Complexity of resolution of systems of equations over partial orders

A Yu Nikitin  
Omsk State Technical University, Omsk pr. Mira 11, Russian Federation  
E-mail: nikitinlexey@gmail.com

Abstract. Many modern models of informational defence represents by graphs and partial orders (posets). It is very important to resolve such algorithmic problems as searching of elements that satisfies some conditions (or, another words, to resolve recognition problems) in this models. Algebraic geometry is resolving such problems and find elements from algebraic structure that is solution of system of equations. But before searching the effective algorithms of resolving the systems, first better to answer the essential question: can we resolve the system efficiently? In this article we find the condition to poset which can be efficiently resolve every finite system of equations over the partial orders.

1. Introduction  
Universal algebraic geometry studies system of equation over arbitrary algebraic structures such as group, field, graph or something else. Classic algebraic geometry studied fields in a greater degree. Period of classic algebraic geometry was from the end of XVIII century to the beginning of XX century. Universal algebraic geometry was developed really great in XX century. Main methodologies to study was suggested by B.I. Plotkin [4] and this is algebraic methodology. Second main methodology was suggested by three authors E.Yu. Daniyarova, A.G. Myasnikov and V.N. Remeslennikov in the series of works [5], [6], [7], [8], [9] and this is theoretical-model methodology. Speaking about second methodology in universal algebraic geometry, the was started studies in groups, lattices, semitallices and graphs [10], [11]. Graphs and partial orders became more popular at the present time. This mathematical objects have deep theoretical searches as well as wide practical usage. Then solving systems of equations over such structures rising in natural way. And it provides to study structures of algebraic sets and coordinate algebras of partial orders. But, before solving systems we should ask the essential question: is it possible effectively solve the systems over concrete partial orders at all? If it’s not then why? If it is then over what partial orders we could solve systems effectively too? This article is devoted to finding answers on this questions.

2. Preliminaries  
In this section we reminds basic definitions of partial orders theory, graph theory and universal algebraic geometry over partial orders.

Partial ordered set (partial order or poset) is algebraic system \( P = (P, \leq, A) \) where \( \leq \) is binary order relation, \( A \) is constant symbols and the structure satisfies 3 axioms:
(i) \( \forall p \in P \ p \leq p \) (reflexive relation);
(ii) \( \forall p_1, p_2 \in P \ p_1 \leq p_2 \land p_2 \leq p_1 \rightarrow p_1 = p_2 \) (antisymmetry relation);
(iii) \( \forall p_1, p_2, p_3 \in P \ p_1 \leq p_2 \land p_2 \leq p_3 \rightarrow p_1 \leq p_3 \) (transitive relation).

Language (signature) of posets with constants we notes as \( L_A \). Poset \( \mathcal{P} = \langle P \mid L_A \rangle \) where between \( P \) and \( A \) exists bijection is called diophantine. In this article we observe only diophantine posets.

For poset \( \mathcal{P} \) the elements of it \( x \) and \( y \) are comparable is \( x \leq y \) or \( y \leq x \). Otherwise, elements are incomparable.

Let the partial order \( P \) is given. For any subset \( A \) of poset’s \( P \) elements we could define sets \( A_\uparrow = \{ x \in P \mid \forall a \in A \ a \leq x \} \) and \( A_\downarrow = \{ a \in P \mid \forall a \in A \ x \leq a \} \). These sets is called upper set (upset) and lower set (down set) of \( A \) or up and down fan of \( A \). For one-element set \( A = \{ a \} \) notations are \( a_\uparrow \) and \( a_\downarrow \) correspondence.

Let \( X = \{ x_1, \ldots, x_n \} \) is a set of variables. Term of \( L_A \) language of poset \( \mathcal{P} \) from variables \( X \) is any constant from language’s constants \( A \) or any variable from \( X \). Atomic formula of language \( L_A \) of poset \( \mathcal{P} \) from variables \( X \) defines as follows:

(i) if \( t_1, t_2 \) – terms, then \( t_1 = t_2 \) – atomic formula;
(ii) if \( t_1, t_2 \) – terms, then \( t_1 \leq t_2 \) – atomic formula.

Equation over partial order \( P \) in language \( L_A \) from variables \( X \) is any atomic formula over \( \mathcal{P} \). It’s easy to see all types of equations under \( \mathcal{P} \):

(i) \( a_i = a_j \), where \( a_i, a_j \in A \),
(ii) \( a_i \leq a_j \), where \( a_i, a_j \in A \),
(iii) \( x_i = a_j \), where \( x_i \in X \), \( a_j \in A \),
(iv) \( x_i = x_j \), where \( x_i, x_j \in X \),
(v) \( a_i \leq x_j \), where \( x_j \in X \), \( a_i \in A \),
(vi) \( x_i \leq a_j \), where \( x_i \in X \), \( a_j \in A \),
(vii) \( x_i \leq x_j \), where \( x_i, x_j \in X \).

System of equations \( S(X) \) from variables \( \{ x_1, \ldots, x_n \} \) is any set of equations from variables \( X \). If \( P \) is domain of \( \mathcal{P} \) then point \( p = (p_1, \ldots, p_n) \in P^n \) is solution of system \( S(X) \) iff arter substitution point’s coordinates on the place of correspondence variables there is provide a true statements over \( \mathcal{P} \): \( \mathcal{P} \models \varphi_i(p_1, \ldots, p_n), \varphi_i(x_1, \ldots, x_n) \in S(X) \). The set of solutions of system \( S(X) \) defines in natural way. Set of solutions of system \( S(X) \) denotes as \( V(X) \). Two systems \( S_1 \) and \( S_2 \) are equivalent iff their sets of solutions are the same (denotes \( S_1 \sim S_2 \)).

Let’s reach agreement about notation of equation’s sets. Equations \( a_i = a_j \), where \( a_i, a_j \in A \) in system \( S(X_n) \) denotes as \( S_{a_i = a_j} \). The rest equation types are similar: \( S_{a_i \leq a_j}, S_{x_i = x_j}, S_{a_i \leq x_j}, S_{x_i \leq a_j} \) and \( S_{x_i \leq x_j} \) for system \( S(X) \).

Now, let’s reminds some definitions from graph theory. Graph \( G = \langle V \mid E \rangle \) is algebraic structure with set of vertices \( V \) as domain and relation of adjacency of vertices \( E \) as predicate. Pair of vertices \( u, v \in V \) is called edge of graph \( G \) if \( E(u, v) \) is true over \( G \). We also denotes edge between \( u \) and \( v \) as \( (u, v) \). If the order between vertices \( u \) and \( v \) is irrelevant the edge is called undirected or just edge. If the order between vertices is relevant then the edge is called directed or just arc. Graph is undirected if each of it edges is undirected. And graph is directed if each of it edge is directed. If edge has start and and in the same vertex then such edge is called loop.

For directed graph \( G \) denote undirected graph \( U(G) \) that was produced from \( G \) by substitution of arcs by undirected edges.

For undirected graphs there is true symmetry axiom: \( \forall u, v \in V \ u \neq v \rightarrow E(u, v) = E(v, u) \).
In the article we need some special graph defined in [2]. Let’s define set of intervals \( \{I_1, \ldots, I_n\} \) on real straight line \( \mathbb{R} \). Undirected graph \( G = (V|E) \) where \( V = \{I_1, \ldots, I_n\} \) and \( (I_i, I_j) \in E(G) \) is called interval iff \( I_i \cap I_j \neq \emptyset \). Then, define two assemblies of intervals \( \{I_1, \ldots, I_n\} \) and \( \{J_1, \ldots, J_n\} \). Directed graph \( G = (V|E) \) is called interval if for every vertex \( v \) of \( G \) corresponds pair \( (I_u, I_v) \) and \( (u, v) \in E(G) \) iff \( I_u \cap J_v \neq \emptyset \). Interval digraph \( G \) is called adjusted when for every vertex \( v \) of \( G \) intervals \( I_v \) and \( J_v \) have the same left point.

Let’s describe well known connection between graphs and posets [3]. Let’s there is given a poset \( P \). We can build graph \( H \) by \( P \) as follows: vertices of \( H \) is elements of \( P \) and arc \( E(p_i, p_j) \) is in graph iff formula \( p_j \leq p_i \) is true over the \( P \). Produced graph is reflexive, transitive, antisymmetric and acyclic(if we observe graph without loops). Let’s call graph produced from poset as \( p\)-graph.

**Homomorphism** between graphs \( G \) and \( H \) is such mapping \( \varphi : V(G) \to V(H) \) that if \( (u, v) \) is edge in \( G \) then \( (\varphi(u), \varphi(v)) \) is edge in \( H \).

3. Complexity of system of equations resolution problem

This section will prove \( NP \)-completeness of system resolution problem over the certain types of partial orders. Also there will be formulated necessary condition on partial orders for polynomial resolution of systems of equations over posets. Let’s fix partial order \( P \) in language \( L_A \). Denote the system resolution problem over poset \( P \) by \( S(P) \). The problem \( S(P) \) is formulated as follows: is the system \( S(X_n) \) consistent over poset \( P \) in language \( L_A \).

It turns out that the problem \( S(P) \) is connected with well-known list homomorphism problem or list \( H \)-coloring. By the work [1] denotes list homomorphism problem for graph \( H \) as \( L - HOM(H) \). The problem \( L - HOM(H) \) is formulated as follows: is there exists homomorphism from given graph \( G \) to fix graph \( H \) that for every vertex \( v \in V(G) \) vertex \( \varphi(v) \) is in defined list of vertices \( L(v) \subseteq V(H) \). For every vertex \( v \) is defined it’s own list \( L(v) \).

If all lists \( L(v) \) are equal to \( V(H) \) the problem \( L - HOM(H) \) is just common homomorphism graph problem \( HOM(H) \). In this article we also interested in \( RET(H) \) problem. Problem \( RET(H) \) is \( L - HOM(H) \) problem where for every vertex \( u \) input graph \( G \) list \( L(u) \) or single vertex \( v \) \( V(H) \) (and for every vertex \( u \) there is defined it’s own vertex \( v \)) or all set of vertices \( V(H) \).

The problem \( S(P) \) could be reduced to \( L - HOM(H) \) problem by polynomial time. Next we describe algorithm of reduction.

Input of \( L - HOM(H) \) problem is graph \( G \) with lists of vertices \( L \). Input for \( S(P) \) problem is system of equations \( S(X_n) \) over \( P \). It possible to divide system \( S(X_n) \) on 7 non-overlapping subsystems by the type of equations: \( S_{=a}, S_{a\leq a} \), etc. Then we should note that for every terms \( t_1 \) and \( t_2 \) and every equation \( t_1 = t_2 \) over \( P \) is equivalent to pair of equations \( t_1 \leq t_2 \) and \( t_2 \leq t_1 \). So the system \( S(X_n) \) could be replaced equivalent system of equations \( S'(X_n) \) that contains only four types of equations: \( S_{a\leq a}, S_{x\leq a}, S_{a\leq x} \) and \( S_{x\leq x} \).

Now let’s talk about coding the problem’s input. Graph is presented by adjacency matrix. Posets also can be presented as adjacency matrix because of poset is a special type of graph, and equations can be coded as triples: first argument, second argument and predicate.

First, let’s suggest that subsystem \( S_{a\leq a} \) in \( S'(X_n) \) is consistent. In that way it can exclude subsystem \( S_{a\leq a} \) from \( S'(X_n) \) because of this subsystem doesn’t have influence on consistency of full system \( S'(X_n) \). Then it can build graph \( H' \) from poset \( P \) by the method we describe in previous section. After that we build graph \( G' \) by system of equations \( S_{x\leq x} \) as follows: variables \( X_n \) in system \( S_{x\leq x} \) is vertices of \( G' \) and if \( S_{x\leq x} \) contains equation \( x_j \leq x_i \) then graph \( G' \) contains arc \( (x_i, x_j) \). Lists of vertices \( L \) of graph \( H' \) builds by subsystems \( S_{x\leq x} \) and \( S_{a\leq x} \). For every variable \( x_i \) in \( S_{x\leq x} \) separates such equations that depends only from variable \( x_i \): \( S_{x\leq x} \) and \( S_{a\leq x} \). Next, choose all constants from \( S_{x\leq x} \) and denote it as \( A_{x_i} \). Choose the constants from \( S_{a\leq x} \), and denote it as \( A_{x_i} \), the same way. Finally, define set of vertices \( L'(x_i) = A_{x_i} \cap A_{x_i}^+ \). If subsystem
$S_{a \leq a}$ is empty then $A_{a \leq a} = \emptyset = A$. To conclude, we were built graph $G'$ and lists of vertices $L'$ for $L - HOM(H')$ problem input by the system $S(X_n)$ over poset $P$.

Now, if subsystem $S_{a \leq a}$ inconsistent. It means that $S(X_n)$ inconsistent and there is exist equation $a_i \leq a_j$ that false over $P$. Lets define three equations $x_1 = a_i, x_2 = a_j$ and $x_1 \leq x_2$ by $a_i \leq a_j$ that equivalent to it. Then it’s easy to see that these three equations is inconsistent system $S''(x_1, x_2)$ that equivalent to origin system $S(X_n)$. Next reducing procedure of $S''(x_1, x_2)$ to graph goes the same way as the first case.

**Argumentation:** If system $S(X_n)$ consistent over $P$ there is exists some solution of it $p = (p_1, \ldots, p_n)$. So we build graphs $G'$ with lists of vertices $L'$ and $H'$ by system of equations $S(X_n)$ and poset $P$ correspondently. In solution $p$ for every element of poset $p_i$ finds correspondent vertex $v_i$ in graph $H'$. We produced vector of vertices $v = (v_1, \ldots, v_n)$ by this method. This vector defines such homomorphism $\varphi : V(G') \to V(H')$ that is structures as follows. Vertex $g_i$ of $G'$, that corresponds to variable $x_i$ of system $S(X_n)$, maps to vertex $v_i$ of $H'$. If an arc $(g_i, g_j)$ is in $G'$ then equation $x_i \geq x_j$ is in system $S(X_n)$. And image of this arc $(\varphi(g_i), \varphi(g_j)) = (v_i, v_j)$ also be an arc in graph $H'$ because of statement $p_i \geq p_j$ is true over $P$. Moreover, $p_i$ satisfies all equations in $S_{a \leq a} \cup S_{a \equiv a} \subseteq S(X_n)$ and it provides to $v_i \in L'(g_i)$. So if system $S(X_n)$ resolves over $P$ then there is exists homomorphism $\varphi : G' \to H'$ that satisfies all lists $L'$.

The converse is also true. Is there exist homomorphism between graphs then we can build solution of equation by it.

Reduction algorithm is polynomial by number of poset’s elements (let it be $m$) and number of system’s variables (let it be $n$). First step of algorithm is substitution of subsystems $S_{a \leq a}, S_{a = a}$ and $S_{a = a}$ by $S_{a \leq a}, S_{a \equiv a}, S_{a \leq a}$ and $S_{a \leq a}$ is increasing number of equations not more then two times. Next step of algorithm is check consistency of $S_{a \leq a}$ subsystem. There is not more then $m^2$ checks of equations. Then we builds $p$-graph $H'$ by poset $P$. This procedure is, actually, copying of adjacency matrix. Building of $G'$ graph is building of adjacency matrix with $n$ vertices and not more than $n^2$ arcs. Finally, building of lists of vertices is building of fan’s intersections. Building $a^\uparrow$ and $a^\downarrow$ for any element $a$ in poset is going on by comparing element $a$ with all elements of poset. So complexity of it’s operation is $O(m)$. Building fan $A_{a \equiv a}$ for arbitrary subset of poset works as follows. First, we build $a^\uparrow$ for some element $a \in A_{x_1}$. Next, the rest of elements in $A_{x_1}$ compares with $a^\uparrow$. This procedure measures as $O(m^2)$. And we must do $n$ this procedures. To conclude, algorithm of reduction $S(P)$ problem to $L - HOM(H)$ problem is polynomial by number of poset’s elements and number of system’s variables. It means that algorithm of reduction is Karp’s reduction.

**Figure 1.** Partial order $M$

Note that $L - HOM(H)$ problem can’t reduce to $S(P)$ problem. It means that $L - HOM(H)$ problem wider than $S(P)$. For instance, there is Hasse diagram of poset $M$ on fig. 1. If we
define list \( L(v) = \{ a_1, a_3 \} \) then it’s easy to see that there isn’t any system of equation that defines \( L(v) \) by fans intersection.

However, it is possible to reduce \( RET(H) \) problem to \( S(P) \) problem for p-graphs \( H \). Reverse reduction builds system of equations \( S(X_n) \) and poset \( P \) by graphs \( G \) and \( H \) with lists of vertices \( L(v) \). Poset \( P \) builds from p-graph \( H \) by method described above. Lets suggest that \( G \) graph contains \( n \) vertices. System of equations \( S_{x \leq x} \) builds from graph \( G \) as follows. For every vertex of \( G \) searches variables from \( X_n \). If arc \((g_i, g_j)\) is in graph \( G \) then the equation \( x_i \geq x_j \) adds to system \( S_{x \leq x} \). System of equations \( S_{x=a} \) is built by lists \( L(v) \) that way: if for vertex \( g_i \) from \( G \) list \( L(g_i) = h_j \) then we add \( x_i = a_j \) wo system. variable \( x_i \) corresponds to vertex \( g_i \) and constant \( a_j \) corresponds to vertex \( h_j \). Otherwise, if \( L(g_i) = V(H) \) then there is nothing to add to the system. Finally, built system will be \( S_{x \leq x} \cup S_{x=a} \).

As in reduction \( S(P) \) to \( L – HOM(H) \) problem it can be observed correspondence homomorphisms between \( G \) with lists \( L \) and \( H \) and solutions of system \( S(X_n) \) over \( P \).The reduction is polynomial(by number of vertices and edges of \( G \) and \( H \)) because of building \( P \) from graph \( H \) is also copying of adjacency matrix. Building system \( S_{x=a} \) measures by number of arcs in graph \( G \) and building system \( S_{x \leq x} \) measures by number of vertices of \( G \).

Now we show that for certain partial orders problem of resolve system of equations over poset is \( NP \)-complete. For this purpose we refer on theorem from [1].

**Theorem 1** If \( H \) is a reflexive digraph such that \( U(H) \) is a cycle with at least four vertices, then the retraction problem \( RET(H) \) is \( NP \)-complete.

Now we formulate the result that bases on this theorem and proved reduction.

**Theorem 2** Lets fix finite partial order \( P \). Is correspondence p-graph \( H \) in \( U(H) \) contains cycle with at least four vertices without chords, then \( S(P) \) problem in \( NP \)-complete.

Such posets are exists. One of those is depicted on fig. 2.

![Figure 2](image)

**Figure 2.** Poset \( P \) that has graph \( U(H) \) with cycle length 4 without chords.

Next we formulate the result that characterize polynomial resolution of \( S(P) \) problem. Use for it criterion of polynomial resolution of \( L – HOM(H) \) problem

**Theorem 3** ([1]) Let \( H \) is reflexive digraph. If \( H \) is adjusted interval digraph then \( LHOM(H) \) is polynomial solvable problem; otherwise \( L – HOM(H) \) problem is \( NP \)-complete.

And it’s easy to produce next criterion from formulated statements.

**Theorem 4** Let \( P \) is partial order and \( H \) is correspondence this poset p-graph. If \( H \) is adjusted interval digraph then \( S(P) \) polynomial solvable problem.
Finally, note the case of poset $\mathcal{P}$ in language without constants. In this case the problem $S(\mathcal{P})$ solves obvious because of every system $S(X_\alpha)$ over $\mathcal{P}$ will consist of two types of equations: $S_{x=x}$ and $S_{x<\varepsilon}$. Consequently, every variable can be every element in $\mathcal{P}$. And this means in such systems always be a diagonal solution (every coordinate in solution are the same).

4. Example of partial order class

Look at class partial orders for which $S(\mathcal{P})$ problem resolves in polynomial time. This is the class $\mathcal{T}$ of tree-like partial orders. The Hasse diagram of it’s partial orders is oriented tree. It means (i) such partial orders have root, (ii) each vertex except of root has only one input edge and (iii) from every vertex, except of leafs, output arbitrary final number of edges.

Let’s show that every tree-like partial order is adjusted interval digraph. Now we describe algorithm of building interval form of graph.

Algorithm draw an intervals on coordinate axis, so for draw the interval there will note coordinate of the beginning (left point) of the interval and it’s length. All intervals will be parallel to X-axis. Define functions for every vertex of graph: $l(v)$ is depth of the element $v$, $c(v)$ is childs of $v$ on level $l(v)+1$. Next we take agreement to denote for interval $I_v$ of vertex $v$ left point as $a_v$ and right point as $b_v$.

**Input:** p-graph $T$;

**Output:** adjusten interval form of graph $T$.

Choose a little enough value $\varepsilon$. Let $r$ be a root of $T$. For $r$ draw an interval $I_r$ length of $\varepsilon$ from point $(0,0)$. Next we draw an interval $I_r$ length 1 from point $(0,-\varepsilon)$. Then starts procedure $\text{divide}(v)$.

Procedure $\text{divide}(v)$ devides interval $(a_v + \varepsilon, b_v)$ into $|c(v)| = n$ equal parts length of $t = (b_v - a_v - \varepsilon)/n$. For each child $u_i = c(v)[i](i = 0, \ldots, n - 1)$ draws 2 intervals: $J_{u_i}$ length $\varepsilon$ from point $(a_v + \varepsilon + t \cdot i; -l(v)-1)$ and interval $I_{u_i}$ length $t$ from point $(a_v + \varepsilon + t \cdot i; -l(v)-1 - \varepsilon)$. After that starts procedure $\text{divide}(u_i)$.

**Argumentation:** Look at graph’s vertex $v$. We’re see, by building, that $\forall u \ l(u) < l(v) \ J_u \cap I_v = \emptyset$. And also offsets by X-axis was defined in that way what $\forall u \ l(u) = l(v) \ J_u \cap I_v = \emptyset$. Find intersection of intervals $J_u$ and $I_v$ could be not empty only for vertices $u$ that $l(u) > l(v)$ and arc $(v, u) \in E(T)$. So, in conclusion, algorithm builds adjusted interval form of graph $T$.

We showed that $S(\mathcal{P})$ problem is solvable in polynomial time in tree-like class of partial orders $\mathcal{T}$.

Acknowledgments

The research was supported by Russian Science Foundation (project No. 18-71-10028)

References

[1] Feder T, Hell P, and Huang J 2007 List homomorphisms and retraction to reflexive digraphs Manuscript
[2] Feder T, Hell P, Huang J and Rafiey A 2010 Interval Graphs, Adjusted Interval Digraphs, and Reflexive List Homomorphisms *Discrete Applied Mathematics* **160** 697-707
[3] Ore O 1962 *Theory of Graphs* (American Mathematical Society Colloquium publications)
[4] Plotkin B 1999 Algebraic logic, varieties of algebras and algebraic varieties *International Algebraic Conference St.Peterburg*-95 189–271
[5] Daniyarova E Y, Myasnikov A G and Remeslennikov V N 2012 Algebraic geometry over algebraic structures II: Foundations *Journal of Mathematical Sciences* **185** 389-416
[6] Daniyarova E Y, Myasnikov A G and Remeslennikov V N 2011 Algebraic geometry over algebraic structures III: Equationally Noetherian property and compactness *Southeast Asian Bulletin of Mathematics* **35** 35-68
[7] Daniyarova E Y, Myasnikov A G and Remeslennikov V N 2011 Algebraic geometry over algebraic structures IV: Equational domains and codomains *Algebra and Logic* **49** 483-508
[8] Daniyarova E Y, Myasnikov A G and Remeslennikov V N 2012 Algebraic geometry over algebraic structures V: The case of arbitrary signature *Algebra and Logic* **51** 28-40
[9] Daniyarova E Y, Myasnikov A G and Remeslennikov V N 2012 Algebraic geometry over algebraic structures VI: Geometrical Equivalence *Algebra and Logic* **56** 281-94

[10] Shevlyakov A N 2015 Elements of algebraic geometry over a free semilattice *Algebra and Logic* **54** 258-71

[11] Harju T, Karhumaki J and Petrich M 1997 Compactness of systems of equations on completely regular semigroups *Structures in Logic and Computer Science* LNSC **1261** Springer