AUTOMATA–BASED METHOD FOR SOLVING SYSTEMS OF LINEAR CONSTRAINTS IN {0,1}

Sergey Krivoi, Lyudmila Matvyeyeva, Wioletta Grzywacz

Abstract: We consider a finite state automata based method of solving a system of linear Diophantine equations with coefficients from the set {-1,0,1} and solutions in {0,1}.

Keywords: system of the linear Diophantine equations, set of the basis solutions, outputless finite state automaton.

ACM Classification Keywords: F.1.1 Models of Computation - finite automata, systems of linear Diophantine constraints

Introduction

Algorithms for computing the basis of solutions of a system of homogenous linear Diophantine equations (SHLDE) in the set of natural numbers can be applied to numerous problems. These applications include the Petri nets (determining invariants, traps and deadlocks), proving contraction of a set of disjuncts, Presburgers arithmetic, logical deductions in the predicate logic, etc. The special case of SHLDE is one with solutions over the set {0,1} and coefficients coming from the set {-1, 0, 1}. In this work we present an algorithm for computing the basis of all solutions of a SHLDE of this type. We'll apply this algorithm to a general form of a SHLDE and to a system of non-homogenous inequalities and equations as well.

1. Necessary Knowledge

The system in the following format is called SHLDE:

$$S = \begin{cases} a_{11}x_1 & + & \dots & + & a_{1q}x_q & = & 0 \\ a_{21}x_1 & + & \dots & + & a_{21}x_q & = & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{p1}x_1 & + & \dots & + & a_{pq}x_q & = & 0 \end{cases} \qquad (1)$$

where $a_{ij} \in \{-1,0,1\}, x_j \in \{0,1\}, i \in [1,p], j \in [1,q]$. Vectors $e_1 = (1,0,\dots,0,0), e_2 = (0,1,\dots,0,0),$ $\dots, e_q = (0,0,\dots,0,1)$ are the vectors of canonical basis. The vector $c = (c_1,\dots,c_q), c_i \in \{0,1\}$ is the solution of SHLDE S, when following identity holds:

$$\forall i \in [1,p] \;\; a_{i1}c_1 + a_{i2}c_2 + \dots + a_{iq}c_q \equiv 0,$$

where sign $\equiv$ denotes relation of identity. Comparison of two vectors $x = (x_1,\dots,x_q)$ and $y = (y_1,\dots,y_q)$ from set N^q can be defined as follows:

$$x \le y \Leftrightarrow \forall i \in [1,q] \; x_i \le y_i .$$

This relation is a partial order relation and the minimal vectors of the solution of the SHLDE, w.r.t. this order, are basis solution of a SHLDE. We present an algorithm of building this basis based on the theory of the finite automata.

2. Algorithm Description

Presented algorithm is based on the outputless finite automaton theory. To describe this algorithm first we consider the case of a single Diophantine equation and next the case of the system of equations.

2.1. Case of a Single Equation

Consider the equation:

$$a_1x_1 + a_2x_2 + \ldots + a_qx_q = 0 \tag{2}$$

where $a_j \in \{-1,0,1\}, x_j \in \{0,1\}, j \in [1,q]$. Using the TSS algorithm for computing the set of the solutions of the equation (2) the set of basis solutions of this equation is obtained. This follows from the theorem:

Theorem 1 If an equation $a_1x_1 + a_2x_2 + \ldots + a_qx_q = 0$ is given, where $a_j \in \{-1,0,1\}, j \in [1,q]$ then the TSS of this equation consists of the set of basis solutions of this equation.

Proof: Let $c = (c_1,\ldots,c_q), c_i \in \{0,1\}$ be any solution of equation (2), and its non-negative coordinates $c_{j1},\ldots,c_{jq}$ be given. Lets exclude from this set the set of non – zero coordinates, all of which correspond to zero – valued coefficients of the equation (2). It means that we obtain the vector $c' = c - e_{j1} - \ldots - e_{jr}$, where $e_{ji}(i \in [1,q])$ are vectors of canonical basis, which are also the elements of the TSS. Obviously the vector c' is the solution of the equation (2) as well. If $c' = 0$ then the theorem is proved, and $c = e_{j1} + \ldots + e_{jr}$. If $c' \neq 0$, then the number of non–zero coordinates is even and moreover one half of these coordinates corresponds to the coefficients equal to –1 in the equation (2), and the second half — equal to +1. Since the TSS contains all vectors corresponding to the combinations of the coefficients equal to -1 and 1 then indeed c' can be presented in the form of non–negative linear combination presented below:

$$c' = e_{i1} + \ldots + e_{it},$$

where $e_{ij} \in TSS$ of the equation (2). Since $c' = c - e_{j1} - \ldots - e_{jr}$ then $c = e_{j1} + \ldots + e_{jr} + e_{i1} + \ldots + e_{it}$.

Theorem is proved.

Corollary:

a) Number of the basis solutions of the equation (2) equals $k * m + r$, where k, m - are positive and negative coefficients of this equation respectively.
b) Computing the set of the basis solution of the equation (2) is done in O(q^2) time.

Example 1. Let the following equation be given:

$$1x_1 + 0x_2 - 1x_3 - 1x_4 + 1x_5 + 1x_6 - 1x_7 = 0.$$

The TSS of above equation consists of the set of the basis solution in the form of following vectors:

$$s_1 = (0,1,0,0,0,0,0), s_2 = (1,0,1,0,0,0,0), s_3 = (1,0,0,1,0,0,0), s_4 = (1,0,0,0,0,0,1),$$

$$s_5 = (0,0,1,0,1,0,0), s_6 = (0,0,1,0,0,1,0), s_7 = (0,0,0,1,1,0,0), s_8 = (0,0,0,1,0,1,0),$$

$$s_9 = (0,0,0,0,1,0,1), s_{10} = (0,0,0,0,0,1,1).$$

Number of zero coefficients of this equation equals 10, because $r = 1, k = 3, m = 3$, and we get $k * m + r = 3 * 3 + 1 = 10$ being resulted from the corollary.

2.2. Automata-based Presentation of a Basis Solution

Let $A = (A, X, f, \{a_0\}, F)$ be the outputless finite automaton, where A - a finite set of states, X - a finite set of input symbols, $f : A \times X \to A$— transition function, $a_0 \in A$ - starting state, $F \subseteq A$- a set of the terminal states. Presentation of the set of basis solutions of the equation (2) by means of the automata has the following form. Set of states of the automaton $A = \{0,1,-1\}, X = \{e_1, e_2, \dots e_q\}$, where e_i $(i \in [1, q])$, are the vectors of the canonical basis from N^q, the starting state and terminal state are the state 0, and transition function f is defined as follows:

$f(0, e_i) = 0$ if e_i corresponds to coefficient 0;

$f(0, e_i) = 1$ if e_i corresponds to coefficient 1;

$f(0, e_i) = -1$ if e_i corresponds to coefficient -1;

$f(1, e_i) = 0$ if e_i corresponds to coefficient -1;

$f(-1, e_i) = 0$ if e_i corresponds to coefficient 1;

in other cases function f is unspecified.

Example 2. For the equation from the example 1 we have:

$$A = \{0,1,-1\}, X = \{e_1, e_2, \dots, e_7\}, a_0 = 0, F = \{a_0\},$$

and transition function f is defined in the form of transition table:

F	0	1	-1
e_1	1	-	0
e_2	0	-	-
e_3	-1	0	-
e_4	-1	0	-
e_5	1	-	0
e_6	1	-	0
e_7	-1	0	-

The vector of the solutions is built per the word p, which moves the automaton from the state 0 to the state 0 in the following way. If $p = e_{i1}e_{i2}\dots e_{ik}$ then vector of solutions $x_p = e_{i1} + e_{i2} + \dots + e_{ik}$, where $e_{ij} \in X$, $j \in [1, k]$. It is clear that we are interested in words corresponding to the cycles in automaton, from starting state 0 to terminal state 0. For example, in the above automaton the words: $p_1 = e_1e_3, p_2 = e_1e_4,$
$p_3 = e_1e_7, p_4 = e_5e_3, p_5 = e_5e_4, p_6 = e_5e_7, p_7 = e_6e_3, p_8 = e_6e_4, p_9 = e_6e_7, p_{10} = e_2$
creates the set of the basis solutions composed of the vectors $s_1 - s_{10}$ (see example 1).

Theorem 2. Automaton defined in the above form presents the set of all solutions of the equation (2).

Proof follows directly from the idea of automata building. Actually all the basis vectors of the solution corresponding to the zero-coefficients are presented in the given automaton, because corresponding words are obtained by transition from the state 0 to the state 0. Then all basis solutions corresponding to combination of –1 and 1 are presented by these words as well. Since all these vectors form the set of basis solutions of (2), the theorem is proved.

2.3. Case of the System of the Equations

Let a SHLDE consisting of two equations be given. We'll present the sets of the basis solutions of every equation in the form of automaton. We'll build a product from the given automata and then we'll search all the words corresponding to the paths from the state (0,0) (this state is starting and terminal state) to the same state. Given set includes all of the basis vectors of the set of solution of the SHLDE. Let us consider the following example at first.

Example 3. Let the system of equations be given:

$$S = \begin{cases} 1x_1 + 0x_2 - 1x_3 - 1x_4 + 1x_5 + 1x_6 - 1x_7 = 0 \\ 1x_1 - 1x_2 + 0x_3 + 0x_4 - 1x_5 - 1x_6 + 1x_7 = 0 \end{cases}$$

Transition table of automaton corresponding to the first equation this SHLDE is described above (see example 1), and transition table corresponding to the second equation has the following form:

f_1	0	1	-1
e_1	1	-	0
e_2	-1	0	-
e_3	0	-	-
e_4	0	-	-
e_5	-1	0	-
e_6	-1	0	-
e_7	1	-	0

Transition table of the product of automata obtained before, after excluding unreachable and deadlock states has the following form:

f_1	(0,0)	(0,-1)	(-1,0)	(1,-1)	(-1,1)	(1,0)	(0,1)
e_1	-	(1,0)	(0,1)	-	-	-	-
e_2	(0,1)	-	-	-	-	(0,0)	(0,0)
e_3	(-1,0)	-	-	-	-	-	-
e_4	(-1,0)	-	-	-	-	-	-
e_5	(1,-1)	-	(0,-1)	-	(0,0)	(1,0)	(1,0)
e_6	(1,-1)	-	(0,-1)	-	(0,0)	(1,0)	(1,0)
e_7	(-1,1)	(-1,0)	-	(0,0)	-	-	-

Starting and terminal state in this automaton is the state (0,0). The search of cycles in obtained automata results in the following words:

$p_1 = e_2e_1e_3$, $p_2 = e_2e_1e_4$, $p_3 = e_3e_1e_5e_4$, $p_4 = e_3e_1e_6e_4$, $p_5 = e_5e_7$,

$p_6 = e_6e_7$, $p_7 = e_7e_6$, $p_8 = e_2e_7e_1e_5e_3$, $p_9 = e_2e_7e_1e_6e_3$ etc. These words correspond to the following set of the basis solutions:

$$s_1 = (1,1,1,0,0,0,0), s_2 = (1,1,0,1,0,0,0), s_3 = (1,0,1,1,1,0,0),$$

$$s_4 = (1,0,1,1,0,1,0), s_5 = (0,0,0,0,1,0,1), s_6 = (0,0,0,0,0,1,1).$$

The analysis of the example presented above results in the following conclusion:

- the size of the automaton corresponding to the SHLDE is not bigger than $3 * q$;

- the memory required for representing original automaton is proportional to $3 * p * q$;
- the length of the longest cycle in product of initial automata is not bigger than q .

Theorem 3. The set of solutions of a SHLDE, corresponding to words representing simple cycles in an automaton being a direct product of the original automata, contains all the basis solutions of the given SHLDE.

Proof: Let us assume that the SHLDE consists of two equations. Assuming that x is a solution of SHLDE, then the vector x is equivalent to the word p in the automaton representing the first equation. The same vector x represents word p in the automaton representing the second equation. This is the result of the assumption that the original automata represent the set of all the possible solutions of the SHLDE. Under this assumption the word p , belonging to the common part of the two regular languages represented as a product of the original automata, should take the automaton from a starting state to a starting state (being at the same time a terminal state of this automaton). This means that word p corresponds to some cycle in the automaton. We'll represent this word as a concatenation of its subwords $p_1, p_2, \dots p_k$, representing cycles. As words $p_1, p_2, \dots p_k$ generate the basis solutions we shall call them $e_1, e_2, \dots, e_k$. Then it becomes obvious that $x = e_1 + e_2 + \dots + e_k$.

3. Time Characteristics of the Algorithm

From the general automata theory we know that the number of states of an automaton that represents a set of all solutions of a SHLDE having p equations does not exceed 3^p . This estimation shows us that the time required to build such an automaton, to find all the cycles and distinguish the basis solutions will be proportional to 3^p , which is not very optimistic value. Obtained estimation does not contradict the result saying that the complexity of the algorithm for the basis of the set of SHLDE solutions creation will be exponential. In most cases the number of states of the resulting automaton is significantly smaller than 3^p . Obviously the redundancy of the calculations in the presented algorithm comes from the fact that some of the calculations are made multiple times. For example the result (0,0,0,0,1,0,1) is obtained twice (the same as the other results) and the automaton, which transition table is presented below, presents the same set of solutions as the automaton in the example 3, but it has lesser number of states.

f_3	(0,0)	(0,-1)	(-1,0)	(-1,1)	(1,0)
e_1	-	(1,0)	-	-	-
e_2	(0,-1)	-	-	-	-
e_3	(-1,0)	-	-	-	-
e_4	(-1,0)	-	-	-	-
e_5	-	-	(0,-1)	(0,0)	(1,0)
e_6	-	-	(0,-1)	(0,0)	(1,0)
e_7	(-1,1)	-	-	-	-

It should be mentioned as well that because the solutions of a SHLDE belong to a set {1,0} all the vectors in the algorithm are Boolean which simplifies the calculations.

The biggest advantage of this algorithm is that the computation of the basis of the set of all the solutions does not require the explicit form of the automaton being the product of the original automata to be built, instead we create the basis directly using the original automata by going in parallel through the corresponding states of those automata. This allows for a strong memory usage optimization and as a result to obtain greater calculations efficiency. For example the creation of the explicit form of the automaton in the example 3 is not necessary.

The basis can be built using only the original automata which present the sets of solutions of the first and the second equation respectively.

Finally, as it has been shown above, the presented algorithm allows for further optimization. This possibility comes from the fact that during the creation of the basis of the set of solutions for the automaton representing the complete set of solutions many results are encountered multiple times. This redundancy comes from the fact that the concatenation in this case is associative. This presents another problem: *how to minimize finite automaton taking into account the associativity of the concatenation of the words in the input alphabet of the automaton*.

4. Second Approach to the Creation of an Automaton which Represents the Basis of the Set of Solutions

We'll describe another method of the creation of an automaton representing the basis of the set of all solutions of SHLDE, which does not require computing the product of the automata.

Let $X = \{s_1, s_2, \ldots s_n\}$ - the basis of the set of solutions of the first equation SHLDE S and $L_2(x) = 0$ - the second equation in the system S. Using the set S and function $L_2(x)$ we'll build a finite automaton $A = (A, X, \{0\}, f, \{0\})$ in which the state 0 is starting and terminal state and the transition function is as follows:

$$f(a, s_i) = \begin{cases} L_2(s_i), & if \quad a = 0; \\ a + L_2(s_i), & if \quad a \neq 0 \,\&\, |a + L_2(s_i)| < |a|, \\ unspecified & int\, heothercase \end{cases}$$

After creating the automaton in this way we compute the basis of the set of solutions of the subsystem $L_1(x) = 0 \,\&\, L_2(x) = 0$ of the system S. Elements of this set play a part of the new alphabet. This alphabet and the function $L_3(x)$ (if exists), being on the left-hand side of the equation $L_3(x) = 0$, is used to build the next automaton and so on, until we have created the basis of the set of all solutions of the SHLDE S.

Lets consider the SHLDE S from the example 3. The results of the function $L_2(x)$ for the vectors from the set $X = \{s_1, s_2, \ldots, s_{10}\}$ are the values: -1,1,1,2,-1,-1,-1,-1,0,0. The automaton A representing the alphabet X and function $L_2(x)$ is presented below:

f	0	-1	1	2
s_1	-1	-	0	1
s_2	1	0	-	-
s_3	1	0	-	-
s_4	2	-	-	-
s_5	-1	-	0	1
s_6	-1	-	0	1
s_7	-1	-	0	1
s_8	-1	-	0	1
s_9	0	-	-	-
s_{10}	0	-	-	-

We create the vector-solutions of SHLDE. These vectors correspond to the simple cycles from the state 0 into the same state 0. In our case these vectors are the following:

$s_1 + s_2 = (1,1,1,0,0,0,0), s_1 + s_3 = (1,1,0,1,0,0,0), s_5 + s_2 = (1,0,2,0,1,0,0), s_6 + s_2 = (1,0,2,0,0,1,0),$

$s_6+s_3=(1,0,1,1,0,1,0), s_7+s_2=(1,0,1,1,1,0,0), s_7+s_3=(1,0,0,2,1,0,0), s_8+s_2=(1,0,1,1,0,1,0),$

$s_8+s_3=(1,0,0,2,0,1,0), s_4+s_1+s_5=(1,1,1,0,1,0,1), s_4+s_1+s_6=(1,1,1,0,0,1,1), s_4+s_1+s_7=(1,1,0,1,1,0,1),$

$s_4+s_1+s_8=(1,1,0,1,0,1,1), s_4+s_5+s_6=(1,0,2,0,1,1,1), s_4+s_5+s_7=(1,0,1,1,2,0,1),$

$s_4+s_5+s_8=(1,0,1,1,1,1,1), s_4+s_6+s_7=(1,0,1,1,1,1,1), s_4+s_6+s_8=(1,0,1,1,0,2,1),$

$s_4+s_7+s_8=(1,0,0,2,1,1,1), s_9=(0,0,0,0,1,0,1), s_{10}=(0,0,0,0,0,1,1).$

The basis vectors are:

$s_1=(1,1,1,0,0,0,0), s_2=(1,1,0,1,0,0,0), s_3=(1,0,1,1,1,0,0),$

$s_4=(1,0,1,1,0,1,0), s_5=(0,0,0,0,1,0,1), s_6=(0,0,0,0,0,1,1)$,

which is confirmed by the solution of this SHLDE obtained by the previous method.

5. Summary

Presented algorithms can be applied to a general form SHLDE, systems of non-homogenous equations and systems of linear inequalities as well. In these cases we introduce one (for the system of non-homogenous equations) or more (for the system of inequalities) additional variables thus reducing them to a problem of solving a SHLDE in the set {0,1}.

Given algorithms are especially effective for determination of the incompatibility of SHLDE and for computing the first solution of SHLDE. Obviously, when the SHLDE is incompatible, then the product of automata or an automaton created according the second algorithm, which represents the basis of the set of solutions of the given SHLDE, will be empty. From the method of creation of these automata we conclude that they will be created fast enough. If a first cycle appears in the process of building of such automata, it means that the SHLDE is compatible and, if necessary, the solution of SHLDE, corresponding to this cycle, can be compute and further computing can be canceled accordingly.

It should be mentioned that in the second approach the same problem of minimization of a finite automaton arises taking into account the associativity of the concatenation of the words in the input alphabet X of the automaton.

Bibliography

[Murata, 1989] T. Murata. Petri Nets: Properties, Analysis and Applications. In Proceedings of the IEEE, 1989, Vol. 77, No:4, pp. 541-580.

[Baader, 1994] Baader F., Ziekmann J. Unifications Theory. In Handbook of Logic in Artificial Intelligence and Logic Programming. – Oxford University Press. 1994. pp. 1-85.

[Common, 1999] Common H. Constraint solving on therms: Automata techniques (Preliminary lecture notes). Intern. Summer School on Constraints in Computational Logics: Gif-sur-Yvette, France, September 5-8. 1999. 22 p

[Lloyd, 1987] Lloyd J. Foundations of Logic Programming (2-d edition). Springer Verlag. 1987. 276 p.

[Krivoi, 2003] Krivoi S. Algorithms of solving systems of linear diophantine constraints in space [0,1]. Cybernetics and system analysis. 2003, number 5. pp. 58-69.

Authors' Information

Krivoi Sergey – Glushkov Institute of Cybernetics NSA Ukraine, Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: krivoi@i.com.ua

Lyudmila Matvyeyeva – Institute of Cybernetics NAS of Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: luda@iss.org.ua

Grzywacz Wioletta – Technical University of Czestochowa, Institute of Computer and Information Sciences, Czestochowa, Poland, e-mail: wiola@icis.pcz.czest.pl