill A. Kulakov, Dr. Dmitry G. Korzun

Department of Computer Science, Petrozavodsk State University

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185910, Russia

E-mail: {`kulakov`, `dkorzun`}`@cs.karelia.ru`

Abstract

The paper addresses the problem of generating homogenous systems of non-negative linear Diophantine equations for a particular class: systems associated with context-free grammars. We give a general theoretical base for solving this problem and, as particular solutions of this problem, present two algorithms for generation. Applicability of the proposed theoretical and algorithmic results is shown for automated testing and experimental analysis of the syntactic solver—an implementation of a special algorithm for solving any homogenous linear Diophantine system of the considered class through parsing in the CF-grammar.

Contents

1 Preliminaries

1.1 Homogeneous non-negative linear Diophantine equations

Let $\mathbb{Z}$ be the set of all integers, $\mathbb{Z}_+$ be the set of all non-negative integers, and $\mathbb{N}$ be the set of all natural numbers ($\mathbb{N} = \mathbb{Z}_+ \setminus \{0\}$). A homogeneous system of non-negative linear Diophantine equations (homNLDE system) is defined as the linear system

$$Ax = \mathbb{O}\,, \tag{1.1}$$

where $A \in \mathbb{Z}^{n\times m}$ is a matrix of the coefficients, $\mathbb{O}$ is the null vector of $\mathbb{Z}^n$, $n \in \mathbb{N}$ is the number of equations, and $m \in \mathbb{N}$ is the number of unknowns. Diophantness means that coefficients and solutions of (1.1) are integer. Non-negativeness states the extra requirement $x \geq \mathbb{O}$ for any solution of (1.1).

A homNLDE system has at least the trivial null solution $x = \mathbb{O}$. A non-null solution $h \in \mathbb{Z}_+^m$ of (1.1) is *indecomposable*, if it is not a sum of two non-null solutions of (1.1). All indecomposable solutions form the following finite set:

$$\mathcal{H} = \{h^{(1)}, \dots, h^{(q)}\},$$

where q is the number of solutions. This set is called *Hilbert basis*, according to [1]. Any solution x of a homANLDE system can be represented (ambiguously) as a linear non-negative integer combination of basis ones:

$$x = \sum_{s=1}^{q} c_s h^{(s)}\ , \qquad \text{for some } c_s \in \mathbb{Z}_+\,,\ s = 1, 2, \dots, q.$$

Searching Hilbert basis is one of the central problems in linear Diophantine analysis. It is related to a lot of interesting algorithmic and complexity issues, see [2] for a survey of the area.

Example 1. For $n = 3$ and $m = 5$ consider the homNLDE system:

$$\begin{cases} x_1 + x_3 - 2x_4 - x_5 = 0 \\ 5x_1 - x_2 + 7x_3 + 2x_4 + 3x_5 = 0 \\ x_4 - 2x_5 = 0 \end{cases}$$

Its Hilbert basis contains the following six solutions:

$$\mathcal{H} = \left\{ \begin{pmatrix} 5\\32\\0\\2\\1 \end{pmatrix}, \begin{pmatrix} 4\\34\\1\\2\\1 \end{pmatrix}, \begin{pmatrix} 3\\36\\2\\2\\1 \end{pmatrix}, \begin{pmatrix} 2\\38\\3\\2\\1 \end{pmatrix}, \begin{pmatrix} 1\\40\\4\\2\\1 \end{pmatrix}, \begin{pmatrix} 0\\42\\5\\2\\1 \end{pmatrix} \right\}.$$

The particular solution $x = (5, 74, 5, 4, 2)^{\mathsf{T}}$ can be represented as the following linear non-negative integer combinations:

$$x = h^{(1)} + h^{(6)} = h^{(2)} + h^{(5)} = h^{(3)} + h^{(4)}\,.$$

1.2 Associated with CF-grammar a homANLDE system

homNLDEs and their systems are topical subject of research, see for instance [2–7]. This paper concentrates on a particular class of (1.1), introduced by the second author of this paper [8].

Let $I = I^{n,m} = \{I_0, I_1, \dots, I_n\}$ be a partition of $\mathbb{N}_m = \{1, \dots, m\}$ for $n, m \in \mathbb{N}$, $n \leq m$:

$$I_k \cap I_j = \varnothing \quad \forall\, 0 \leq k < j \leq n\,, \qquad I_k \neq \varnothing \quad \forall\, k \in \mathbb{N}_n\,, \qquad \bigcup_{k=0}^{n} I_k = \mathbb{N}_m\,,$$

where I_0 may be empty.

This partition can be represented by a boolean $n \times m$ matrix $\mathrm{E} = \mathrm{E}^{n,m} = \mathrm{E}(I^{n,m})$ such that $\mathrm{E}_{ki} = 1$ if $i \in I_k$, and $\mathrm{E}_{ki} = 0$ otherwise for $k = 1, 2, \dots, n$ and $i = 1, 2, \dots, m$.

Note that if $I_0 \neq \varnothing$, then the corresponding columns in E are zero.

Consider the following homNLDE system of n equations in m unknowns:

$$\mathrm{E}(I^{n,m})x = \mathrm{A}x\,, \tag{1.2}$$

where $\mathrm{A} \in \mathbb{Z}_+^{n\times m}$ is a matrix of non-negative integer coefficients for the right-hand side.

According to [8], homNLDE system (1.2) is called *a homogeneous system of associated with a context-free grammar NLDEs* (or a homANLDE system). These systems are the target subject of this paper.

The case $I_0 \neq \varnothing$ is simply reduced to the case with $I_0 = \varnothing$ by adding the corresponding I_0-unknowns to an arbitrary equation. Therefore, we assume below that always $I_0 = \varnothing$.

System (1.2) is associated with a CF-grammar as follows. The grammar has m rules $(r_i)_{i=1}^m$ and n nonterminals $(A_k)_{k=1}^n$. The terminal alphabet is empty. For each nonterminal A, $|\alpha|_A$ is the number of occurrences of A in string α. The Parikh-map Ψ_n from $\{A_1, A_2, \ldots, A_n\}^*$ onto $\mathbb{Z}_+^n$ is defined by $\Psi_n(\alpha) = (|\alpha|_{A_1}, |\alpha|_{A_2}, \ldots, |\alpha|_{A_n})$. Grammar rule r_i is $A_k \to \alpha_i$ for some $k \in \{1, 2, \ldots, n\}$ and nonterminal string α_i, and this leads to $\mathrm{E}_{ki} = 1$, $\mathrm{E}_{ji} = 0$ $\forall\, j \neq k$ and the i-th column of the matrix A to be $\Psi_n(\alpha_i)$.

Therefore, unknowns are associated with grammar rules and equations—with nonterminals. The value of x_i determines the number of applications of rule r_i in a derivation. There is a mapping between the set of all solutions of (1.2) and the set of all cycle derivations $A_k \Rightarrow^* A_k$, $k = 1, 2, \ldots, n$. Any solution has the corresponding derivation(s) and vice versa; see [8] for the proofs, some examples can also be found in [9]. The original idea came from M. Filgueiras and A. P. Tomás [10].

Example 2. Consider the homNLDE system from Example 1. It can be rewritten in the homANLDE form as

$$\begin{cases} x_1 + x_3 &= 2x_4 + x_5 \\ x_2 &= 5x_1 + 7x_3 + 2x_4 + 3x_5 \\ x_4 + x_5 &= 3x_5 \end{cases}$$

where $n = 3$, $m = 5$, $I_0 = \varnothing$, $I_1 = \{1, 3\}$, $I_2 = \{2\}$, $I_3 = \{4, 5\}$,

$$\mathrm{E}(I^{3,5}) = \mathrm{E}(I_1, I_2, I_3) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ and } \mathrm{A} = \begin{pmatrix} 0 & 0 & 0 & 2 & 1 \\ 5 & 0 & 7 & 2 & 3 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}.$$

The corresponding CF-grammar is

$$\begin{array}{ll} r_1: & A \to B^5 \\ r_2: & B \to \varepsilon \\ r_3: & A \to B^7 \\ r_4: & C \to A^2 B^2 \\ r_5: & C \to A B^3 C^3 \end{array}$$

Basis solution $h^{(1)} = (5, 32, 0, 2, 1)^{\mathsf{T}}$ is for instance mapped to the cycle derivation $C \overset{r_5}{\Longrightarrow} AB^3C^3 \overset{2\times r_4}{\Longrightarrow} A^5B^7C \overset{5\times r_1}{\Longrightarrow} B^{32}C \overset{32\times r_2}{\Longrightarrow} C$.

1.3 HomANLDE system $\sum_i x_i = \sum_j a_j y_j$

A particular subclass of homANLDE systems is a single equation of the following form:

$$\sum_{i=1}^{m_1} x_i = \sum_{i=1}^{m_2} a_i y_i\,, \qquad a_1, a_2, \ldots, a_{m_2} \in \mathbb{N}\,. \tag{1.3}$$

This equation we shall meet in Section 2, because some homANLDE systems can be reduced to it. This equation is analytically solvable [13,14] and allows efficient application of Theorem 1.

We limit this paper with a particular case of (1.3), which will be used in Section 3 for Algorithm 4:

$$x = \sum_{i=1}^{p} a_i y_i\,. \tag{1.4}$$

This equation has Hilbert basis:

$$\begin{pmatrix} x \\ y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix} : \ \mathcal{H} = \left\{ \begin{pmatrix} a_1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} a_2 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} a_p \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \right\}.$$

1.4 Problem of generating homANLDE systems and its applications

An algorithm for solving homANLDE systems was proposed in [8], it is based on syntactic properties and uses parsing in the CF-grammar, associated with a given homANLDE system, to search solutions of (1.2), see also [9]. This algorithm is a promising tool for solving some large-dimensional problems. The aim of our research is to validate the implementation of the algorithm (the syntactic solver) and to experiment with it measuring its complexity and comparing with available alternatives.

This goal requires a tool to generate automatically: 1) test homANLDE systems with known Hilbert basis, which is needed to test a solver for correctness of a produced solution, and 2) sample homANLDE systems with a given property, which used to estimate the solver complexity for this particular subclass of homANLDE systems.

This moves us to the problem of generating a homANLDE system with input parameters like the number of equations n and unknowns m. It is possible to define some other limits like the maximum number of basis solutions q, the maximum value of coefficients in the matrix A, etc.

The problem is not so easy from a complexity point of view as it looks at first. It can not be reduced to simple random generation of matrices E and A. Firstly, this naive way results in homANLDE systems that are usually (the overwhelming part of generated ones) inconsistent (has empty Hilbert basis) and thus, not adequate for testing and analysis applications. Secondly, for testing there is a need to know the Hilbert basis of the generated system and use it as a primary pattern to check solutions produced by a solver under testing. Thirdly, experimental analysis requires restricting the generated sample homANLDE systems to a particular class; the corresponding limits often need knowledge of the Hilbert basis, the number of solutions in it or the maximum of components in basis solutions.

Therefore, the problem of generating meets the same complexity difficulties as the problem of solving. In this paper we introduce an interesting theoretical result (see Section 2) that allows to solve the problem of generating on a general base. Some solutions are presented in Section 3 in form of algorithms for generating homANLDE systems. In Section 4 we describe their applications to testing and experimental analysis of the syntactic solver.

2 Basic theorem

In this section we introduce the key theoretical result of the paper. The idea is a transformation of an arbitrary homANLDE system to a reduced one through a sequence of intermediate homANLDE systems. The final system can be solved easier because of smaller dimensions and certain properties. Solutions of the reduced system together with the transformation can be used for solving the initial system by back-substituting and computing solutions of intermediate systems in the inverse order of the transformation.

Theorem 1. *The problem of searching Hilbert basis for homANLDE system (1.2) can be reduced to the problem of searching Hilbert basis either for the equation:*

$$\sum_{i\in I_k^*} x_i = \sum_{i\in J} a_{ki}x_i \quad \textit{for some } k\in\mathbb{N}_n\,, \tag{2.1}$$

where $I_k^*\subseteq I_k$ and $J\subseteq\mathbb{N}_m$; or for the system of equations:

$$\sum_{i\in I_{k_l}^*} x_i = \sum_{i\in J_l} x_i\,, \quad l=1,\dots,p \quad \textit{for some } p\in\mathbb{N}_n\,, \tag{2.2}$$

where $I_{k_l}^*\subseteq I_{k_l}$ and $\bigcup_{l=1}^{p} J_l = \bigcup_{l=1}^{p} I_{k_l}^* \subseteq \bigcup_{k=1}^{n} I_k = \mathbb{N}_m$.

The proof is based on explicitly constructing the required transformation. At each step one equation and some unknowns are removed from the current homANLDE system, see details in [12]. Hence, the transformation finally results in a homANLDE system with smaller dimensions (the number of equations and unknowns). Moreover, in the case of (2.1) this equation is analytically solvable (see Section 1.3), and in the case of (2.2) coefficients of the reduced system are $\{0,1\}$.

Searching the Hilbert basis for (1.2) can, therefore, start from searching the Hilbert basis for simpler cases (2.1) or (2.2) and then computing the Hilbert bases of intermediate homANLDE systems in accordance with the inverse order of the transformation. The last computed Hilbert basis is a target solution of the initial problem.

The following examples show the essentials of this transformation.

Example 3. Let us consider the system from the previous example:

$$\begin{cases} x_1+x_3 &= 2x_4+x_5\\ x_2 &= 5x_1+7x_3+2x_4+3x_5\\ x_4+x_5 &= 3x_5 \end{cases} \tag{$*$}$$

The unknown x_2 appears only in one equation (2nd equation). Removing this equation and unknown x_2, one has the simpler system:

$$\begin{cases} x_1+x_3 &= 2x_4+x_5\\ x_4+x_5 &= 3x_5 \end{cases} \tag{$*$: 1-reduced}$$

Now x_1 and x_3 are presented only in one equation (1st one). Removing this equation and unknowns x_1 and x_3, one gets the only equation:

$$x_4+x_5 = 3x_5\,, \tag{$*$: 2-reduced}$$

which is (2.1). Eq. ($*$: 2-reduced) has Hilbert basis with the only solution $(x_4,x_5)^\mathsf{T} = (2,1)^\mathsf{T}$. Back-substituting them to ($*$: 1-reduced) gives $x_1+x_3=5$ and the following Hilbert basis of ($*$: 1-reduced):

$$\begin{pmatrix}x_1\\x_3\\x_4\\x_5\end{pmatrix} : \left\{ \begin{pmatrix}5\\0\\2\\1\end{pmatrix}, \begin{pmatrix}4\\1\\2\\1\end{pmatrix}, \begin{pmatrix}3\\2\\2\\1\end{pmatrix}, \begin{pmatrix}2\\3\\2\\1\end{pmatrix}, \begin{pmatrix}1\\4\\2\\1\end{pmatrix}, \begin{pmatrix}0\\5\\2\\1\end{pmatrix} \right\}.$$

This finally leads to the Hilbert basis of $(*)$:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} : \mathcal{H} = \left\{ \begin{pmatrix} 5 \\ 32 \\ 0 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 4 \\ 34 \\ 1 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 36 \\ 2 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 38 \\ 3 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 40 \\ 4 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 42 \\ 5 \\ 2 \\ 1 \end{pmatrix} \right\}.$$

Example 4. Let us consider another homANLDE system:

$$\begin{cases} x_1 + x_5 = x_2 + x_3 \\ x_3 + x_4 = x_1 + 4x_5 \\ x_2 = x_4 + 2x_5 \\ x_6 = x_1 + 2x_3 + x_4 \end{cases} \qquad (**)$$

The unknown x_6 is only in one equation (4th equation). Remove this equation and the unknown:

$$\begin{cases} x_1 + x_5 = x_2 + x_3 \\ x_3 + x_4 = x_1 + 4x_5 \\ x_2 = x_4 + 2x_5 \end{cases} \qquad (**\text{: 1-reduced})$$

Sum of all equations for the last system gives the aggregate equation

$$0 = 5x_5 .$$

Thus, $x_5 = 0$ for any solution of ($**$: 1-reduced) and the system is finally reduced to

$$\begin{cases} x_1 = x_2 + x_3 \\ x_3 + x_4 = x_1 \\ x_2 = x_4 \end{cases} \qquad (**\text{: 2-reduced})$$

which is (2.2).

The Hilbert basis of ($**$: 2-reduced) consists of the following two solutions $(x_1, x_2, x_3, x_4)^\mathsf{T} = (1, 1, 0, 1)^\mathsf{T}$ and $(x_1, x_2, x_3, x_4)^\mathsf{T} = (1, 0, 1, 0)^\mathsf{T}$. The Hilbert basis of ($**$: 1-reduced) is computed by adding $x_5 = 0$ to these solutions. For the initial system ($**$) one should compute x_6 according to the 4th equation and the target Hilbert basis is

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} : \mathcal{H} = \left\{ \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 3 \end{pmatrix} \right\}.$$

3 Algorithms for generating homANLDE systems

We developed and proved three algorithms for generating homANLDE systems of certain subclasses of the whole homANLDE class. Figure 1 shows the relation between them. These algorithms can be proved independently while Theorem 1 gives a general base for all of them.

In this section we briefly describe the essentials of the first two algorithms. The third algorithm is an enhancement of the second one and for brevity we skip its description here; see [12, 13] for details and formal proofs. Moreover, [13] contains the fourth algorithm, which is able to generate any of homANLDE systems, i.e., it is a generator for the whole homANLDE class.

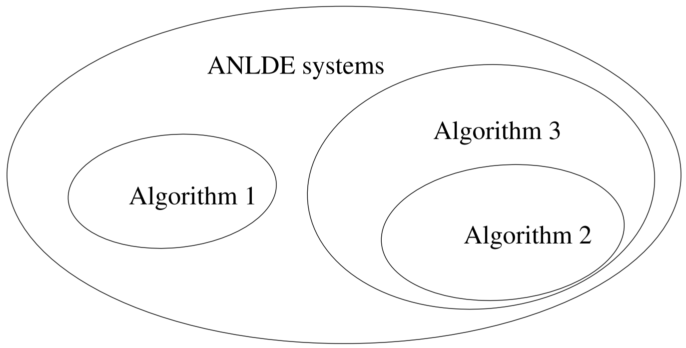


Figure 1: *Subclasses of homANLDE systems, generated by the developed algorithms, and their relations*

3.1 Algorithm 1: An analogue of the Gauss–Jordan elimination

Let n and m be fixed and $0 < n \le m$. Algorithm 3 shows a formal description of the first algorithm.

The idea is in construction of a homANLDE system (1.2) with Hilbert basis that contains first n canonical unit vectors[1] of $\mathbb{Z}^m$:

$$\mathcal{H} = \{e_1, e_2, \dots, e_n\} . \qquad (3.1)$$

[1] $e_{ki} = 1$ if $k = i$, otherwise $e_{ki} = 0$; $k = 1, 2, \dots, n$; $i = 1, 2, \dots, m$.

Algorithm 3 A Gauss–Jordan analogue for generating a homANLDE system and its Hilbert basis

Require: n is the number of equations, m is the number of unknowns, $0 < n \leq m$.

Ensure: $(\mathrm{E}^{n,m}, \mathrm{A}, \mathcal{H})$.

1: $\mathcal{H} := \{e_1, e_2, \ldots, e_n\}$;

2: generate randomly matrix $B \in \{0,1\}^{n\times(m-n)}$ that satisfies (3.2);

3: generate randomly matrix $\Delta \in \mathbb{Z}_+^{n\times(m-n)}$ that satisfies (3.3);

4: $\mathrm{E}^{n,m} := (\mathbb{I} \mid B)$; $\quad \mathrm{A} := (\mathbb{I} \mid B + \Delta)$;

The matrix $\mathrm{E}^{n,m}$ of left-hand side coefficients (1.2) is $\mathrm{E}^{n,m} = (\mathbb{I} \mid B)$, where $\mathbb{I}$ is the identity $n \times n$ matrix, $B \in \{0,1\}^{n\times(m-n)}$ is arbitrary matrix that satisfies the following condition:

$$\sum_{k=1}^{n} B_{ki} = 1 \qquad \forall\, i = 1, 2, \ldots, m-n\,. \tag{3.2}$$

It guarantees the left-hand side satisfies homANLDE property.

The matrix A for right-hand side coefficients of (1.2) is $\mathrm{A} = (\mathbb{I} \,|\, B{+}\Delta)$, where $\mathbb{I}$ and B are the same as for $\mathrm{E}^{n,m}$ and the matrix $\Delta \in \mathbb{Z}_+^{n\times(m-n)}$ satisfies the condition:

$$\sum_{k=1}^{n} \Delta_{ki} > 0 \qquad \forall\, i = 1, 2, \ldots, m-n\,. \tag{3.3}$$

This condition is equivalent to the property that homANLDE system $\sum_{i=1}^{m-n} \Delta_{ki} x_{n+i} = 0$, $k = 1, 2, \ldots, n$ has the only solution $x = \mathbb{O}$. However, (3.3) is easier for random generating Δ.

Finally, the constructed system has the following form:

$$(\mathbb{I} \,|\, B)x = (\mathbb{I} \,|\, B + \Delta)x\,. \tag{3.4}$$

Assuming that $x = \begin{pmatrix} x' \\ x'' \end{pmatrix}$ this system can be simplified to

$$\mathbb{I}x' = \mathbb{I}x' + \Delta x''\,.$$

Obviously $x'' = \mathbb{O}$ for any solution (due to (3.3) and $\Delta \geq \mathbb{O}$), and the system is reduced to the trivial (identical) case:

$$(\mathbb{I} \mid \mathbb{O}) \cdot \begin{pmatrix} x' \\ \mathbb{O} \end{pmatrix} = (\mathbb{I} \mid \mathbb{O}) \cdot \begin{pmatrix} x' \\ \mathbb{O} \end{pmatrix},$$

which evidently has Hilbert basis (3.1). This is a straight scheme to prove Algorithm 3.

Another way is applying Theorem 1 to (3.4). The transformation results in the identical system $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i$, which is (2.2). The intermediate transformation steps lead to $x_{n+i} = 0$ for $i = 1, 2, \ldots, m-n$.

System (3.4) is associated with a CF grammar that has rules

$$\begin{aligned} r_i &= A_i \to A_i \quad \text{for } i = 1, 2, \ldots, n\,, \\ r_{n+i} &= A_k \to A_1^{c_1} A_2^{c_2} \cdots A_n^{c_n} \quad \text{for } i = 1, 2, \ldots, m-n\,, \end{aligned}$$

where $(c_1, c_2, \ldots, c_n)^\mathsf{T}$ is the i-th column of the matrix $B + \Delta$; k is an index of the row, which contains the only 1 in the i-th column of B. The simplest (one step) cycle derivations are $A_i \stackrel{r_i}{\Rightarrow} A_i$ for $i = 1, 2, \ldots, n$, and they correspond to all basis solutions (3.1).

Example 5. **Input:** $n = 3$ and $m = 4$.

Step 1: $\mathcal{H} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$. **Step 2:** $B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$. **Step 3:** $\Delta = \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}$.

Step 4: $\mathrm{E}^{3,4} = (\mathbb{I} \mid B) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$, $\quad \mathrm{A} = (\mathbb{I} \mid B{+}\Delta) = \begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{pmatrix}$.

Finally we have the following homANLDE system and the CF-grammar:

$$\begin{cases} x_1 + x_4 = x_1 + 3x_4 \\ \phantom{x_1 + {}} x_2 = x_2 + 2x_4 \\ \phantom{x_1 + {}} x_3 = x_3 + 3x_4 \end{cases} \qquad \begin{aligned} r_1 &: A \to A \\ r_2 &: B \to B \\ r_3 &: C \to C \\ r_4 &: A \to A^3 B^2 C^3 \end{aligned}$$

System (3.4) is similar to linear systems constructed by the Gauss–Jordan elimination method. More precisely, unknowns $x' = (x_1, \ldots, x_n)^\mathsf{T}$ play a role of free variables and may take arbitrary non-negative integer values; $x'' = (x_{n+1}, \ldots, x_m)^\mathsf{T}$ are leading variables, which, in the case of (3.4), are restricted to zero values only. For this reason we call Algorithm 3 a Gauss–Jordan analogue.

It is worth noting that systems like (3.4) are useful mainly for the testing application. For experimental analysis of the complexity these systems are not suitable because, due to their triviality, they do not seem to appear in practical applications of linear Diophantine analysis.

3.2 Algorithm 2: An analogue of the Gauss elimination

Let n and m fixed and $0 < n < m$. Algorithm 4 shows formal details of our second algorithm.

This algorithm constructs homANLDE system (1.2) with the following Hilbert basis:

$$\mathcal{H} = \text{set of all columns of matrix } (F \mid \mathbb{I})^{\mathsf{T}}, \tag{3.5}$$

where $F \in \mathbb{Z}_+^{(m-n)\times n}$ and $\mathbb{I}$ is the identity $(m-n)\times(m-n)$ matrix. Note that F is unknown a priori and has to be computed by the algorithm. This is an important difference comparing with Algorithm 3.

The matrix of left-hand side coefficients (1.2) is $\mathrm{E} = (\mathbb{I} \mid B)$, where $\mathbb{I}$ is the identity $n\times n$ matrix and $B \in \{0,1\}^{n\times(m-n)}$ satisfies condition (3.2). The matrix of right-hand side coefficients (1.2) is $\mathrm{A} = (D \mid B + \Delta)$, where $D \in \mathbb{Z}_+^{n\times n}$ is defined as

$$D_{kj} = \begin{cases} 0\,, & \text{if } k \ge j \\ d_{kj} \in \mathbb{Z}_+, & \text{if } k < j+1 \\ d_{kj} \in \mathbb{N}\,, & \text{if } k = j+1 \end{cases} \tag{3.6}$$

and the matrix $\Delta \in \mathbb{Z}_+^{n\times(m-n)}$ satisfies the condition:

$$\Delta_{ni} > 0 \qquad \forall\, i = 1, 2, \ldots, m-n\,. \tag{3.7}$$

As result, the following homANLDE system is generated:

$$(\mathbb{I} \mid B)x = (D \mid B + \Delta)x \tag{3.8}$$

Algorithm 4 A Gauss analogue for generating a homANLDE system and its Hilbert basis

Require: n is the number of equations, m is the number of unknowns, $0 < n < m$.

Ensure: $(\mathrm{E}, \mathrm{A}, \mathcal{H})$.

generate randomly matrix $D \in \mathbb{Z}_+^{n\times n}$ that satisfies (3.6);

generate randomly matrix $B \in \{0,1\}^{n\times(m-n)}$ that satisfies (3.2);

generate randomly matrix $\Delta \in \mathbb{Z}^{n\times(m-n)}$ that satisfies (3.7);

$\mathrm{E} := (\mathbb{I}^{n\times n} \mid B)$; $\quad \mathrm{A} := (D \mid B + \Delta)$;

compute matrix $F \in \mathbb{Z}_+^{(m-n)\times n}$ by Gauss-similar back-substituting;

$\mathcal{H} :=$ set of all columns of matrix $(F \mid \mathbb{I}^{(m-n)\times(m-n)})^{\mathsf{T}}$;

Applying Theorem 1 one obtains that (3.8) is reduced to form (2.1):

$$x_n = \sum_{i=1}^{m-n} \Delta_{ni} x_{n+i}\,.$$

According to the facts presented in Section 1.3 (see Eq. (1.4)), this equation has Hilbert basis

$$\begin{pmatrix} x_n \\ x_{n+1} \\ x_{n+2} \\ \vdots \\ x_m \end{pmatrix} : \quad \left\{ \begin{pmatrix} \Delta_{n1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \Delta_{n2} \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} \Delta_{n,m-n} \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \right\}.$$

This makes condition (3.7) more understandable: it provides a way to manage values of Hilbert basis.

One can start from the Hilbert basis of the reduced equation, then by back-substituting sequentially compute the Hilbert basis of (3.8), see Example 6. This is close to back-substituting in the Gauss elimination method for searching a basis of a linear system. Unknowns $x_1, \ldots, x_n$ are leading variables and $x_{n+1}, \ldots, x_m$ are free ones. This is a reason to call Algorithm 4 a Gauss analogue.

Example 6. **Input:** $n = 3$, $m = 5$.

Step 1: $D = \begin{pmatrix} 0 & 2 & 3 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}$. **Step 2:** $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$. **Step 3:** $\Delta = \begin{pmatrix} 0 & 0 \\ 4 & 2 \\ 3 & 4 \end{pmatrix}$.

Step 4: $\mathrm{E} = (\mathbb{I} \mid B) = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$, $\mathrm{A} = (D \mid B + \Delta) = \begin{pmatrix} 0 & 2 & 3 & 1 & 0 \\ 0 & 0 & 5 & 4 & 3 \\ 0 & 0 & 0 & 3 & 4 \end{pmatrix}$.

Thus, the following homANLDE is generated (the corresponding CF-grammar is also shown):

$$\begin{cases} x_1 + x_4 &= 2x_2 + 3x_3 + x_4 \\ x_2 + x_5 &= 5x_3 + 4x_4 + 3x_5 \\ x_3 &= 3x_4 + 4x_5 \end{cases} \qquad \begin{aligned} r_1 &: A \to \varepsilon \\ r_2 &: B \to A^2 \\ r_3 &: C \to A^3B^5 \\ r_4 &: A \to AB^4C^3 \\ r_5 &: B \to B^3C^4 \end{aligned}$$

Step 5: Using back-substituting, we compute F:

$$\begin{aligned} x_3 &= 3x_4 + 4x_5 & &\rightsquigarrow F_{\cdot 3} = (3,4)^{\mathsf{T}} \\ x_2 &= 5x_3 + 4x_4 + 2x_5 & &\rightsquigarrow F_{\cdot 2} = (19,22)^{\mathsf{T}} \\ x_1 &= 2x_2 + 3x_3 & &\rightsquigarrow F_{\cdot 1} = (47,56)^{\mathsf{T}} \end{aligned}, \quad \text{i.e., } F = \begin{pmatrix} 47 & 19 & 3 \\ 52 & 20 & 4 \end{pmatrix}.$$

$$\textbf{Step 6:}\ (F \mid \mathbb{I})^{\mathsf{T}} = \begin{pmatrix} 47 & 56 \\ 19 & 20 \\ 3 & 4 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \rightsquigarrow \quad \mathcal{H} = \left\{ \begin{pmatrix} 47 \\ 19 \\ 3 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 56 \\ 20 \\ 4 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

Despite their simplicity, homANLDE systems (3.8) are applicable both for testing and experimental analysis of the complexity (see Section 4). These systems are associated with CF-grammars that have certain interesting properties, but for brevity this topic is out of the scope of the paper scope and we are confined to the example.

4 Testing and experimental analysis of the syntactic solver

The algorithms, described in the previous section can be applied to the following two problems: 1) testing homANLDE solvers, i.e., implementations of solving algorithms, and 2) experimental analysis of the computational complexity and related properties of such solvers.

4.1 The implemented software

We developed and implemented a special-target software system to provide automated testing and experimental analysis of homANLDE solvers. The software includes 1) implementations of the proposed algorithms for generation homANLDE systems, and 2) support to test solver outcomes for correctness and to measure solver resource consumption.

The high-level architecture of the software system is shown in Figure 2. The main module manages the whole execution process, which is iterative. At each iteration the main module calls the homANLDE generation module to generate a test/sample homANLDE system. Then the solver and the analysis module start in parallel. The former takes the generated homANLDE system as input. The analysis module inspects on a regular basis the solver execution to collect statistics (memory and time consumption). The testing module is activated whenever the solver has finished its execution, and tests the solution for correctness using the solver outcome and Hilbert basis, the latter was produced by the homANLDE generation module. The main module saves a record on solving the homANLDE system and moves to the next iteration.

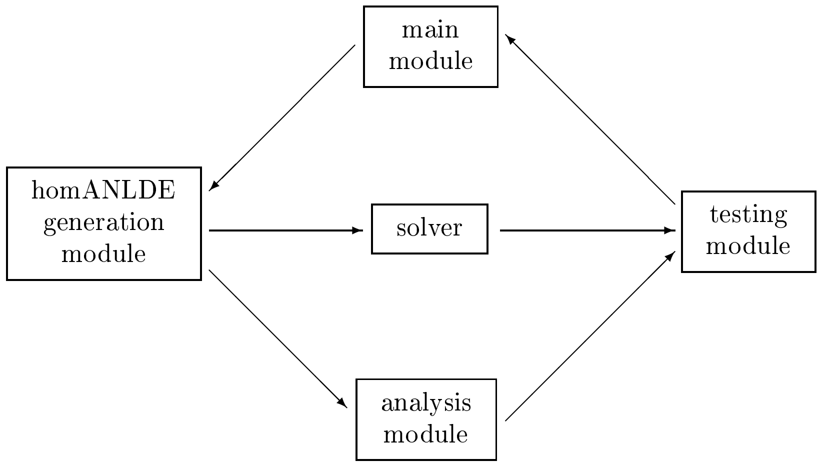


Figure 2: *The high-level architecture of the software system*

The number of iterations depends on the goals of testing or experiments. Exhaustive testing needs a lot of different test homANLDE systems to be solved and these systems should cover the whole set of homANLDE systems. Unfortunately, not all homANLDE systems can be generated by Algorithms 3 and 4 (see Figure 1 on p. 267). We design and implement two other algorithms that are more adequate in this case, see [12, 13].

The needs of experimental analysis require sample homANLDE systems from particular subclasses of homANLDE systems. For these subclasses the software system allows to estimate complexity metrics of the target solver, e.g. average time to solve a system of this subclass or percent of systems in this subclass that are solvable in space less than 1 Mb.

In addition, the software system also targets the comparison problem of different solvers. This feature is useful both for testing and analysis. For testing purposes this allows to compare solutions produced independently by different solvers; in this case test homANLDE systems can be generated regardless of the Hilbert basis. Analysis of experimental complexity of different solvers gives grounded comparison of them.

Characteristics of the implementation are presented in Table 1. The implementation is portable to Linux and Windows platforms. Computing LinesOfCode we did not count comments and empty lines.

Table 1: *Size of the software implementation*

Module	Language	LinesOfCode
Main	C++	739
homANLDE generation		
Algorithm 1	C++	55
Algorithm 2	C++	171
Algorithm 3	C++	321
Analysis	C++	145
Testing	Perl	63

4.2 Testing and experimentation experience

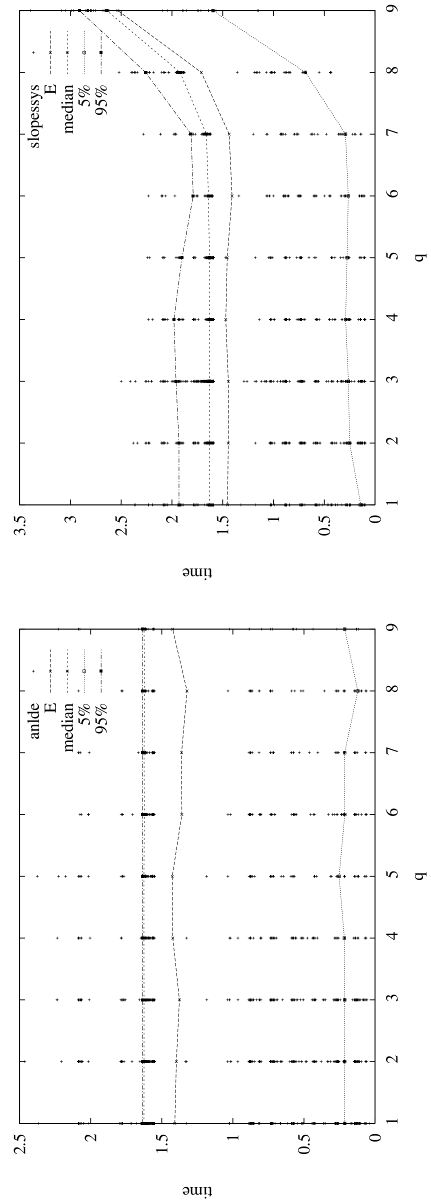
We used the software system developed to test and analyze an implementation of the syntactic algorithm—the syntactic solver, proposed in [8]. Another application of this software system was experimental comparison of the syntactic solver and the slopes solver. The latter was developed by M. Filgueiras and A. P. Tomás [11]; it can solve homNLDE systems, not only homANLDE ones.

Over 1.5 million homANLDE systems were generated for testing the syntactic solver. Both Algorithm 3 and Algorithm 4 were used (50:50). This was a stress testing that tried to find incorrectly solvable systems. The range of the generated systems was from small dimensions $n, m, \mathrm{A} \sim 10$ to large $n, m \sim 10^3$ and $\mathrm{A} \sim 10^4$. All test systems were successfully solved. This validated the correctness of the syntactic solver.

For the experimental comparison of the syntactic and slopes solvers 10000 sample homANLDE systems were generated, mainly (≈ 9000) by Algorithm 4. All of them had small dimensions ($n, m < 20$, $\mathrm{A} < 100$) due to the slopes solver's limitations. Even in this case some of the samples (≈ 500) the slopes solver could not solve, unlike the syntactic solver.

Taking into account only samples solved by both solvers (≈ 9500) the comparison results are shown in Figure 3 as experimental dependence of solving time on the number of solutions in the Hilbert basis q: Figure 3(a) is the dependence plot for the syntactic solver and Figure 3(b) is the same but for the slopes solver.

Note that q is an important implicit parameter of a homNLDE system. The growth behavior of the time with increasing q characterizes applicability of the solver for large-dimensional problems.



(a) Performance of the syntactic solver

(b) Performance of the slopes solver

Figure 3: *Experimental plot for dependency of solving time on the size of Hilbert basis*

For range $1 \leq q \leq 9$ this behavior for the syntactic solver is close to a constant, i.e., no growth at all. While for the slopes solver one can see rapid (similar to exponential) growth starting from $q = 7$.

Summing up the experiments we can conclude that the anlde solver is able to efficiently solve homANLDE systems even for large dimensions. The efficiency of the slopes solver is comparable with the syntactic one only for very small dimensions.

5 Conclusion

In this paper we addressed the problem of generation homANLDE systems. This is more complex than just random generation of integer matrices. There is a need to generate both: a homANLDE system and its Hilbert basis, and this problem meets with the similar computational complexity difficulties as the problem of solving homANLDE systems.

We introduced Theorem 1, which can be used as a general base for construction and proof of various homANLDE generators. As an example we describe two such algorithms. Recently we finish research in two other algorithms, which allow generating homANLDE systems with more interesting properties and have more features to manage characteristics of generated systems, see [12, 13] for details.

The proposed algorithms have important applications to automated and statistical testing of homANLDE solvers as well as to experimental analysis of solvers complexity. This also includes experimental comparison of different solvers. We developed and implemented the special-target software system for these applications.

By using this software system we performed stress testing of the syntactic homANLDE solver, which is an implementation of one algorithm from [8]. Our results show the correctness of the solver.

We also compared the complexity of the syntactic solver and its alternative—the slopes solver by M. Filgueiras and A. P. Tomás [11]. Our results show that in terms of time consumption, both solvers are comparable for small homANLDE systems while the syntactic solver is significantly superior for large-dimensional systems and, thus, is more applicable for practical needs.

Bibliography

[1] F. Giles, W. Pulleyblank. Total dual integrality and integer polyedra. *Linear algebra and its applications.* No. 25, 1979. pp. 191–196.

[2] Schrijver A., *Theory of Linear and Integer Programming.* John Willey & Sons Ltd., 1986.

[3] Huet G. *An algorithm to generate the basis of solutions to homogenous linear Diophantine equations.* Information Processing Letters. Vol. 3. No. 7. 1978. pp. 144–147.

[4] Domenjoud E., Tomás A. *From Elliott-MacMahon to an algorithm for general constraints on naturals.* In U. Montanari, F. Rossi (eds.), Principles and Practice of Constraint Programming — CP'95. Lecture Notes in Computer Science 976. Springer-Verlag, 1995. pp. 18–35.

[5] Clausen M., Fortenbacher A. *Efficient Solution of Linear Diophantine Equations.* Journal of Symbolic Computation. Vol. 8. No. 1&2. 1989. pp. 201–216.

[6] Contejean E., Devie H. *An efficient algorithm for solving systems of Diophantine equations.* Information and Computation. Vol. 113. No. 1. 1994. pp. 143–172.

[7] Pottier L. *Minimal solutions of linear Diophantine systems: bounds and algorithms.* Proceedings of the 4th International Conference on Rewriting Techniques and Applications (RTA'91). Como, Italy. 1991. pp. 162–173.

[8] D. G. Korzun. Sintaksicheskie algoritmy resheniya neotricatelnih lineynyh diophantovih uravneniy i ih prilozhenie k modelirovaniyu struktury nagruzki kanala Internet. (Eng. title: Syntactic algorithms for solving non-negative linear Diophantine equations and their application to modelling a structure of Internet link workload). PhD Thesis, Department of Computer Science, Petrozavodsk State University, 2002. 185 p.

[9] D. G. Korzun. Grammar-Based Algorithms for Solving Certain Classes of Linear Diophantine Systems in Non-negative Integers. Proceedings of Finnish Data Processing Week at the Petrozavodsk State University (FDPW'2000), vol. 3, pp. 52–67.

[10] M. Filgueiras, A. P. Tomás, *Solving Linear Constraints on Finite Domains through Parsing.* Proc. of EPTA'91, 1991. pp. 1–16.

[11] A. P. Tomás, M. Filgueiras. Algorithms for Solving Diophantine Equations in Naturals. DCC-FC & LIACC, Universidade do Porto, 2001. `http://www.ncc.up.pt/~apt/dioph/`

[12] K. A. Kulakov. Testirovanie i experimentalny analiz algoritmov resheniya neotricatelnyh lineyniy diophantovyh uravneniy. (Eng. title.: Testing and experimental analysis of algorithms for solving non-negative linear Diophantine equations). BSc. Thesis, Department of Computer Science, Petrozavodsk State University, 2003. 47 p. (in Russian).

[13] K. A. Kulakov. Algoritmy generacii i resheniya dlya odnorodnyh associirovannyh s KS-grammatikami sistem neotricatelnyh lineynyh diophantovyh uravneniy. (Eng. title.: Algorithms for testing and solving homogenous, associated with CF-grammars, systems of non-negative linear Diophantine equations). MSc. Thesis, Department of Computer Science, Petrozavodsk State University, 2005, to appear. (in Russian).

[14] D. G. Korzun. Reshenie odnogo klassa lineynyh diophantovyh uravneniy v neotricatelnyh celyh chislah metodami teorii formalnyh yazykov. (Eng. title: Solution of One Class of Linear Diophantine Equations in Nonnegative Integers through Parsing). Transactions of the University of Petrozavodsk on "Applied mathematics and computer science". Vol. 7. Petrozavodsk, 1999, pp. 93–116. (in Russian)

TCP Profile for Next Generation Wireless Networks

Andrei V. Gurtov†, Dr. Yury A. Bogoyavlenskiy‡

†Department of Computer Science, University of Helsinki,
†,‡Department of Computer Science, Petrozavodsk State University

†P.O. Box 68, FIN-00014 University of Helsinki, Finland
‡Lenin St., 33, Petrozavodsk, Republic of Karelia, 185910, Russia

E-mail: gurtov@cs.helsinki.fi, ybgv@cs.karelia.ru

Abstract

TCP remains the dominant transport protocol in the Internet providing reliable end-to-end data delivery. Performance problems of TCP in wireless networks are well known. In this paper, we describe the link characteristics of 2.5G and 3G wireless networks. Furthermore, a profile of TCP options is presented that in most cases provides good TCP performance in 2.5G and 3G wireless networks. In particular, the effect of the TCP timestamp option is considered.

Contents