Journal of Computer and System Sciences 91 (2018) 104–114

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

On the speed of constraint propagation and the time complexity of arc consistency testing

Christoph Berkholz [a], Oleg Verbitsky [b,*,1]

[a] RWTH Aachen University, Institut für Informatik, D-52056 Aachen, Germany
[b] Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, D-10099 Berlin, Germany

ARTICLE INFO

Article history:
Received 12 June 2014
Received in revised form 27 August 2017
Accepted 7 September 2017
Available online 19 September 2017

Keywords:
Constraint satisfaction problem
Constraint propagation
Arc consistency
Time complexity
Existential 2-pebble game
Existential-positive two-variable logic

ABSTRACT

Establishing arc consistency on two relational structures is one of the most popular heuristics for the constraint satisfaction problem. We aim at determining the time complexity of arc consistency testing. The input structures G and H can be supposed to be connected colored graphs, as the general problem reduces to this particular case. We first observe the upper bound $O(e(G)v(H) + v(G)e(H))$, which implies the bound $O(e(G)e(H))$ in terms of the number of edges and the bound $O((v(G) + v(H))^3)$ in terms of the number of vertices. We then show that both bounds are tight as long as an arc consistency algorithm is based on constraint propagation (as all current algorithms are). Our lower bounds are based on examples of slow constraint propagation. We measure the speed of constraint propagation observed on a pair G, H by the size of a combinatorial proof that Spoiler wins the existential 2-pebble game on G, H.

1. Introduction

According to the framework of [1], the *constraint satisfaction problem* (CSP) takes two finite relational structures as input and asks whether there is a homomorphism between these structures. In this paper we consider structures with unary and binary relations and refer to unary relations as colors and to binary relations as directed edges. Note that this restricted class of binary CSPs is known to be polynomial time equivalent to general CSPs [1]. In fact, most of the time we deal with structures where the only binary relation E is symmetric and irreflexive relation, i.e., with vertex-colored graphs. This is justified by a linear time reduction from the CSP on binary structures to its restriction on colored graph; see Section 5.1.

Let G and H be an input of the CSP. It is customary to call the vertices of G *variables* and the vertices of H *values*. A mapping from $V(G)$ to $V(H)$ then corresponds to an assignment of values to the variables, and the assignment is *satisfying* if the mapping defines a homomorphism. Let a *domain* $D_x \subseteq V(H)$ of a variable $x \in V(G)$ be a set of values such that for every homomorphism $h : G \to H$ it holds that $h(x) \in D_x$. The aim of the arc consistency heuristic is to find small domains in order to shrink the search space. The first step of the arc consistency approach is to ensure *node consistency*, that is, D_x is initialized to the set of vertices in H that are colored with the same color as x. The second step is to iteratively shrink the domains according to the following rule:

* Corresponding author.
E-mail address: verbitsk@informatik.hu-berlin.de (O. Verbitsky).

1 Supported by DFG grant VE 652/1-1. On leave from the Institute for Applied Problems of Mechanics and Mathematics, Lviv, Ukraine.

http://dx.doi.org/10.1016/j.jcss.2017.09.003
0022-0000/

If there exists an $a \in D_x$ and a variable $y \in V(G)$ such that $\{x, y\} \in E(G)$ and $\{a, b\} \notin E(H)$ for all $b \in D_y$, then delete a from D_x.

A pair of graphs augmented with a set of domains is *arc consistent* if the above rule cannot be applied and all domains are nonempty. We say that arc consistency *can be established* for G and H, if there exists a set of domains such that G and H augmented with these domains is arc consistent. Our aim is to estimate the complexity of the following decision problem.

AC-Problem	
Input:	Two colored graphs G and H.
Question:	Can arc consistency be established on G and H?

Using known techniques, we observe that the AC-Problem can be solved in time $O(v(G)e(H) + e(G)v(H))$, where $v(G)$ and $e(G)$ denote the number of vertices and the number of edges respectively. This gives us only a quadratic upper bound in the overall input size (where the graphs are encoded using adjacency lists), so there could be a chance for improvement: Is it possible to solve the AC-Problem in sub-quadratic or even linear time? In fact, we cannot rule out this possibility completely. The first author [2] recently obtained lower bounds for higher levels of k-consistency (note that arc consistency is equivalent to 2-consistency). In particular, 15-consistency cannot be established in linear time and establishing 27-consistency requires more than quadratic time on multi-tape Turing machines. The lower bounds are obtained in [2] via the deterministic time hierarchy theorem and, unfortunately, these methods are not applicable to arc consistency because of the blow-up in the reduction.

However, we show lower bounds for every algorithm that is based on constraint propagation. A *propagation-based arc consistency algorithm* is an algorithm that solves the AC-Problem by iteratively shrinking the domains via the arc consistency rule above. Note that all currently known arc consistency algorithms (e.g. AC-1, AC-3 [3]; AC-3.1/AC-2001 [4]; AC-3.2, AC-3.3; AC-3_d [5]; AC-4 [6]; AC-5 [7]; AC-6 [8]; AC-7 [9]; AC-8 [10], AC-$*$ [11]) are propagation-based in this sense. Different AC algorithms differ in the principle of ordering propagation steps; for a general overview we refer the reader to [4]. The upper bound $O(v(G)e(H) + e(G)v(H))$ implies $O(e(G)e(H))$ in terms of the number of edges and $O(n^3)$ in terms of the number of vertices $n = v(G) + v(H)$. Our main result, Theorem 5.3 in Section 5, states that both bounds are tight up to a constant factor for any propagation-based algorithm.

We obtain the lower bounds by exploring a connection between the *existential 2-pebble game* and propagation-based arc consistency algorithms. In its general form, the existential k-pebble game is an Ehrenfeucht–Fraïssé like game that determines whether two finite structures can be distinguished in the existential-positive k-variable fragment of first order logic. It has found applications also outside finite model theory: to study the complexity and expressive power of Datalog [12], k-consistency tests [13–15,2] and bounded-width resolution [16,17]. It turns out that the existential 2-pebble game exactly characterizes the power of arc consistency [13], i.e., Spoiler wins the existential 2-pebble game on two colored graphs G and H iff arc consistency cannot be established.

The connection between the existential 2-pebble game and arc consistency algorithms is deeper than just a reformulation of the AC-Problem. We show that every constraint propagation-based arc consistency algorithm computes in passing a proof of Spoiler's win on instances where arc consistency cannot be established. On the one hand these proofs of Spoiler's win naturally correspond to a winning strategy for Spoiler in the game. On the other hand they reflect the propagation steps performed by an algorithm. We consider three parameters to estimate the complexity of such proofs: length, size and depth. The length corresponds to the number of propagation steps, whereas size also takes the cost of propagation into account. The depth corresponds to the number of "nested" propagation steps and precisely matches the number of rounds $D(G, H)$ Spoiler needs to win the game. We observe that the minimum size of a proof of Spoiler's win on G and H bounds from below the running time of sequential propagation-based algorithms, whereas the minimal depth matches the running time of parallel algorithms.

We exhibit pairs of colored graphs G, H where $D(G, H) = \Omega(v(G)v(H))$ and hence many nested propagation steps are required to detect arc inconsistency. Because these graphs have a linear number of edges this implies that there is no sub-quadratic propagation-based arc consistency algorithm. It should be noted that CSP instances that are hard for sequential and parallel arc consistency algorithms, in the sense that they require many propagation steps, were explored very early in the AI-community [18,19]. Such examples were also proposed to serve as benchmark instances to compare different arc consistency algorithms [20]. Graphs G and H with large $D(G, H)$ can be derived from the old Domino example [20], consisting of structures with two binary relations. We also provide a new example, which we call Co-Wheels, that shows the same phenomenon of slow constraint propagation for a more restricted class of rooted loopless digraphs.

The rest of the paper is organized as follows. In Section 2 we give the necessary information on the existential 2-pebble game and use it to analyze the Domino pattern. Our Co-Wheels pattern is introduced and analyzed in Section 3. Section 4 is devoted to the winner proof system for the existential 2-pebble game. The facts obtained here are used in Section 5 to prove our main results on the complexity of propagation-based algorithms for the AC-Problem.

A preliminary version of this paper appeared in [21].