Contents

A Dynamic Parsing Algorithm for Pregroup Grammars
Katarzyna Moroz 1

Aspects of Non-Deterministic Automata Minimisation
Daniel Quernheim 2

Bisection Width of Multidimensional Product Graphs
Jordi Arjona Aroca, Antonio Fernández Anta 3

Complexity of Two-Variable Logics Extended with Monadic Datalog Programs
Witold Charatonik, Piotr Witkowski 4

Continuous Reductions of Regular Languages
Filip Mazowiecki 5

Doubly Unambiguous Regular Languages of Infinite Trees
Marcin Bilkowski 6

Equational Theories of Profinite Structures
Michał Skrzypczak 7

Fast Exact Algorithm for L(2,1)-Labeling of Graphs
Konstanty Junosza-Szaniawski, Jan Kratochvíl, Mathieu Liedloff, Peter Rossmanith, Paweł Rzążewski 8

Fault-containing Self-stabilization
Sven Köhler 9

Generalized Maneuvers in Route Planning
Ondrej Moríš 10

Hyper-Minimisation of Weighted Automata
Daniel Quernheim 11

Optimal (Fully) LZW-compressed Pattern Matching
Paweł Gawrychowski 12

Oriented Chromatic Number of Grids
Anna Nenca, Janusz Dybizbański 13

Permutation Languages
Grzegorz Madejski 14

Preventing Cactus Fire
Peter Floderus 15

Seeking Fast Operations in MWMR Atomic Register Implementations
Nicolas Nicolaou 16

Steady Marginality: A Uniform Approach to Shapley Value for Games with Externalities
Oskar Skibski 17

The Effects of Tie-Breaking Rules on Repeated Auctions with Discrete Bidding
Josué Ortega 18

The Stubborn Problem Is Stubborn No More (A Polynomial Algorithm for 3-Compatible Colouring and the Stubborn List Partition Problem)
Michał Pilipczuk 19
A Dynamic Parsing Algorithm for Pregroup Grammars

Katarzyna Moroz

Adam Mickiewicz University, Poznań

Pregroup grammars are used as an algebraic tool for the syntactical analysis of natural languages. The formalism was introduced by Lambek [2] and it belongs to the tradition of categorial grammars. The problem of parsing pregroup grammars has been a subject of study in recent years.

A pregroup is a partially ordered monoid with two additional unary operations (adjoints). In a pregroup grammar each word in the lexicon is assigned one or more syntactic types.

We present a cubic, dynamic parsing algorithm for ambiguous pregroup grammars. The algorithm modifies essentially the recognition algorithm of Savateev [4] defined for unidirectional Lambek grammars. It works on a special form of a string $W$ containing all possible type assignments for words of the string to parse, it decomposes $W$ into atomic parts and using an auxiliary function $M$ it decides dynamically which substrings can be ”accepted”. The given string is a sentence if and only if the whole string of types is ”accepted” by the function $M$, see [3].

In the proof we show that the algorithm computes all values of the function $M$ according to its definition and we show that the algorithm finds correctly all substrings that are ”accepted” by the function $M$.

We extend the algorithm to the full parsing algorithm. We give some examples of the run of the algorithm on English sentences. Similar algorithms are defined for pregroup grammars with letter promotions and pregroup grammars with letter promotions with 1 [1].

References


Aspects of Non-Deterministic Automata Minimisation

Daniel Quernheim

Universität Stuttgart

Automata theory is a main branch of theoretical computer science with applications in many different fields. Since in practice, automata can be very large, efficient algorithms for standard operations (minimisation, intersection, determinisation, composition) are crucial for performance. Usually, these automata are also weighted (wfa) with weights from a semiring to model e.g. costs and probabilities.

In practice, two properties are generally desirable: we want the automata to be deterministic (dfa) and as small as possible (minimal). In this respect, weighted and unweighted fa behave quite differently:

- Every unweighted fa can be determinised (transformed in an equivalent dfa), but not every wfa can.
- Unweighted dfa can be uniquely minimised in $O(n \log n)$ with a classic algorithm by Hopcroft. The minimal automaton is generally not unique in the weighted dfa case, and the $O(n \log n)$ strategy only works in certain semirings.
- Nondeterministic minimisation of unweighted fa is NP-complete, but (big surprise!) can be achieved in $O(n^3)$ for wfa over fields with an algorithm due to Schützenberger (see, e.g. [1, III.4.3]).

The last result is little known and has not been generally implemented in toolkits (with the notable exception of Vaucanson\(^1\)) and deserves attention, because: a) It might help create smaller models without the need to determinise (which is not possible in every case); b) When applying operations in sequence, the order can have a significant impact on efficiency; for instance, minimisation followed by determinisation might perform better than the other way around.

In this talk, I will illustrate the algorithm as well as an empirical evaluation on random and actual automata and discuss the different use cases.

References


\(^1\)http://www.lrde.epita.fr/cgi-bin/twiki/view/Vaucanson/
Bisection Width of Multidimensional Product Graphs

Jordi Arjona Aroca$^{1,2}$, Antonio Fernández Anta$^{1}$

$^{1}$Institute IMDEA Networks, $^{2}$Universidad Carlos III de Madrid

In this paper we will provide two general results that allow to obtain upper and lower bounds on the bisection width of a product graph as a function of some properties of its factor graphs. The most interesting contribution of this paper is the exact value of the bisection width of a $d$-dimensional torus, as this problem has been open for almost 20 years [2]. Our work is partially based on the work by Azizoğlu and Eğecioğlu. In [1] they study the relation between the isoperimetric number and the bisection width of different product networks and obtained and exact value for the bisection width of the $d$-dimensional array studying it as a product of paths. Similarly, we have been able to provide a lower and an upper bound for the bisection width of product graphs whose factor graphs have the same maximal congestion with multiplicity $r$, for the former case, or the same central cut, for the latter one.

The general results provided are used to obtain exact bounds on the bisection width of several product graphs. The factor graphs used are paths, rings, complete binary trees (CBTs), and extended trees (which are CBTs with the leaves connected as a path). Then, we show that the Cartesian product of rings (i.e., the torus) of sizes $k_1 \geq \ldots \geq k_d$ has bisection width $2 \sum_{i=1}^{e} C_i$, where $C_i = \prod_{j=i+1}^{d} k_j$ for $i \in [1, e]$, and $e$ being the lowest dimension with an even number of vertices. (If there is no such dimension, $e = d$.) Additionally, we show that the Cartesian product of a mixture of XTs and rings has the same bisection width. (When all factor graphs are XTs, $e = d$.) Finally, we show that the Cartesian product of a mixture of CBTs and paths has bisection width $\sum_{i=1}^{e} C_i$. (When all factor graphs are CBTs, $e = d$.)

References


Complexity of Two-Variable Logics Extended with Monadic Datalog Programs

Witold Charatonik, Piotr Witkowski
University of Wrocław

Since 1930s, when Alonzo Church and Alan Turing proved that the satisfiability problem for first-order logic is undecidable, much effort was put to find decidable subclasses of this logic. Still, all such fragments have a limited expressive power and a lot of effort was put to extend (at least some of) them beyond the first order logic while preserving decidability. When verifying imperative programs manipulating dynamically allocated pointer structures one often wants to express in such logics the following facts: a data structure that resides on a heap is acyclic, there is a path between certain nodes or the heap contains no garbage elements. These reachability properties are naturally specified using transitive closure or least fixed point operators. However, even quite simple logics like two variable fragments of first order logic or even two variable fragment of Bernays-Schönfinkel class extended with either transitive closure or least fixed point operators are undecidable [3,4]. In [1] the authors define a logic, which is a combination of the two variable Bernays-Schönfinkel class and the monadic Datalog and prove that its satisfiability problem is decidable in 2NEXPTIME over some class of relational structures. The logic may be considered a subclass of the first order logic with least fixed point and monadic Datalog is powerful enough to express some of reachability properties. The result is used there to give a bounded model checking procedure for programs with pointers. In [2] we extend the logic in order to be useful in Hoare-style verification and prove NEXPTIME-completeness of the satisfiability problem for both the extension and the original logic. In the follow up work we derive and analyze some other combinations of two variable logics with Datalog. The long term goal is to establish complexity of the satisfiability problem for all these combinations.

References


This paper describes the hierarchies of regular languages over a finite alphabet \( A \). The Borel hierarchy measures the minimum number of alternations of complements and countable sums of basic sets to define a language. The Wadge hierarchy is a more precise language description. Wagner’s result was the hierarchy for languages of infinite words recognized by Büchi automata for Wadge (continuous) reductions. For two subsets \( L, M \subseteq A^\omega \) we say that \( L \) is Wadge (continuous) reducible to \( M \) if there exists a continuous function \( f : A^\omega \to A^\omega \) such that \( f^{-1}(M) = L \).

We define the metric on \( A^\ast \) by embeddings \( f : A^\ast \to (A \cup \{\$\})^\omega \) where \$ is an extra letter: \( d_f(x, y) = d(f(x), f(y)) \) for \( x, y \in A^\ast \). We propose a definition of embeddings which preserve some properties, so we can work on the image of the words. In all cases \( A^\ast \) unlike \( A^\omega \) is not compact. That is why we consider the hierarchy for continuous reductions and uniformly continuous reductions separately. The space \( A^\ast \) with a given metric can be considered a uniform space. We show that with our definition of embeddings there are only two uniform structures up to isomorphism. Thus they have the same hierarchies of languages for continuous and uniformly continuous reductions.

We define two examples of embeddings inducing metrics for both types of the uniform structures. The first metric (discrete) is induced by \( f_1(x) = x\$\omega \). In this case all of the points in \( A^\ast \) are isolated. The second metric (rational) is induced by \( f_2(x) = x10^\omega \), where \( 0, 1 \in A \). In this case none of the points are isolated. For both of these embeddings we describe complete hierarchies of regular languages for continuous and uniformly continuous reductions. In the case of the discrete metric the hierarchy for regular languages defines equivalence classes also for all other languages in \( A^\ast \). For the rational metric we show examples of languages that are not regular and are not in an equivalence class with any regular language.

We also consider the hierarchy of languages for 1-Lipschitz reductions. In \( A^\omega \) space we show that the hierarchy of Borel languages for 1-Lipschitz reductions is an extension of the hierarchy for Wadge reductions. If a Borel language \( A \) is not self-dual then its equivalence class is the same as for continuous reductions. If \( A \) is self-dual then its equivalence class separates into \( \omega \) classes. Then we show that analogous constructions give similar results in the case of \( A^\ast \) with metrics defined by embeddings.

Other papers considered metrics on \( A^\ast \) defined by a variety of languages. We show that the discrete metric corresponds to a variety of languages of the form \( FA^\ast + G \) where \( F, G \) are finite and that there is no such variety for the rational metric.
Doubly Unambiguous Regular Languages of Infinite Trees

Marcin Bilkowski  
University of Warsaw

It is very convenient to have some kind of canonical form of finite automata when dealing with regular languages. For regular languages of finite or infinite words, deterministic automata can be used for that purpose. On the other hand, not all regular languages of finite or infinite trees can be recognized by deterministic (top-down) automata [3].

The search for canonical form of tree automata has led to the introduction of unambiguous automata and unambiguous languages. An automaton is unambiguous if for every input it has at most one accepting run. A tree language is unambiguous if there exists an unambiguous automaton recognizing this language. It is known that all regular languages of finite trees are unambiguous. Unfortunately, it was shown in [1] that this is not the case for regular languages of infinite trees.

This rises another question: is it decidable if a regular language of infinite trees is unambiguous? The answer is still unknown. In this work we give an answer to a simpler question. We show that it is decidable if both a tree language and its complement are unambiguous. To prove it, we use a slightly modified algebra for regular languages of infinite trees introduced in [2]. We also introduce a new subclass of regular languages of infinite trees – thin languages.

References


Equational Theories of Profinite Structures

Michał Skrzypczak
University of Warsaw

In this work we consider a general way of constructing profinite structures based on a given framework — a countable family of objects and a countable family of recognisers (e.g. formulas). The main theorem states:

A subset of a family of recognisable sets is a lattice if and only if it is definable by a set of profinite equations.

This result extends Theorem 5.2 from [1] expressed only for finite words and morphisms to finite monoids.

One of the applications of our theorem is the situation where objects are finite relational structures and recognisers are first order sentences. In that setting a simple characterisation of lattices of first order formulas arise.

Profinite structures can be defined in various different ways. The idea of this particular construction is that it seems to be straightforward and quite simple. Moreover, it turns out that the above characterisation of lattices follows only from topological properties of profinite structures.

References

Fast Exact Algorithm for L(2,1)-Labeling of Graphs

Konstanty Junosza-Szaniawski¹, Jan Kratochvíl², Mathieu Liedloff³, Peter Rossmanith⁴, Paweł Rzążewski¹

¹Warsaw University of Technology, ²Charles University, ³Université d’Orléans, ⁴RWTH Aachen University

The L(2,1)-labeling is a graph coloring model which arises from channel assignment in telecommunication. It asks for such a labeling with nonnegative integer labels, that no vertices in distance 2 in a graph have the same label and labels of adjacent vertices differ by at least 2.

By \( \lambda(G) \) we denote an L(2,1)-span of \( G \), which is the smallest value of \( k \), for which there exists an L(2,1)-labeling of \( G \) with no label exceeding \( k \). Determining \( \lambda(G) \) was proven to be NP-complete by Griggs and Yeh [1].

In this talk we present the fastest currently known exact algorithm for L(2,1)-labeling [2]. It works in time \( O(2.6488^n) \) and exponential space.

The algorithm is based on dynamic programming and exploits the idea for fast matrix multiplication, presented by Strassen [3]. It also uses auxiliary combinatorial results, which are interesting on their own.

References


Fault-containing Self-stabilization

Sven Köhler
Hamburg University of Technology

Self-stabilizing distributed protocols recover from transient faults of any scale or nature, including memory corruptions and topology changes. However, self-stabilizing protocols are often only optimized with respect to their reaction upon large-scale faults. Bounding the impact of transient small-scale faults has become an additional objective being pursued with two independent focuses: Optimizing the protocol’s reaction upon topological changes (e.g. super-stabilization [1]) and speeding up the recovery from memory corruptions (e.g. fault-containment [2]). Super-stabilizing protocols provide certain guarantees while they adapt to a new topology. However, we argue that they are based on mechanisms for detecting topology changes which break in the face of memory corruptions. The output of fault-containing self-stabilizing protocols recovers in constant time from small-scale memory corruptions. But all known transformations cease to work reliably in the face of topology changes.

Our contribution is twofold: First, we describe a transformation that adds fault-containment of small-scale memory corruptions to any silent self-stabilizing protocol (published as [3]). It is the first transformation for which the fault-gap, that is the minimal time between two containable faults, is constant. Also, the transformation increases the time complexity by a constant factor only. This is a considerable improvement over previously known transformations. Second, we describe how this transformation is extended such that containment of memory corruptions succeeds even in face of topology changes (currently submitted for publication). We conclude our contribution with the result that the impact of memory corruptions and topology changes can be bounded even if both happen simultaneously.

References


Generalized Maneuvers in Route Planning

Ondrej Moriš
Masaryk University

We study an important practical aspect of the route planning problem in road networks – maneuvers. Informally, maneuvers are walks that might either decrease or increase the weight of a route by additional costs. By maneuvers one can model a wide range of route restrictions or traffic regulations such as turn-prohibitions, traffic light delays, forbidden passages and so on. Maneuvers are studied only partially by attacking only specific (rather simple) types of restriction such as turn-penalties or path prohibitions. Unfortunately, a sufficiently generic approach for arbitrarily complex maneuvers is still missing despite the fact that such a solution might able to capture not only route restrictions and traffic regulations, but also some sort of dynamic road network properties such as car accidents, time dependent restrictions and others.

In this work, a new approach to tackle maneuvers during route planning queries without prior adjustments of the underlying road network graph is presented. First, a generic maneuver is defined so that we are able to model almost arbitrarily complex route restriction. In the rest we show how to modify Dijkstra’s algorithm to respect maneuvers together with a formal proof of correctness and a complexity analysis.

Definition 1. A maneuver $M$ of $G$ is a walk in $G$ that is assigned a penalty $\Delta(M) \in \mathbb{R}$. A set of all maneuvers of $G$ is denoted by $\mathcal{M}$.

There are two specific kinds of maneuvers; restricted ones of the penalty 0 and the prohibited ones of the penalty $\infty$. The penalized weight of a walk is the sum of its weight and penalties of maneuvers contained as its subwalks. A graph, its weighting and maneuvers form a road network. It is called a proper road network if no two maneuvers of negative penalties overhang, all walks have non-negative penalized weight and there are no conflicts between restricted maneuvers.

Theorem 1. Let a proper road network $(G, w, \mathcal{M})$ and vertices $s, t \in V(G)$ be given. $\mathcal{M}$-Dijkstra’s algorithm computes a valid walk from $s$ to $t$ in $G$ optimal w.r.t. the penalized weight in time $O\left(c_m^2|E(G)| + c_m|E(G)| + c_m|V(G)| \log |V(G)|\right)$ where $c_m = \max_v |\{X \in \mathcal{M} \mid v \in V(X)\}|$ is the maximum number of maneuvers per vertex.

References

Hyper-Minimisation of Weighted Automata

Daniel Quernheim
Universität Stuttgart

In this talk, I will address “hyper-minimisation” [1], which is a recent technique that can be used to compress dfa (deterministic finite automata) beyond the minimal dfa by allowing a finite change in the language. Two dfa are almost-equivalent when their languages are almost-equal (their symmetric difference is finite), and a dfa $M$ which is minimal among all almost-equivalent automata is called hyper-minimal. The best known algorithms for hyper-minimisation run in $O(n \log n)$ [2,3], and there is an $O(n^2)$ algorithm which returns the “best” hyper-minimal dfa in terms of mistakes [5]. I will give a short survey about the state of research.

Then, I will discuss a generalisation of hyper-minimisation to weighted dfa over semifields [4]. In this setting, minimality is well-defined and efficiently computable. We define weighted almost-equivalence as a finite number of words which are assigned a different weight. Existing algorithms for dfa minimisation (in particular, the canonicalisation known as “pushing”) and unweighted hyper-minimisation can be straightforwardly combined into an $O(n \log n)$ algorithm for weighted hyper-minimisation. Finally, I will briefly present ongoing and unpublished work in which the case where the weight structure is not a semifield, and minimisation is not well-defined, is addressed.

References


Even though the processing capabilities of a modern hardware are growing very rapidly, so is the amount of the digital data we gather and store. This motivates not only working on inventing efficient compression methods (with efficient meaning quick and resulting in a significant size decrease) but also algorithms for processing the data using just the compressed representation. One of the most natural problems in this line of research is the following variant of the classical pattern matching: given an uncompressed pattern $s[1..m]$ and a compressed representation of a string $t[1..N]$, does $s$ occur in $t$? This was first considered in 1994 by Amir, Benson, and Farach [1] who developed two algorithms working in $O(n \log m)$ and $O(n + m^2)$ time, the latter soon improved by Kosaraju [4] to take just $O(n + m^{1+\epsilon})$, where $n$ is the size of compressed $t$. I will briefly sketch the main idea underlying the optimal linear time solution for this problem [3]. Then I will try to give some intuition how to apply this result to develop a (unpublished) linear time algorithm for the fully compressed version of the problem, where both the pattern and the text are compressed, hence improving the linear-logarithmic solution of Gąsieniec and Rytter [2].

References


Oriented Chromatic Number of Grids

Anna Nenca, Janusz Dybizbański

University of Gdańsk

Fertin et al. [1] stated two conjectures. The first one says every 2-dimensional grid can be colored by homomorphism to $T_7$ ($T_7$ is a graph with the set of vertices $V = \{0, 1, \ldots, 6\}$ and the set of arcs $A = \{(x, x + b \mod 7) \mid x \in V, \ b = 1, 2, \text{ or } 4\}$). The second one says every 2-dimensional oriented grid can be colored with seven colors. The last one is the equivalent to the fact that for every such grid $G$, there is a homomorphism from $G$ to an oriented graph $H$ with seven vertices, without self loops, and with no arcs going in opposite directions. The first conjecture was disproved by Szepietowski et al. [2]. We have disproved the second one by showing that there exist 2-dimensional oriented grids that cannot be colored with seven colors. Our proof is computer-aided. We describe the algorithm that for given graph $H$ builds an oriented graph that cannot be colored by $H$. Using this algorithm we have found eight oriented grids, such that no coloring graph can color all of them. One can build one big grid which contains these eight grids as subgraphs and which is not colorable by any graph with seven vertices.


Permutation Languages

Grzegorz Madejski

University of Gdańsk

In the theory of formal languages, one of the most important classes of languages are context-free and context-sensitive languages, denoted $CF$ and $CS$ respectively. Context-free languages are commonly used in compilers. However, this class is too weak to describe such structures as natural languages or some models of parallel processes. Context-sensitive language class is, on the other hand, too broad and complex to be useful. To fill this gap, several classes between these two were introduced, one of which is the class of permutation languages.

Following [1], we define $Perm$ as a class of languages having only context-free rules and rules of the form $AB \rightarrow BA$, and mention some of its properties.

We recall the shuffle operation $\circ$ of two words $u, v$ over any alphabet $\Sigma$ as $u \circ v = \{u_1v_1u_2v_2...u_kv_k : u_i, v_i \in \Sigma^*, 1 \leq i \leq k, u = u_1u_2...u_k, v = v_1v_2...v_k\}$. For two languages $L_1, L_2 \subset \Sigma^*$ we have $L_1 \circ L_2 = \{u \circ v : u \in L_1, v \in L_2\}$. We also define shuffle closure of a language $L \in \Sigma^*$ as $L^\circ = \bigcup_{i=0}^{\infty} L^\circ i$ where $L^\circ 0 = \{\lambda\}$ (empty word) and $L^\circ i = L^\circ i−1 \circ L$ [4]. By extending regular expressions with $\circ$ and $\otimes$, we obtain shuffle expressions generating a class of shuffle languages $Sh$.

We investigate some relations between $Perm$ and $Sh$ classes and try to compare their expressive power. The membership problem for $Sh$ is in $P$, as was shown in [3]. However, using the results of [2], we show that the membership problem for $Perm$ is $NP$-complete. For arbitrary regular languages $L, K, M$, we show that $L^\circ \in Perm$ and $(KL^\circ M)^\circ \in Perm$. It remains an open problem to show whether or not $((L^\circ K)^* M)^\circ \in Perm$.

Finally, we introduce some other open problems related to the $Perm$ class. It is an interesting task to develop tools for showing that a language is not in $Perm$. One can also allow interchange rules to be of length more than 2 (e.g. $ABC \rightarrow CBA$) and investigate if such grammar’s expressive power increases with the growing length of the rules [1].

References


The firefighter problem simulates a fire spreading throughout a topology defined by a graph. To stop the fire, we are allowed to place firefighters during this process. The objective is to find the set of vertices on where to place firefighters and when to place them in order to maximize the number of saved vertices. This is hard, even for trees of degree greater than 3 this is NP-hard, so we turn our attention to approximation methods. For general graphs, it is NP-hard to approximate the problem within a factor $n^{1-\varepsilon}$ [2], but for trees a constant factor approximation algorithm exists [1]. In this paper, we extend the greedy heuristic defined by Hartnell and Li to work on some generalizations of trees called “Huisimi trees” or “Cactus Graphs”. The approximation method used in this paper is a greedy heuristic very similar to the one used by Hartnell and Li to show that the greedy heuristic on trees yields a $\frac{1}{2}$-approximation. First, we observe that the property that allow the problem to be efficiently approximated on trees is that the set of paths from any vertex to the root only contains a single element. We extend this property to having the sets contain at most two disjoint subpaths between each vertex and the root. This yields exactly the Cactus Graphs. The greedy heuristic is modified to consider pairs of vertices since the set of paths from any vertex to the root contains at most two elements. This method applied to the Cactus Graphs is shown to yield a $\frac{1}{3}$-approximation. This is essentially shown by proving that the greedy heuristic will place a “good” firefighter for each two “bad”. The bound is tight up to some arbitrarily small factor $\varepsilon$.

References


2. Elliot Anshelevich and Deeparnab Chakrabarty and Ameya Hate and Chaitanya Swamy Approximation Algorithms for the Firefighter Problem: Cuts over Time and Submodularity. ISAAC’09, 974-983, 2009
In this work we explore the communication and computation costs of multiple-writer multiple-reader (MWMR) atomic read/write register implementations in asynchronous message-passing systems with crash-prone processors. We consider algorithms that use quorum systems, collections of subsets of replica hosts with pairwise intersection, called quorums. A quorum system has intersection degree \( n \) (also called \( n \)-wise quorum system), if every \( n \) quorum members of this system have a non-empty intersection. Given this definition we show that a MWMR atomic register implementation deploying an \( n \)-wise quorum system, allows up to \( n - 1 \) consecutive fast write operations.

In order to enable fast write operations we introduce a new technique we call server side ordering (SSO), that transfers partial responsibility of the ordering of write operations to the replica hosts. Using this idea we design algorithm \( SFW \) that uses \( n \)-wise quorum systems and relies on predicates to allow fast read and write operations. The algorithm uses tag-value pairs to order the write operations and combines a global ordering imposed by the servers with a local ordering established by each writer participant. A predicate is used during each read and write operation to establish if sufficient number of servers assign the same order to a written value. If the predicate holds then the read or write operation completes in a single round. \( SFW \) implementation is near optimal in terms of the number of successive fast operations it allows.

We formulate a new combinatorial problem that captures the computational burden of evaluating the predicates in algorithm \( SFW \) and we show that it is NP-Complete. To make the evaluation of the predicates feasible, we present a polynomial log-approximation algorithm for this problem and we show how to use it with algorithm \( SFW \). Lastly we identify that \( SFW \) allows fast operations under restrictions on the construction of the underlying quorum system. We then design an algorithm that trades the speed of write operations for removing any constraints on the quorum system construction. The new algorithm, called \( CwFr \), incorporates Quorum Views – algorithmic techniques presented in the SWMR model – to enable fast read operations. \( CwFr \) allows single round reads and two round write operations.

References


\(^1\)This work is supported in part by the Cyprus Research Promotion Foundation’s grant IIENEK/0609/31 and the European Regional Development Fund.
Steady Marginality: A Uniform Approach to Shapley Value for Games with Externalities

Oskar Skibski
University of Warsaw

The Shapley value [1] is one of the most important solution concepts in cooperative game theory. In coalitional games without externalities, it allows to compute a unique payoff division that meets certain desirable fairness axioms. However, in many realistic applications where externalities are present, Shapley’s axioms fail to indicate such a unique division. Consequently, there are many extensions of Shapley value to the environment with externalities proposed in the literature built upon additional axioms. Two important such extensions are “externality-free” value by Pham Do and Norde [2] and value that “absorbed all externalities” by McQuillin [3]. They are good reference points in a space of potential payoff divisions for coalitional games with externalities as they limit the space at two opposite extremes. In a recent, important publication [4], De Clippel and Serrano presented a marginality-based axiomatization of the value by Pham Do and Norde. We propose a dual approach to marginality which allows us to derive the value of McQuillin. Thus, we close the picture outlined by De Clippel and Serrano.

References


The Effects of Tie-Breaking Rules on Repeated Auctions with Discrete Bidding

Josué Ortega
University of Guanajuato

I study a model with three bidders and two items that will be allocated through a repeated second price auction, in which allowed bids will be restricted to a discrete space $\Omega$. In consequence, allowed bids are finite, between the interval $[\underline{v}, \overline{v}]$ and the probability of two bids two be the same is strictly positive.

I analyze this model using the common tie breaking rule (in case of tie of highest bids, the item is allocated randomly with the same probability to be awarded by any of the players who tied, paying the bid they offered), and using my own tie breaking rule. This new tie breaking rule specifies that in case of a tie of the highest bids, the item will be awarded equally than before, but now paying a third price, that in this model is equal to the lower bid. Cases of triple tie are irrelevant, and in the case the triple ties happens, the auction is repeated without any cost, but now using the common tie-breaking rule.

This new tie-breaking rule can be consider as a mechanism which, in specific cases, increases the auction efficiency by giving incentives to one bidder to lie about his real valuation(specifically in the second period), which leads to an increase on seller’s revenue, as long as the tie does not happens. This model is the first one in the literature in which, in a second price auction, players lie about their private information.
The Stubborn Problem Is Stubborn No More (A Polynomial Algorithm for 3-Compatible Colouring and the Stubborn List Partition Problem)

Michał Pilipczuk
University of Warsaw

Based on the article with the same title, published in the proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011; joint work with Marek Cygan, Marcin Pilipczuk and Jakub Onufry Wojtaszczyk.

The $k$-compatible colouring problem asks, given a complete graph with each edge coloured in one of $k$ colours, whether one can assign each vertex one of these $k$ colours in order not to create a monochromatic edge, i.e., an edge with the same colour as both of its endpoints. The 2-compatible is easily tractable in polynomial time; however, for $k \geq 4$ the problem becomes NP-complete. The question of tractability of 3-compatible colouring in polynomial time has been a long-lasting open question. The evidence that the answer should be affirmative was the quasipolynomial algorithm by Hell et al. [SODA 2005], running in time $n^{O\left(\frac{\log n}{\log \log n}\right)}$. In our paper at SODA 2011 we showed a polynomial time algorithm for 3-compatible colouring with time complexity $O(n^7)$.

The motivation of 3-compatible colouring stems from the research on the famous CSP Dichotomy Conjecture, formulated in 1993 by Feder and Vardi, which roughly states that every problem expressible as a Constraint Satisfaction Problem is either NP-complete or polynomial time solvable. A large amount of research has gone into checking various specific cases of this conjecture. One such variant, which attracted a lot of attention in the recent years, is the List Matrix Partition problem. In 2004 Cameron et al. [SODA 2004] classified time complexity of List Matrix Partition for almost all the $4 \times 4$ matrices. The only two cases that resisted classification were polynomially equivalent to a problem that, due to its resistance to the attacks, became known as the stubborn problem. It appears that the stubborn problem is polynomially reducible to 3-compatible colouring; therefore, the result also finishes the classification of List Matrix Partition problems for $4 \times 4$ matrices.