Collection of Abstracts

13th Joint Conference on Mathematics and Informatics
October 1-3, 2020
Contents

Invited Talks

Szilárd András: Ulam-Hyers stability of some second order differential equations defining special functions ................................................................. 7
István Faragó: Operator splittings and their applications .................................. 8
András Lőrincz: Human Machine Intelligent Interaction: Has time arrived? ....... 10
Dan Mircea Suciu, Dănuţ Ilisei: Automatic Person Identification based on Physical Activity Patterns Recognition .................................................. 12

Regular Contributions

Gereltsetseg Altangerel, Máté Tejfel: Some optimization possibilities in data plane programming ................................................................. 13
Bence Babati, Norbert Pataki: The Role of Implementation-specific Static Analysis . 14
Ágnes Backhausz, Edit Bognár: Epidemic spread on random graphs with multiple type nodes ................................................................. 17
Tamás Balla, Gábor Kusper, Csaba Biró, Tibor Tajti: BaW 2.0 - A Problem Specific SAT Solver for Balatonboglár Models Generated from Digraphs ............ 19
Brigitta Baranyai, Melinda Tóth, István Bozó: Supporting Secure Coding with RefactorErl ................................................................. 21
Teshome Bayleyegn, Ágnes Havasi: Multiple Richardson Extrapolation and its Combination with the Implicit Euler Method ................................ 24
András Báleczki, Bálint Molnár: Defining vertex types and metric for graph-based Information System models ........................................ 26
Katalin Bene, László Szabó: Lloyd’s clustering method is not 1-separability detecting . 28
Péter Bereczky, Dániel Horpács, Simon Thompson: A Survey on Comparing Various Formal Semantics Definition Styles ........................................ 30
Lívia Boda: Investigation of some operator splitting methods ................................ 32
Dávid Bóka, Péter Burcsi: Canonical expansion of integers for families of roofline polynomials ................................................................. 33
Attila Péter Boros, Péter Lehotay-Kéry, Attila Kiss: Performance impact of network encryption on log processing with Spark .............................................. 37
Khawla Bouafia, Maxim Kumundzhiev, Bálint Molnár: Application of models and hypergraphs on dynamic aspects of business process performance analysis .... 39
Péter Burcsi, Gábor Nagy: Applications of numeration systems for preserving privacy ................................................................. 40
Péter Burcsi, Gábor Nagy, Attila Réti: Remarks on the quantum complexity of some numeration related problems ........................................ 42
Iulia-Elena Chiru: Regular elements and generalized inverses in (matrix) rings of residue classes ................................................................. 44
Tiberiu Coconeţ: Frobenius induction for algebras ........................................... 46
Septimiu Crivei: Relatively divisible and relatively flat objects in exact categories . 48
István Donkó, Ambrus Kaposi, Melinda Tóth: Formalizing a relational model of concurrent programs in a dependently typed environment ...................... 50
Tamás Dózsa, Ferenc Schipp: Blaschke-products and Hyperbolic Geometry ........ 52
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zsófia Erdei, Melinda Tóth, István Bozó : Graph-based duplicated code detection with RefactorErl</td>
<td>49</td>
</tr>
<tr>
<td>Szabolcs Levente Fancsali, János Szente : About spatially homogeneous space-time models</td>
<td>51</td>
</tr>
<tr>
<td>István Faragó, R. Mosleh : Positivity Preserving Numerical Method for an Extended Ross Model for Malaria Propagation</td>
<td>53</td>
</tr>
<tr>
<td>Izabella Ingrid Farkas, Attila Kovács : (2,3)-Simultaneous Number Systems over the Eisenstein lattice</td>
<td>55</td>
</tr>
<tr>
<td>Hayder K. Fatlawi, Attila Kiss : Efficiency Improvement of Adaptive Random Forest using Principle Component Analysis for Mining Data Stream</td>
<td>56</td>
</tr>
<tr>
<td>István Fazekas, Attila Barta : Theoretical and simulation results for a 2-type network evolution model</td>
<td>57</td>
</tr>
<tr>
<td>István Fazekas, Suja Michael Ochieng : Asymptotic results for contaminated runs of heads</td>
<td>59</td>
</tr>
<tr>
<td>Anett Fekete, Zoltán Porkoláb : Identifying Vulnerable Software Modules with Static Analysis and Version Control</td>
<td>61</td>
</tr>
<tr>
<td>Ádám Fodor : Enhancing Apparent Personality Trait Analysis with Cross-Modal Embeddings</td>
<td>63</td>
</tr>
<tr>
<td>Dávid Fonyó, Péter Burcsi : Automated defense against side-channel attacks</td>
<td>65</td>
</tr>
<tr>
<td>Áron Fóthi : Train multi object tracker on similar instances</td>
<td>67</td>
</tr>
<tr>
<td>Endre Fülöp, Norbert Pataki : A Pragmatic Static Analysis Method with An Applied Model of Formalized Concurrency</td>
<td>69</td>
</tr>
<tr>
<td>Endre Fülöp, Norbert Pataki, Csaba Rotter : Modeling Resource Allocations in Cloud Deployment with P Colonies</td>
<td>70</td>
</tr>
<tr>
<td>Napsugár F. Gáti, Attila Kiss : Sound classification with transfer learning</td>
<td>72</td>
</tr>
<tr>
<td>Eduard Ştefan Grigoriec : On some subclasses of holomorphic functions whose derivative has positive real part</td>
<td>73</td>
</tr>
<tr>
<td>Hristina Gulabovska, Zoltán Porkoláb : Status of Static Analysis Tools for Detecting Python Security Vulnerabilities</td>
<td>74</td>
</tr>
<tr>
<td>Attila Gyén, Norbert Pataki : Discovering Shared Variables for Comprehension of Multi-threaded C++ Programs</td>
<td>76</td>
</tr>
<tr>
<td>Gábor Horváth, Norbert Pataki : Synthesizing Same-Language Summaries for Symbolic Execution</td>
<td>78</td>
</tr>
<tr>
<td>Ferenc Izsák, Béla J. Szekeres : Fractional operators in relativistic quantum mechanics: the square-root Klein–Gordon equation</td>
<td>80</td>
</tr>
<tr>
<td>Mohammed B. M. Kamel, Peter Ligeti, Christoph Reich : Region-Based Distributed Hash Table for Fog Computing Infrastructure</td>
<td>82</td>
</tr>
<tr>
<td>Attila Kántor, Attila Kiss, László Grad-Gyenge : Semantic Encoder Tasks for the Hungarian Language</td>
<td>84</td>
</tr>
<tr>
<td>Ambrus Kaposi, Norbert Luksa : A calculus of single substitutions for simple type theory</td>
<td>85</td>
</tr>
<tr>
<td>Meriem Kherbouché, Bálint Molnár : Modelling to Program in the case of Workflow Systems Theoretical background and literature review</td>
<td>87</td>
</tr>
<tr>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Melinda Kiss, Adrián Csizsárik, Ákos Matszangosz, Balázs Maga, Dániel Varga: Global Sinkhorn Autoencoder - Optimal transport on the latent representation of the full dataset</td>
<td>88</td>
</tr>
<tr>
<td>Attila Kiss, Ádám Kiss: The Compressed Program Dependence Graph</td>
<td>90</td>
</tr>
<tr>
<td>Smiljana Knezev, Gordana Rakić, Zoran Budimac, Melinda Tóth, István Bozó: Evaluation of a Recursion Aware Complexity Metric</td>
<td>92</td>
</tr>
<tr>
<td>Ábel Kocsis, Zoltán Gera, Melinda Tóth: Diagnosing vulnerabilities with static analysis in order to support code comprehension</td>
<td>98</td>
</tr>
<tr>
<td>Adrienn Koncz, Attila Gludovátsz: Indirect electricity consumption calculation in the product manufacturing</td>
<td>100</td>
</tr>
<tr>
<td>László Kopácsi: A Self-Supervised Method for Bodypart Segmentation and Keypoint Detection of Rat Images</td>
<td>102</td>
</tr>
<tr>
<td>Réka Kovács, Gábor Horváth, and Zoltán Porkoláb: Detecting lifetime errors of std::string_view objects in C++</td>
<td>104</td>
</tr>
<tr>
<td>Dániel Leskó, Máté Tejfél: Random program generation via Lambda terms</td>
<td>106</td>
</tr>
<tr>
<td>Jianhao Li, Viktória Zsók: Distributed Storage Pattern</td>
<td>107</td>
</tr>
<tr>
<td>Levente Lőczi: Real-time web-based visualization of the Radon transform</td>
<td>109</td>
</tr>
<tr>
<td>Dániel Lukács, Gergely Pongrácz, Máté Tejfél: Extracting deep control flow graphs from P4 syntax trees</td>
<td>110</td>
</tr>
<tr>
<td>Tamás Lukovszki, Péter Vadász: Randomized dispersion of mobile robots – theory and experiments</td>
<td>113</td>
</tr>
<tr>
<td>Chuangtao Ma, Bálint Molnár: Semantic Consistency behind Ontology Learning and Mapping for Heterogeneous Data Integration</td>
<td>115</td>
</tr>
<tr>
<td>Zsigmond Márási, Bálint Molnár: Representation of Product Catalog in a Multifaceted Demand Structure by Hypergraphs</td>
<td>117</td>
</tr>
<tr>
<td>Gábor Maros: Quadratures for the numerical solution of fractional differential equations</td>
<td>118</td>
</tr>
<tr>
<td>Virgilinus-Aurelian Minutá: Group graded Morita equivalences induced by wreath products</td>
<td>119</td>
</tr>
<tr>
<td>Mádálina Moga: Some properties of the fixed point equation with Meir-Keeler operator</td>
<td>120</td>
</tr>
<tr>
<td>Gregory Morse, Tamás Kozsik: Fully Dynamic Strong Connectivity and Reachability in Digraphs</td>
<td>121</td>
</tr>
<tr>
<td>Mirtill Boglárka Naghi, David Iclanzan: Computational Evolutionary Perception: a Study on Scalability</td>
<td>123</td>
</tr>
<tr>
<td>Zsuzsanna Nagy-Csiaha, Margit Pap: A representation of quaternionic Blaschke group</td>
<td>125</td>
</tr>
<tr>
<td>Gergely Nagy, Zoltán Porkoláb: Correctness of a High-level RCU Implementation</td>
<td>126</td>
</tr>
<tr>
<td>Zsolt Németh, Gergely Nagy: Color image analysis and recognition using orthogonal quaternion Zernike moments</td>
<td>127</td>
</tr>
<tr>
<td>László Németh, András Zempléni: A new bootstrap resampling scheme for INAR processes with trend</td>
<td>129</td>
</tr>
</tbody>
</table>
Boróka Oltéan-Péter : Comparing epidemiological models with the help of visualization dashboards ........................................ 130
Alexandru Orzan, Nicolae Popovici : On a special class of fractional type set-valued functions ........................................ 131
Zsolt Páles : Existence and uniqueness theorems for ordinary differential equations with mixed initial and boundary value conditions ........................................ 132
Tamás Pál, Bálint Molnár, Ádám Tarcsi : Lightweight, length invariant models and dimensionality reduction in respiratory disease detection ........................................ 133
Ioan Papuc : On a Dirichlet problem for the Darcy-Forchheimer-Brinkman system with application to lid-driven porous cavity flow with internal square block ........ 134
Oana Părva, Daniel Breaz : Univalence properties of an integral operator ........ 135
Y. Paul, S. Fridli : Sleep states detection using Halfwave and Franklin transformation .......................... 137
Manuela Petrescu : Digitalization process of public services. Romania’s example ........ 139
Csaba Pintér, Ábel Fóthi : Community structures of autism genes on functional interaction graphs support the omogenic background of complex neurodevelopmental disorders ........................................ 149
Heinz-Joachim Rack, Robert Vajda : Explicit solution, for \( n = 7 \), to a Markov-type extremal problem initiated by Schur ........................................ 151
Simona-Maria Radu : Transfer of (dual) CS-Rickart properties via functors between abelian categories ........................................ 153
Dániel Balázs Rátaí, Zoltán Horváth, Zoltán Porkoláb, Melinda Tóth : Cell-oriented Programming ........................................ 154
Anna Reale, Benedek Kovács, Melinda Tóth, Zoltán Horváth : Services for an Edge-native application ........................................ 156
Ádám Révész, Norbert Pataki : A Language and Its Compiler for Programming Serverless Applications ........................................ 158
Zsombor Rigmányi, Zsolt Kovács, David Iclanzan : Optimal Scale for the Classification of Immunohistochemically Stained Colorectal Carcinoma Samples ........................................ 159
Gábor Román : Primality Proofs with Elliptic Curves: Probabilistic Factorization .... 161
Bence Rozner : Stochastic processes on random graphs with multiple type edges ........ 162
Nikolai Ryzhkov, Ildikó László : Stability of triangularization of polynomial matrices ........ 164
Pramod Kumar Sethy, Erzsébet Csuhaj-Varjú : Networks of Evolutionary Processors with Length-based Communication ........................................ 166
Levente Simon, Anna Soós : Coupled fixed point theorem and fractals on mixed patterns 168
Sabina Surdu : Machine Learning-Powered Data Exploration Using SQL Queries - the Negation Query in a New Light ........................................ 170
Sándor Szabó : Scheduling with time restriction and clique search ........................................ 172
Róbert Szabó, Dániel Katona, Mártón Csillag, Adrián Csiszárík, Dániel Varga : Visualizing Transfer Learning ........................................ 173
Richárd Szalay, Zoltán Porkoláb : Applying Modules for Modern C++ Libraries ........ 178
Richárd Szalay, Ábel Sinkovics, Zoltán Porkoláb : In situ Enhancement of Type Safety using Fictive Types ........................................ 179
Bálint Takács, Róbert Horváth, István Faragó: Numerical methods for space-dependent SIR models with constant delay .................................................. 181

Gabriella Tóth, Máté Tejel: Error detection and analysis of packet processing paths of P4 programs ................................................................. 182

Viktória Tóth, Robin Kiss: Real-life applications of pseudorandom generators ...... 185

László Tóth: Another generalization of Euler’s arithmetic function and of Menon’s identity ........................................................................ 187

Radu Truşcă: Local fixed point theorems and open mapping principles for generalized contractions ............................................................... 188

M. J. Uray: A family of barely expansive polynomials .................................. 189

Balázs Varga, István Bozó, Melinda Tóth: Refactoring concurrent Erlang applications for distribution ......................................................... 191

Viktor Varga: Label Propagation with Graph Neural Networks in Interactive Video Segmentation Annotation ................................................. 193

Márton Véges: Scale-free Loss for Absolute Pose Estimation ......................... 195

Anita Windisch, Péter L. Simon: The dynamics of the Hopfield model for homogeneous weight matrix ......................................................... 197

Yuping Yan, Peter Ligeti: Anonymization Techniques in Social Networks ........... 199
Invited Talks
Ulam-Hyers stability of some second order differential equations defining special functions

Szilárd András
Department of Mathematics and Computer Science of the Hungarian Study Line,
Babeș-Bolyai University, Cluj Napoca
andrasz@math.ubbcluj.ro

In the last fifteen years several results were published about the Ulam-Hyers stability of functions, ordinary differential equations, integral equations, partial differential equations, stochastic differential equations and several $\delta$-fictions were also developed such as Ulam-Hyers-Rassias stability or $\delta$-ficed Ulam-Hyers-Rassias stability.

The history of Ulam-Hyers (and of many $\delta$-ficed Ulam-Hyers type) stability started in 1940, when S.M. Ulam posed the question of stability of group homeomorphisms: Let $G_1$ be a group and let $G_2$ be a metric group with the metric $d(\cdot, \cdot)$. Given $\varepsilon > 0$ does there exists a $\delta > 0$, such that if a function $h : G_1 \rightarrow G_2$ satisfies the inequality

$$d(h(xy), h(x)h(y)) < \delta, \forall x, y \in G_1,$$

then there exists a homomorphism $H : G_1 \rightarrow G_2$ with $d(h(x), H(x)) < \varepsilon, \forall x \in G_1$? One year later D.H. Hyers solved the problem under the assumption that $G_1$ and $G_2$ are Banach spaces. He proved that each solution of the inequality

$$\|f(x+y) - f(x) - f(y)\| \leq \varepsilon, \forall x, y \in G_1$$

can be approximated by an exact solution, an additive function. In this case, the Cauchy additive functional equation

$$f(x+y) = f(x) + f(y)$$

is said to have the Ulam-Hyers stability. Since then the initial question was extensively $\delta$-ficed to other types of equations. The main question is always if near an approximated solution of a given equation we can find an exact solution such as the distance between the exact solution and the approximated solution is somehow controlled using the error produced by the approximated solution. Of course this formulation needs to be clarified in terms of the mathematical context we use.

The special functions such as Chebyshev’s polynomials, Legendre polynomials, Bessel functions, $\delta$-ficed Bessel functions, Laguerre polynomials, $2$ functions are widely used in applications, so the Ulam-Hyers stability of the differential equations used for defining these functions is also relevant. In this paper we extend and unify some results published in [1], [2], [3], [4], [5], [6], [8]. Our method is based on a series of observations published in [7] and the results are related to some earlier results published in [9].

References


Operator splittings and their applications

István Faragó
Department of Applied Analysis, Eötvös Loránd University, Hungary
Department of Differential Equations, Budapest University of Technology and Economics, Hungary
faragois@caesar.elte.hu

In the modelling of complex time-depending physical phenomena the simultaneous effect of several different sub-processes has to be described. The operators describing the sub-processes are as a rule simpler than the whole spatial differential operator. Operator splitting is a widely used procedure in numerical solution of such problems. The point in operator splitting is the replacement of the original model with one in which appropriately chosen groups of the sub-processes, described by the model, take place successively in time. This de-coupling procedure allows us to solve a few simpler problems instead of the whole one.

In the talk several splitting methods will be constructed (sequential splitting, Strang splitting, weighted splitting, additiv splitting, iterated splitting). We discuss the accuracy (local splitting error) of the methods. The stability and the convergence will be also discussed. We also examine the effect of the choice of the numerical method chosen to the numerical solution of the sub-problems in the splitting procedure.

We list the main benefits and drawbacks of this approach.
Human Machine Intelligent Interaction: Has time arrived?

András Lőrinz
Department of Artificial Intelligence, Faculty of Informatics
Eötvös Loránd University, Budapest
lorincz@inf.elte.hu

I will argue that YES, time has arrived for intelligent-like interaction between humans and machines. I will start with a putative example showing that the machine may be able to provide help in a tricky and dangerous driving situation. Given the example, I will list the constraints of a real applications together with recent changes in the world and in my group. These changes concern advances in deep learning software and hardware technologies and constraints from GDPR. I will argue that the human-centered HumanE-AI EU ICT-48 Flagship project comes just-in-time for us.
Automatic Person Identification based on Physical Activity Patterns Recognition

Dan Mircea Suciu¹, Dănut Ilisei²
¹Babes-Bolyai University, Cluj-Napoca, Romania
²Technical University of Munich
¹dsuciu@cs.ubbcluj.ro
²danut.ilisei@tum.de

The ability to provide a proper system that can give accurate information about people’s activity and identity could be a life-changing one. Our talk is about a study on the feasibility of such a framework. Our experiments are based on a public dataset that contains information about the motion activities of 24 participants of different genders and ages while carrying a smartphone, with embedded inertial sensors, kept in the participant’s front pocket. We followed two research directions: the first direction concerns the recognition of the physical activities and the second one concerns the recognition of the participant. The studied activities in this work are walking downstairs, walking upstairs, walking, jogging, sitting, and standing.

References

Regular Contributions
Some optimization possibilities in data plane programming

Gereltsetseg Altangerel, Máté Tejfel
3in Research Group, Martonvásár, Eötvös Loránd University, Budapest
gereltsetseg@inf.elte.hu, matej@inf.elte.hu

Software-Defined Networking (SDN) technology aims to create a highly flexible network by decoupling the control plane and data plane and programming them independently. There has been a lot of research on improving and optimizing the control plane, and data plane programming is a relatively new concept, so study on it is one of the hot topics for researchers. Recently, domain-specific programming languages with simple syntaxes and semantics have emerged to describe packet processing behavior (data plane) of programmable forwarding elements. Researches in data plane programming are often around them and generally classified as follows.

1. The data plane programming languages (P4, Openflow, Protocol-Oblivious Forwarding, Domino) themselves are still under development, so there is a need to improve and optimize them in terms of syntax, semantics, language construct, application programming interface(API) or in terms of optimally extending the language through external functions/libraries [1],[2].
2. Since the network provides a wide range of services, it is not easy to create a single optimized, programmable switch or network solution, so experiments have conducted to create application-based software switches that are optimal for a particular service [3].
3. However, it is also important to look for opportunities to develop general optimization methods that can be addressed in data plane programming to improve network performance, flexibility, scalability, security, bandwidth utilization, and reliability [4].
4. Optimally implementing/creating old/new network protocols and data plane functions and determine where to deploy them more effectively. (Whether to deploy them on a network device itself, on an external device, or at which level of TCP / IP protocol suite) [5].
5. Optimally deploying stateful packet processing in software switch [6].
6. Building software switch with high-performance input/output (I/O) through external frameworks/libraries for fast packet processing such as DPDK, and netmap [7].
7. Optimizing data plane application with Network functions virtualization (NFV) [8].
8. Monitoring and auditing data plane application performance.

At the 2019 Dagstuhl Seminar, the well-known scientists on computer networking discussed challenges and problems in the field of data plane programming that need to be addressed over the next 10 years. Based on this seminar and above research directions, we suggested some possible solutions for optimizing data plane applications to improve packet processing performance, link utilization, and enrich data plane language. These suggestions are:

**Optimization 1: Enriching data plane language with asynchronous external function:** Network functions process packets based on header information or payload. Payload processing functions require higher performance. If these functions are not run asynchronously, performance efficiency may be limited. Most data plane languages do not specify how to integrate an asynchronous external function into a packet processing pipeline. P4, one of the emerging data plane language, standardizes the use of an extern construct to add an external function, but asynchronous execution is unsupported. Therefore, an efficient, simple, and general way to add an asynchronous external function to a data plane language needs to be developed.

**Optimization 2: Compression based on payload size:** We implemented the optimization 1 as an example of an asynchronous compression function. Packets are compressed to improve the link utilization, but compression of all packets increases the computational load of the device, so
criteria can be set as to which packet is best to compress. Based on the studies, the following criteria can be set.

(a) Payload size: Packets with a larger payload (more than 500 B) produce a better compression ratio [9].

(b) Waiting time in queue: If an intermediate node receives a packet with a larger payload, that node will calculate the waiting time of the packet in the queue. If the waiting time is greater than a certain instantaneously calculated threshold, the node will compress the packet’s payload.

Optimization 3: In-network caching: The cache is used to make packet processing decisions faster. Furthermore, in-network (network device itself) caching functionality has been shown to run at line rate on programmable switches. The caching can also be used in our case study (optimization 1). Our case study organized as follows: when a packet comes into software switch, it is processed through main loop (for example, the main loop defines exit interface), then this loop saves the context of the packet, and sends this packet into compression loop. Once the packet is compressed, it will return to the main loop, resume the context, and pass the packet to the defined interface. If the context was cached in memory to be reused to process the next packet of one flow, the packet processing can be faster.

Optimization 4: Offloading external functions to an additional thread, VM or server, etc. In our case study (optimization 1), main packet processing loop and compression function are running different thread. In the future, to run an external function that requires a lot of computation, we can run that external function on an additional thread or some external devices.

We have some prototype implementation for the optimization 1, and optimization 2 (a) on P4, but there is currently no partial analysis of efficiency based on actual measurements. In the future, we are planning to make some measurements, and since all of these ideas are interrelated and complementary, we will implement the other two ideas based on our prototype implementation.

Acknowledgement The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

References


The Role of Implementation-specific Static Analysis

Bence Babati, Norbert Pataki
ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Informatics, 3in Research Group, Martonvásár, Hungary
babati@caesar.elte.hu, patakino@elte.hu

The Standard Template Library (STL) is one of the most frequently used libraries in C++. It follows generic programming paradigm by using templates. It is highly configurable with a lot of functions, majority of the C++ software artifacts uses it.

STL provides containers (e.g. vector) which can store a collection of elements and manage the underlying memory allocations. STL provides algorithms (e.g. find_if) which can operate on range of elements and are independent of the underlying container. Iterators bridge the gap between containers and algorithms and allow the parallel extension of the set of containers and the set of algorithms. Functors or function objects enable the parametrization of library with user-defined code snippets without significant overhead [5].

The usage of STL improves the readability, efficiency and the whole quality of the program. It enables to reduce the number of classical programming errors. However, misuses can lead new kind of errors.

The STL usage is partially validated at compile time, although, many properties are not checked by the library or the compiler. So, many issues can appear in C++ programs related to STL usage.

Validations in the compiler cannot cover all of the possible issues. For instance, iterator invalidation cannot be detected by the compilers. In many cases, static analysis tools may validate the usage of STL [2]. These tools typically work in implementation-independent manner. Moreover, portability problems require special tools [1].

In this paper, we investigate how the library-specific static analysis may result in more precise feedback. We present a special STL-related problem which cannot be detected by the compilers. The statefulness of algorithm predicates could cause strange error, but this property is not validated at all [3]. However, the effect of this issue highly depends on the implementation of STL [4].

For instance, the following functor class can be compiled but its usage may result in unexpected behavior under specific circumstances.

class NthNumber
{
public:
    bool operator()( const int x )
    {
        ++timesCalled;
        return timesCalled == 3;
    }

private:

    std::size_t timesCalled = 0;
};

1The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).
We demonstrate our new static analysis tool to analyze predicate functors. This tool is based on the Clang compiler infrastructure. The goal is to detect stateful predicates when it can cause unexpected issue with the used algorithm. We evaluate our tool with open-source applications and inspect how the library-specific approach decreases the number of false positive findings.

References


Epidemic spread on random graphs with multiple type nodes

Ágnes Backhausz, Edit Bognár
Department of Probability Theory and Statistics, Faculty of Science, ELTE Eötvös Loránd University, Budapest, Hungary
agnes.backhausz@ttk.elte.hu, bognaredith@gmail.com

Modelling epidemic spread has been an important problem for which several approaches have been developed in the last decades, such as differential equations, random graphs, branching processes or other stochastic interacting particle systems. The variety of mathematical tools immediately raises several questions about the respective advantages of the different methods. In order to address a particular aspect of this problem, in the current paper we examine the effect of the structure of the underlying graph together with the consequences of the fact that vertices can belong to different types, e.g. according to their age or their willingness to observe certain rules. We introduce several random graph models with multiple type nodes and investigate the influence of general social distancing restrictions, self-isolation strategies and vaccination. We compare the different graph models with each other and with a simpler compartment model as well.

The accurate modelling, regulating or preventing of a possible epidemic is still a difficult problem of the 21st century. As of the time of writing, a novel strain of coronavirus (SARS-CoV-2) has spread to almost all countries of the world from China and has infected more than 20 million people, although authorities have been taking serious actions to prevent a worldwide outbreak. As for mathematical modelling, choosing the scale on which we treat the whole process is a key point. Differential equations can give very good approximations of real models when the size of the population and the frequency of relationships through which infections can happen are both large. However, in other situations it might be crucial to understand more of the underlying structures, and use agent-based models or a combination of the two.

In this paper we consider random graphs with multiple type vertices from the point of view of isolation and vaccination. In both cases, the age of the vertices are part of their types, and edge densities between different age groups are given. In order to be able to compare our random graph models to the results of compartment models using differential equations, we use the same data as [2]. Given the edge density, we examine different random graph structures such as the Erdős–Rényi random graph model with communities (the stochastic block model), a multitype configuration model or preferential attachment model. The single-type cases of the latter models are widely used [1, 3], here we will define multitype versions, where the randomization of the graph and the choice of types might be closely related. For modelling the spread of the disease, we have chosen a SEIR (susceptible–exposed–infectious–recovered) model with random times spent at different stages.

In the first part of the paper for modelling isolation, we use another characteristic of vertices, representing their consciousness about the virus and thus participating in self-isolation when experiencing symptoms, as well as their willingness to stay in quarantine when some of their contacts gets infected. The fraction of conscious individuals in a population can be increased by raising awareness of the disease. By screening infected individuals and mapping their contacts it is also possible to increase rate of disease-suspicious individuals staying in a quarantine. We examined the process with these two different means and proportion of individuals participating in self-isolation. We run the process with different parameter sets on Configuration model random graphs of different degree distribution (Poisson, Pareto, Geometric) and we confronted overall attack rates and maximum numbers of individuals being infected at the same time. One can conclude that both parameters reduce the intensity of the virus spread, however increasing consciousness of individuals can flatten the curve more effectively, since many infected cases can be traced by en
masse testing. We also study the effect of general restriction measurements (e.g. closing schools or switching to home office). We run the process (with the same transmission rate) not only on a fixed random graph, but on 5 graphs, from which 4 can be obtained as a subgraph of the first one by deleting edges corresponding to decreasing density. This is the equivalent graph representation of restriction measurements applied to individuals. We confronted outcomes of the process on Configuration model graphs by defining 3 different sets of restriction and releasing actions, from which two are executed at predefined time steps regardless of the actual state of the virus spread. In the third case, actions are taken according to the number of infectious individuals, in which reaching a certain threshold at a given day leads to a specific action. One can conclude that aligned actions taken at the right time could not only flatten the curve, but also allow the use of less severe restrictions. However, we still can experience difference in volume between graphs of different degree distributions. Thus not only number of infectious individuals should be taken into consideration but also the steep of the curve, which can be measured by the Malthusian parameter of the disease.

In the second part of the paper, we assume a pre-existing vaccine and compare different vaccination strategies started at the outbreak of the virus, or coming a few days before the outbreak in these multitype random graphs, which can be based on the type and the degree of the vertex as well. Here we investigate if there is any effect of the underlying graph structure more than degree distribution. We compared preferential attachment and Erdős–Rényi random graphs with configuration models, where the degree distribution can be tuned but the random choice of the graph is different. The dynamics of the disease are also effected by a vaccination, thus we are also searching for the main similarities and differences between differential equation models and random graph models when we consider several age groups. We set the degree distribution of Configuration Model graphs to have as small variance as possible to model differential equations. Firstly, we confront outcomes on different random graphs and the numerical solutions of the differential equations from [2]. Then we examine the sensitivity of the model for certain parameters, such as basic reproduction number $R_0$ corresponding to the seriousness of the disease, and set against results on random graphs of different types. Later we pursue finding the best vaccination strategy if both the graph structure and the types of vertices are taken into account. We investigate how different strategies in vaccination can affect the attack rates. Three very different strategies (Uniform strategy, Contacts strategy and Degree strategy) based on age groups or other properties of the graph are modelled, to study how can the whole population benefit from the prioritization of vaccination of certain groups. On both of the examined graph structures not surprisingly vaccination of individuals with the highest number of contacts was proven to be significantly the most effective preventative measurement (especially in case of preferential properties). However, in a real underlying network determining the exact number of edges of a node is rather challenging, particularly with actions taken against the spread of the disease. If a certain disease affects individuals of different age similarly, then vaccination of age groups by contact matrix can also reduce severity of the disease and overall attack rates.

The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

References


In our previous works we have introduced several models of digraphs to represent them as a SAT problem [1]. We work with special digraphs, called communication graphs, where nodes are Boolean variables, or labeled by different Boolean variables, and there is no self loop. This restrictions are very easy ones. We have also worked with generating minimal unsatisfiable SAT instances [2].

In logic the most natural representation of an edge of a directed graph, say $a \rightarrow b$, is to use implication, i.e., $a \implies b$, i.e., the edge $a \rightarrow b$ can be represented by the binary clause: $(\neg a \lor b)$. If a graph contains two edges: $a \rightarrow b$, and $a \rightarrow c$, then those can be represented by the formula: $(a \implies b) \land (a \implies c)$, which is equivalent to two 2-clauses: $(\neg a \lor b) \land (\neg a \lor c)$. We call this as the Strong Model (SM) of directed graphs [3, 4]. This was our first model. It generates always a 2-SAT representation, so it is not very interesting from the viewpoint of Boolean Satisfiability.

Our second model was the Weak Model (WM) [4]. Its idea is the following: If a graph contains two edges: $a \rightarrow b$, and $a \rightarrow c$, then those can be represented by the formula: $(a \implies b) \lor (a \implies c)$, which is equivalent to a 3-clause $(\neg a \lor b \lor c)$. We need to represent cycles of the graph, too. If $a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_n \rightarrow a_1$ is a cycle with exit points $b_1, b_2, \ldots, b_m$, then this cycle can be represented by the clause: $(\neg a_1 \lor \neg a_2 \lor \cdots \lor \neg a_n \lor b_1 \lor b_2 \lor \cdots \lor b_m)$. It generates always a 3-SAT problem, so it is interesting from the viewpoint of Boolean Satisfiability. But on the other hand, we need to find all cycles, even the non-simple ones, which makes this model unpractical.

We present a model generator which generates SAT problems from digraphs.

Our third model was the Balatonboglár Model (BB) model. It uses the trick that instead of detecting each cycle, it generates from each path $a \rightarrow b \rightarrow c$ the following 3-clause: $(\neg a \lor \neg b \lor c)$, which is a Negative-Negative-Positive (NNP) shaped clause, even if there is no cycle which contains the vertices $a$ and $b$. This simplification allows very fast 3-SAT problem generation from a directed graph, and the SAT instance will be a Black-and-White 3-SAT if and only if the input directed graph is strongly connected [1]. This trick generates lots of superfluous clauses.

To overcome this problem, we introduced the Simplified Balatonboglár Model (SBB) [5]. In this model we consider the strongly connected components (SCC) of the directed graph [6, 7].

We have implemented a special SAT solver for this type, called BaW 1.0. This solves the problems generated by the strong model in linear time [8]. In this work we generalize this solver to solve the models generated by our BB and SBB models. The new solver is called BaW 2.0.

We have implemented a special SAT solver for this type, called BaW 1.0. This solves the problems generated by the strong model in linear time [8]. In this work we generalize this solver to solve the models generated by our BB and SBB models. The new solver is called BaW 2.0.

We have implemented a special SAT solver for this type, called BaW 1.0. This solves the problems generated by the strong model in linear time [8]. In this work we generalize this solver to solve the models generated by our BB and SBB models. The new solver is called BaW 2.0.
it creates 4 branches by propagating the following binary assignments: \{a, b\}, \{a, \neg b\}, \{\neg a, b\}, \{\neg a, \neg b\}, respectively. In the theoretical part we show that in case of BB and SBB the first 3 branches we will have units till we create the empty clause, or the input is satisfiable. In the last branch we do the same recursively till we have a 2-cycle. If we run out of 2-cycles then there will be lots of binary clauses.

In this case we start a different method which relies on binary clauses. We branch on the variable which occurs as a positive and also as a negative literal in the binary clauses. It is easy to see that it provides new units on each branch. Then we do BCP till we create an empty clause, or the clause set becomes empty or we run out of units. In this case we call this method recursively.

In the theoretical part we prove the following lemmas and theorems: If the clause set is a BaW problem, then its \textit{NPP}-Graph contains at least one cycle. If the clause set is generated by BB or SBB, then it \textit{NPP}-Graph contains at least one 2-cycle. If \textit{S} is clause set and its \textit{NPP}-Graph is a strong digraph then \textit{S} is not necessary a BaW SAT instance. If \textit{S} is clause set and its \textit{NNP}-Graph is a strong digraph then \textit{S} is not necessary a BaW SAT instance. If \textit{S} is clause set and its \textit{NPP}-Graph is strong and its \textit{NNP}-Graph is strong, then \textit{S} is not necessary a BaW SAT instance. If \textit{S} is clause set generated by BB or SBB, then BCP(UP(UP(\textit{S}, x), y)) is trivially UNSAT if \textit{S} is UNSAT, or it is trivially SAT if \textit{S} is SAT, where \((x, y) \in \{(a, b), (a, \neg b), (\neg a, b)\}\), and \((a, b)\) is a 2-cycle in \textit{S}.

We have tested BaW 2.0 on several instances. We found out that it solves BB and SBB instances faster than state-of-the-art solvers. Although BaW 1.0 is linear on SM instances, BaW 2.0 is also linear on such problems, but polynomial on BB and SBB instances.

We still study how to solve the WM instances with a similar special SAT solver.

References


In the era of the Internet there is a continuous demand for distributed systems which should serve thousands of requests on a daily basis. But with this growing demand companies have to face a growing number of cyber threats as well which can not only harm their customers in the form of data theft or data loss but their own reputation as well. In order to improve the security of the systems [1], there are several standards (CERT, OWASP’s Application Security Verification Standard) and static analyser tools (CodeChecker, SpotBugs, SonarQube, Fortify) to achieve this goal. Unfortunately, these tools only cover popular programming languages like C++, Java, Python or TypeScript.

Certainly a lot has changed since Erlang [2] was initially introduced in the 80s. For any language to remain relevant it is a necessity to adjust to these new requirements imposed by the industry. It is important to try to close this gap that currently exists in the domain of static security analysers in order to keep Erlang as relevant as it is today and to possibly increase its popularity in the programming industry.

The change of the original concept of running Erlang on protected hosts, and the human factor (e.g. inexperienced programmers, the lack of knowledge about a specific application) of the development process makes it necessary to consider secure coding during the development of Erlang applications as well [3, 4, 5, 6].

The main contributions of this paper is providing methodology and tool for the Erlang community to help identifying security vulnerabilities [9, 10] like OS injection, cryptography or atom exhaustion related attacks in an early phase of the development process, therefore improving the security level of their Erlang applications. We examined and categorized the potential vulnerabilities specific to Erlang and based on these categories we defined checkers to identify the source code fragments that violates the security rules. The checkers were defined using a set of static source code analyses (e.g. data-flow, call-graph information) provided by RefactorErl [7, 8].

References


---

3The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002). The research is part of the “Application Domain Specific Highly Reliable IT Solutions” project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme.


[9] Brigitta Baranyai: Funkcionális nyelvek és a statikus kódelemzéssel támogatott biztonságos szoftverfejlesztés, Paper at the Student Association Conference, Faculty of Informatics, Eötvös Loránd University, May 2020, Received 2nd prize.

Multiple Richardson Extrapolation and its Combination with the Implicit Euler Method

Teshome Bayleyegn\textsuperscript{1}, Ágnes Havasi\textsuperscript{2}

\textsuperscript{1}Eötvös Loránd University, 1117 Budapest, Pázmány Péter s. 1/C, Hungary
\textsuperscript{2}Eötvös Loránd University and MTA-ELTE Numerical Analysis and Large Networks Research Group, 1117 Budapest, Pázmány Péter s. 1/C, Hungary

\textsuperscript{1}sbayleyegn130@gmail.com
\textsuperscript{2}havasiagnes@caesar.elte.hu

Classical Richardson extrapolation is a sequence acceleration technique, used to enhance the accuracy of any convergent numerical method for ordinary differential equations \cite{2,3}. It enhances the order of the underlying numerical method by one if the exact solution is sufficiently smooth. We investigate a generalized version of the classical Richardson extrapolation, called multiple Richardson extrapolation (MRE), where Richardson extrapolation is applied to the combination of some underlying method and the classical Richardson extrapolation.

Earlier we analysed the MRE for accuracy (it increases the order of the underlying method from \( p \) to \( p + 2 \)) and absolute stability when combined with explicit Runge–Kutta methods \cite{1}. In this paper we investigate the same issues for the case where the underlying numerical method is the implicit Euler method, widely used for stiff problems due to its excellent absolute stability properties.

It is shown that even if the implicit Euler method itself is A-stable, just like its combination with the classical Richardson extrapolation, the multiple Richardson extrapolation spoils this favourable property. Numerical examples are presented to illustrate that stability problems are to be expected when the product of the time step and any of the eigenvalues of the matrix (or Jacobian matrix) of the system of ordinary differential equations falls into a narrow stripe close to and along the imaginary axis. The intervals of instability along the imaginary axis are studied also for the cases where higher order versions of MRE (2MRE, 3MRE and 4MRE) are used. Comparisons with the corresponding repeated Richardson extrapolation methods (RRE, 2TRRE, 3TRRE, 4TRRE) are also given.

References


Defining vertex types and metric for graph-based Information System models

András Béleczki, Bálint Molnár

Information Systems Department, Eötvös Loránd University of Budapest, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
(bearaaai, molnarba)@inf.elte.hu

Abstract

In many previous studies, we explained the necessities of modeling tools, which are capable of depicting heterogeneous information system models, such as Business Process Models, Data Flow Diagrams, Logical Data Models, in a homogeneous environment. To do so we proposed the usage of our hypergraph modeling which is based on the hypergraph theory and it is extended for general document modeling in an enterprise architecture domain. After realizing the flexibility of these models we also proposed to use it in other domains, such as the Internet of Things, where a multilevel architecture (data lake, fog, and cloud) model can be designed and evaluated with proper graph node definition and metrics. In this paper, we focus on these two: what granularity should we use for the vertex definitions which will provide enough information for the evaluation but not too much to overweight the complexity of the model; and what kind of metrics should be applied for different calculations.

Keywords IoT, hypergraph, verification, transformation, Reliability of Information Systems.

The used sources: [1], [2], [3]

Acknowledgement

The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001), (EFOP-3.6.3-VEKOP-16-2017-00002) and no. ED_18-1-2019-0030 (Application-specific highly reliable IT solutions) program of the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme.

References


Lloyd’s clustering method is not 1-separability detecting

Katalin Bene, László Szabó
Department of Algorithms and Their Applications
Eötvös Loránd University
szabolaszlo@inf.elte.hu

Clustering is a fundamental tool for data analysis. Its goal is natural: to identify groups of similar items within data. While the goal of clustering is simple, formalizing this task is much more challenging. It is well known that most of the common clustering objectives are NP-hard to optimize. In practice, however, clustering is being routinely carried out. One approach for providing theoretical understanding of this seeming discrepancy is to introduce notions of clusterability that distinguish realistically interesting input data from worst-case data sets. Here we focus on one such notion.

We consider a space \((X,d)\) where \(X\) is a set of data elements and \(d\) is a distance function on \(X\). It is assumed that \(d\) is symmetric and non-negative, and \(d(x,x) = 0\) for all \(x \in X\). For an integer \(k \geq 1\), a \(k\)-clustering of \(X\) is a partition \(\mathcal{C} = \{C_1, \ldots, C_k\}\) of \(X\) into \(k\) disjoint non-empty sets. For a \(k\)-clustering \(\mathcal{C}\) of \(X\) and data elements \(x_1, x_2 \in X\), we write \(i_1 \sim_C x_2\) if \(i_1\) and \(i_2\) belong to the same cluster in \(\mathcal{C}\), and \(i_1 \not\sim_C x_2\) otherwise. A \(k\)-clustering \(\mathcal{C}\) of \(X\) is called 1-separable if for any \(i_1, x_2, x_3, x_4 \in X\) such that \(i_1 \sim_C x_2\) and \(i_3 \not\sim_C x_4\) the inequality \(d(x_1, x_2) < d(x_3, x_4)\) holds. Note that the 1-separable \(k\)-clustering of any given data set is unique, if it exists.

One of the most widely used algorithm for clustering is Lloyd’s method. For a given data set \(X\) and initial center set \(S\) in the \(n\)-dimensional Euclidean space Lloyd’s method performs the following steps until two consecutive iterations return the same clustering: (1) assign each point in \(X\) to its closest element of \(S\), (2) replace \(S\) with the set of the centers of gravity of data elements assigned to each cluster.

A common initialization for Lloyd’s method is to select \(k\) random centers from the input data set. Another well-known initialization method is the so-called furthest-centroid initialization. Using this method, given a set \(X\), the initial centers \(c_1, c_2, \ldots, c_k\) in \(S\) are chosen as follows: center \(c_1\) is an arbitrary point in \(X\), then, for each \(i = 2, 3, \ldots, k\), center \(c_i\) is set to be the point in \(X\) that maximizes the distance from the set of the other centers that were already chosen.

Margareta Ackerman, Shai Ben-David, David Loker and Sivan Sabato stated in [1], as Lemma 6.4, that Lloyd’s clustering method with furthest centroid initialization always terminates with a 1-separable clustering if there is such a clustering of the data set.

In this short note we show that this is not true by constructing a data set in the plane with a 1-separable 2-clustering such that Lloyd’s method doesn’t terminate with this clustering regardless of the initialization method.

Consider the data set \(X\) consisting of the 12 points

\[ x_i = (-5 + i, 45) \text{ for } i = 1, 2, \ldots, 9, \]

and

\[ x_{10} = (0, 0), \quad x_{11} = (-22, -40), \quad x_{12} = (22, -40) \]

in the plane. Now

\[ \mathcal{C} = \{\{x_1, x_2, \ldots, x_9, x_{10}\}, \{x_{11}, x_{12}\}\} \]

is a 1-separable clustering of \(X\) since the within-cluster distances are

\[ d(x_i, x_j) \leq d(x_1, x_9) = 8 \text{ for } 1 \leq i < j \leq 9, \]

\[ d(x_1, x_{10}) \leq d(x_1, x_{10}) = d(x_9, x_{10}) = \sqrt{2041} = 45.17 \ldots \text{ for } i = 1, 2, \ldots, 9, \]
and 
\[ d(x_{11}, x_{12}) = 44, \]
while the between-cluster distances are
\[ d(x_i, x_k) \geq d(x_1, x_{11}) = d(x_9, x_{12}) = \sqrt{7549} = 86.88 \ldots \] for \( i = 1, 2, \ldots, 9 \) and \( k = 11, 12 \),
and
\[ d(x_{10}, x_{11}) = d(x_{10}, x_{12}) = \sqrt{2084} = 45.65 \ldots . \]

Run Lloyd’s method on \( X \) and suppose that it returns \( C \) in some iteration. Now the algorithm calculates the centers of gravity of the clusters which are \((0, 40.5)\) and \((0, -40)\) for \( \{x_1, x_2, \ldots, x_9, x_{10}\} \) and \( \{x_{11}, x_{12}\} \), respectively, and assigns each point in \( X \) to its closest center. In this way we obtain the clustering
\[ C' = \{\{x_1, x_2, \ldots, x_9\}, \{x_{10}, x_{11}, x_{12}\}\} \]
which is different from \( C \). The algorithm calculates again the centers of gravity of the clusters which are \((0, 45)\) and \((0, -80/3)\) for \( \{x_1, x_2, \ldots, x_9\} \) and \( \{x_{10}, x_{11}, x_{12}\} \), respectively, and assigns each point in \( X \) to its closest center. Now we obtain the the same clustering \( C' \) as before thus Lloyd’s method terminates with \( C' \).

This implies that Lloyd’s method running on \( X \) never terminates with \( C \) regardless of the initialization method.

This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

References

A Survey on Comparing Various Formal Semantics Definition Styles

Péter Bereczky, Dániel Horpácsi, Simon Thompson
Eötvös Loránd University, Department of Programming Languages and Compilers
berpeti@inf.elte.hu, daniel-h@elte.hu, S.J.Thompson@kent.ac.uk

Formal semantics provides rigorous, mathematically precise definition of programming languages, with which we can argue about program behaviour and program equivalence by formal means; in particular, we can describe and verify our argument with a proof assistant.

There are various approaches of giving formal semantics to programming languages, which apply different abstraction levels and mathematical machinery; the purpose of using the semantics determines which approach to go for. In this paper, we investigate some of the approaches that share their roots with the traditional relational big-step semantics, such as a) functional big-step semantics [7] (definitional interpreter [8]) d) pretty-big-step semantics [4] and e) traditional natural semantics [5]. We compare these approaches with respect to the following requirements in mind: executability of the semantics definition, property proof complexity (e.g. determinism) and conciseness of equivalence proofs expressed with it. We briefly discuss coinductive big-step semantics [6] too which enables reasoning about divergence.

We also present our case study language for comparing the semantics: a sequential subset of Core Erlang, a functional programming language, which is used in the intermediate steps of the Erlang/OTP compiler. For this simple functional language, we have already defined a relational big-step semantics with exception and side effect concepts [1, 2]. The aim with this current work is to benchmark our big-step definition with a variety of other semantics in different styles from the testability [3] and the applicability for refactoring verification point of view.

References


Investigation of some operator splitting methods

Lívia Boda

Department of Differential Equations, Budapest University of Technology and Economics, Hungary
bodalivi@gmail.com

Operator splitting is a widely and successfully used method in numerical analysis. It helps us when we have a very complicated Cauchy-problem, which we want to analyze. By using operator splitting, we get a series of easier Cauchy-problems which are linked through their initial conditions. By applying this method it is significantly easier to solve the problem of finding the numerical solution of the original problem.

The two most popular splitting methods are the sequential splitting (it is a first order method) and the Strang-Marchuk splitting method (it is a second order method). In this talk we analyze the relationship between these two methods, and analyze the following main question: how can we get from first order splitting methods to second order splitting methods. The answer is what we call "alternating" splitting. Furthermore, in the case of second-order methods, we also use the alternating splitting and analyze it.

References
Canonical expansion of integers for families of roofline polynomials

Dávid Bóka, Péter Burcsi
Department of Computer Algebra, ELTE Eötvös Loránd University, Hungary
bdavid1001@inf.elte.hu bupe@inf.elte.hu

We investigate the problem of representing integer polynomials with a restricted coefficient set up to congruence modulo a base polynomial. This is a widely investigated problem in the area of numeration systems. In our paper we examine a special type of base polynomials that we call roofline polynomials: the coefficients first increase, then stay constant, then decrease again (precise definitions are given below). We give necessary conditions for such a polynomial to be a so-called CNS polynomial. We also give results on the representation length when these conditions are met.

Let \( P = \sum_{i=0}^{n} p_i X^i \in \mathbb{Z}[X] \) be a monic integer polynomial with positive degree and \( |P_0| \geq 2 \). Let \( D = \{0, 1, \ldots, |P_0| - 1\} \) the set of digits, and let \( D[X] \) be the set of polynomials with coefficients in \( D \). We say that a polynomial \( A \in \mathbb{Z}[X] \) is canonically represented in base \( P \) by \( B \in D[X] \) if \( A \equiv B \pmod{P} \). If every polynomial has a canonical representative, then \( P \) is called a canonical number system polynomial (CNS polynomial).

A way to determine the canonical representative of a polynomial \( A \) is to reformulate the problem for matrix based numeration systems or shift radix systems as follows. Each residue class modulo \( P \) has a unique representative of degree at most \( d - 1 \), thus we can identify the residue classes with elements of \( \mathbb{Z}^d \). The operation \( A \mapsto X \cdot A \) taken modulo \( P \) is a linear map on this lattice. For each polynomial \( A = a_0 + a_1 x + \cdots + a_{m-1} x^{m-1} \) there is a unique \( d \in D \), for which there exists a \( C \) (also unique modulo \( P \)) such that \( A \equiv C \cdot X + d \pmod{P} \). This \( d \) is the constant term of the canonical representation, if it exists. If we use an appropriate basis for the vectors (see e.g. [1, 2]) the mapping \( A \mapsto C \) can be described as \( \tau = \tau_r : \mathbb{Z}^m \rightarrow \mathbb{Z}^m \),

\[
( z_0, \ldots, z_{d_1} ) \mapsto ( z_1, \ldots, z_{n-1} - \lfloor r z \rfloor ) ,
\]

where \( r = ( \ell_0, \ell_{p-1}, \ldots, \ell_0) / \ell_m \) and \( r z \) denotes the scalar product. The CNS property of \( P \) is equivalent to the property that \( \tau \) has a single periodic orbit 0, and that all orbits are eventually periodic.

It turns out that in the investigations below, it is sometimes convenient to consider a “lifted” representation of the dynamical system defined by \( P \), by which we mean the shift radix system associated to \((X - 1)P(X)\). Here, the original CNS property is equivalent to saying that each vector eventually maps to a vector \((a, a, \ldots, a)\).

The results we present are about families of polynomials we call roofline polynomials. We give necessary conditions for the existence of a canonical representative of \(-1\) (thus also for the CNS property), that in some cases, these are sufficient, and examine the length of the expansion, i.e. the degree of the canonical representative.

To decide the CNS property the orbit of each element of set of witnesses can be computed, but this computation can be hard, if the size of the set of witnesses is too large. For example, let \( p(x) \) be the polynomial

\[
x^8 + 2x^7 + 3x^6 + 3x^5 + 3x^4 + 3x^3 + 3x^2 + 3x + 2.
\]

The smallest set of witnesses of \( p \) contains 241719 elements, so this example indicate that the decision of CNS property can not be in polynomial time. Therefore we investigate a special family of polynomials which is called roofline polynomials - as mentioned above. The members of this family can be described by three parameters \( n, M \) and \( k \). The \( n \) denotes the degree of the polynomial, \( M \) is the maximal coefficient and \( k \) is a positive integer, such that \( k \) divides \( M - 2 \). We can construct the members of this family from these parameters as follows:

\[
f(n, M, k) = (2, 2 + k, 2 + 2k, \ldots, M - k, M, M, \ldots, M, M - 1, M - 1 - k, \ldots, 1 + k, 1)
\]
i.e. the constant term of these polynomials is 2, then the coefficients increase by $k$ to $M$, then constant, then decrease once by 1 and then decrease by $k$. The leading coefficient is 1, so the roofline polynomials are monic (it follows from the construction). For example the $f(12, 14, 3)$ is equal to

$$(2, 5, 8, 11, 14, 14, 14, 14, 13, 10, 7, 4, 1) \in \mathbb{Z}^{13}.$$  

**Theorem 1** Let $f(n, 3, 1)$ be a roofline polynomial. If $n$ is an odd integer then $f$ can not be a CNS polynomial.

**Theorem 2** If the $f(n, M, k)$ is a roofline polynomial and \( \gcd(n + 1, \frac{M - 2}{k} + 1) = d > 1 \), then $f$ can not be a CNS polynomial.

Priorly [3] considered expansion of small integers. We consider roofline polynomials with parameters $(n, M, k)$ where $k = 1$. The following theorem says that for a special case, the condition of Theorem 1 is sufficient for the expansion to exist, and we can also determine the expansion length.

**Theorem 3** Let $M \leq 3$ be an arbitrary integer. If \( \gcd(n + 1, \frac{M - 2}{k} + 1) = 1 \), then $-1$ has a canonical representative. Furthermore, there exists a quadratic polynomial $f(n) = f_M(n)$ whose values give the length of the expansion of $-1$ for the roofline polynomial with parameters $(n, M, k)$ for large enough $n$.

**References**


Performance impact of network encryption on log processing with Spark

Attila Péter Boros, Péter Lehotay-Kéry, Attila Kiss
Department of Information Systems, ELTE Eötvös Loránd University, Budapest, Hungary
attila9778@inf.elte.hu
lkp@caesar.elte.hu
kiss@inf.elte.hu

Various industries maintain large number of machines to run their production lines and services. These types of systems process and produce massive amounts of data to provide high quality and -availability for their customer services. Therefore, these systems should constantly be inspected to not only provide continuously the achieved standard levels but also upgraded to keep up with the market competition.

For this reason, to monitor, analyze and detect anomalies inwards the states and the generated logging information about the behavior of these systems, the difficulty of processing massive amounts of collected data was introduced and large variety of tools were recently developed to provide high performing and secure solutions on such problems.

Our aim is to examine on one of the best systems for our purposes and to find one of the most suitable configurations that performs best on our challenges and can be further applied in real, live scenarios. Consequently, we choose Apache Spark as a prominent distributed log processing framework with the purpose of finding an optimal cluster for our given resources and to elaborate on how different parameters affect the overall performance.

In addition has to be mentioned that several studies and investigations on this field were already made [1] by inventing high level system enhancements [4] for achieving better performance, and proposing theoretically well-based frameworks [2] and 3ly proved formulas [3] to not only estimate the time to run such computations but to suggest a best performing configuration with given allocatable resources and deadline for the computation to complete.

However the importance and effects of these investigations are unquestionably positive, they not consider the security factor of the system during computation. Here besides encrypting temporary files of computational chunks [5] a weak point of the system is the communication between cluster nodes, upon performance impact no similar investigations were found. Therefore, because of the complexity of the system, as a first empirical step on a theoretical road we set different scenarios for examining the impact of securing network communication between cluster nodes.

The presented work entails testing Apache Spark for log processing in client and cluster modes with varying number of worker nodes on different submitted tasks. Also, we examine the performance impact of encrypting network communication between nodes with these setups.

Our results show that by increasing the number of execution nodes computation performance does not always influences performance in positive manner as expected, and that securing network communication significantly increases the overall execution time of submitted jobs.

Acknowledgements

The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).
This publication is the partial result of the Research & Development Operational Programme for the project "Modernisation and Improvement of Technical Infrastructure for Research and Development of J. Selye University in the Fields of Nanotechnology and Intelligent Space", ITMS 26210120042, co-funded by the European Regional Development Fund.
The project was also supported by the Ericsson-ELTE Software Technology Lab. Furthermore thanks to Ericsson coworkers who worked on the project.

References


Application of models and hypergraphs on dynamic aspects of business process performance analysis

Khawla Bouafia, Maxim Kumundzhiev, Bálint Molnár
Information Systems Department, Eötvös Loránd University of Budapest, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
(bouafia, v249c6, molnarba) @inf.elte.hu

Abstract

Talking about Dynamic Business Processes (further "DBP"), we firstly should assume "DBP" [1.] as the modeling of graphical representation of DBP or workflows, as a means of identifying potential improvements.

The major goal of our research is to give formal and operational support for model checking and validation of Enterprise Information Systems (IS). The formal description of the enterprise architecture provides assistance for consistency and integrity controlling, furthermore the formal representation of the enterprise architecture supports the digital transformation and service innovation.

"DBP" graphical representation is often modeled in a way to represent activities, items, entities, its' relation, functions, and how to communicate between them in an enterprise in order to be successful.

As for the "DBP" representation, we decided to use graph representation approaches – especially hypergraphs [2.] – to depict the complex relationships that exist among the artifacts and constituents of Business Processes (BP). For more efficient and accurate manipulations [3.], we used bipartite and further hypergraph formats for storing and curating data.

Most "DBP" models [4.] have been developed as part of existing enterprise architectures, which shows that the intent during development is that the end-user is represented. However, these models have been defined from several different perspectives that include functional, behavioral, organizational, and informational. Experts agree that the combination of these perspectives in process design is the best method.

For that reason, we have investigated the various descriptive languages and representation models of BP as process modeling, workflow and process integration, and object-oriented languages. We have carried out experiments using four different approaches combinations, but for observing quilted representation, we focused on the main consistencies of "DBP".

As the final approach, we used "DBP" stream and data schemes that are defined by us to proceed with using pure Python for manually generating data and external Python libraries to store, curate, and visualize "DBP".

Acknowledgement
The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and (EFOP-3.6.3-VEKOP-16-2017-00001) and by no. ED_18_1-2019-0030 (Application-specific highly reliable IT solutions) program of the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme.
References


Applications of numeration systems for preserving privacy

Péter Burcsi, Gábor Nagy
Department of Computer Algebra, ELTE Eötvös Loránd University, Hungary
bupe@inf.elte.hu nagygabor@inf.elte.hu

In [1] the following problem was considered. Multiple participants provide data to an untrusted data aggregator. The goal is to enable the computation of an aggregate function, specifically the sum of the provided individual data, without exposing the individual data to an eavesdropper or even the aggregation server. In their proposed solution they make use of so-called additive zero-shares: a set of numbers owned by participants of the protocol that sum up to zero. Moreover, they are able to update the zero shares as time advances and thus have a repeated possibility to aggregate data without the need for an expensive setup phase.

In our paper, we investigate whether so-called numeration systems are also applicable for this purpose. Numeration systems are discretized dynamical systems on $\mathbb{Z}^n$. We analyze how repeated iteration of the underlying function affects zero-share properties of a correlated set of integer vectors.

Since in most application areas, it is required that data aggregation is not only performed once, but rather periodically, we need an aggregation scheme where initialization is only done once, and repeated gathering of the data only bears minor additional costs.

Our model problem will be the following. The participants $1 \ldots k$ hold secret, time dependent values $s_i(t)$, where $t \in \mathbb{N}$. We are interested in the sum $\sum s_i(t)$ for each time value $t$. We obtain this sum by generating secret random zero-share values $x_i(t)$, s.t. $\sum x_i(t) = 0$ for each $t$. Then the participants can securely compute the sum by publishing $(s_i(t) + x_i(t))$ The purpose of our investigation is to analyze if $i_i(t+1)$ can be obtained by $i_i(t)$ inexpensively while also keeping the required zero-share property.

Most such “updateable” zero shares use some-kind of homomorphism to update the shares. For example, in [1], a discrete log-based scheme is proposed, thus transferring the domain to a multiplicative setting, and the homomorphic properties of taking powers ensures that if $\prod y_i = 1$, then we also have $\prod y_i^c = 1$ for any choice of $c$.

In our case, we use an approximatively linear, yet – with the appropriate setting of parameters – hard to invert function, denoted by $\tau$. We take this function from the area of numeration systems, specifically matrix-based numeration systems, see [2]. The function $\tau$ can be seen as a rounded version of a linear map $M^{-1}$ for an expanding matrix $M$. A straightforward homomorphism is at hand: the linear map $M^{-1}$ preserves zero sums. However, our map $\tau$ is only a discrete approximation of $M^{-1}$, and thus fails to preserve zero sums even for two shares: there are values of $i$ for which $-\tau(x) \neq \tau(-x)$.

We obtain theoretical and empirical results on how well $\tau$ preserves the zero share property, that is we give upper bounds on $\sum_{i=1}^k \tau^n(x_i) = 0$ where $\tau^n$ is the n-fold iterate of $\tau$. Based on these estimates, we also discuss what parameter settings could conceivable be used in practice.

References


Remarks on the quantum complexity of some numeration related problems
P´eter Burcsi, G´abor Nagy, Attila R´eti
Department of Computer Algebra, ELTE E¨ otvos Lor´ and University, Hungary
bupe@inf.elte.hu nagygabor@inf.elte.hu haieck@inf.elte.hu

Numeration system generalize our concept of representing integers as the linear combination of powers of a base. We present a few remarks and results related to the quantum complexity of a few problems on matrix numeration systems. Our main motivation is to see whether matrix numeration systems can be applied in post quantum cryptography.

Definition 4 Let \( n \) be a positive integer, \( M \in \mathbb{Z}^{n \times n} \) with \( \det(M) \neq 0 \), \( 0 \in D \subseteq \mathbb{Z}^n \) a complete residue system mod \( M \) (i.e. \( |D| = |\det(M)| \) and for all distinct pair of elements \( d, d' \in D \), we have \( M^{-1}(d - d') \notin \mathbb{Z}^n \)). To such a pair \((M, D)\), we associate a discrete dynamical system given by the map \( \tau : \mathbb{Z}^n \to \mathbb{Z}^n \) defined as

\[
\tau(v) = M^{-1}(z - d) \quad \text{with the unique value} \ d \in D \ s.t. \ \tau(v) \in \mathbb{Z}^n.
\]

The pair \((M, D)\) together with \( \tau \) is called a matrix based numeration system. We say that the system has the finiteness property (it is a generalized number system) if for all \( v \in \mathbb{Z}^n \), there exists a positive integer \( k \) s.t. \( \tau^k(v) = 0 \) (here, \( \tau^k \) denotes \( k \)-fold iteration).

The following theorem has been reproved many times in the literature in different contexts, cite.

Theorem 5 Every non-zero vector \( v \) in \( \mathbb{Z}^n \setminus \{0\} \) is uniquely representable as a finite sum

\[
\sum_{j=0}^{k} M^j d_j
\]

with \( k > 0 \), \( d_j \in D \) and \( d_k \neq 0 \).

The above theorem explains the term generalized number system: ordinary (binary, decimal etc.) number systems have similar unique representation for all positive integers.

Example. Let \( n = 2 \), \( M = \begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix} \) and \( D = \{(0, 0), (1, 0)\} \). This system has the finiteness property. Identifying \( \mathbb{Z}^2 \) with the Gaussian integers, multiplication by \( M \) represents multiplication by \(-1 + i\). So Theorem 5 states that for all \( a, b \in \mathbb{Z} \) (not both zero), we have a unique representation

\[
a + bi = \sum_{j=0}^{k} (-1 + i)^j d_j
\]

where \( d_j \in \{0, 1\} \) and \( d_k = 1 \).

The following theorem is also known, see for example [2].

Theorem 6 If \((M, D)\) has the finiteness property, then \( M \) is expansive, i.e. all eigenvalues of \( M \) lie outside the closed unit disc.

Given an expansive matrix \( M \) and a digits set \( D \) we investigate the following two problems:

- Do all inputs eventually reach 0 upon iterating \( \tau \)?
- If there are other periods, how many such periods exist, how long are they, where are they located?

Inspired by ideas from Shor’s algorithm [1], we analyze how a superpositional computation of several orbits reveals information on the period structure.
References


Regular elements and generalized inverses in (matrix) rings of residue classes

Iulia-Elena Chiru

Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania

iulia.chiru@math.ubbcluj.ro

The concept of (von Neumann) regular ring was given by John von Neumann in his famous work [4]. A ring $R$ is called regular if every element $a \in R$ is regular, in the sense that there exists an element $b \in R$ such that $aba = a$. In this case $b$ is called a generalized inverse (also called $\{1\}$-inverse or inner inverse) of $a$. Generalized inverses have created a wide field of applications, mainly towards solving linear systems of equations whose matrices are not necessarily invertible, but admit a generalized inverse. There has also been an interest in developing the theory of regular rings and generalized inverses in particular settings, such as rings of residue classes modulo $n$ or matrices over finite fields [1, 3], sometimes with applications to cryptography [2].

Every regular element $a$ of a ring $R$ has not only a generalized inverse, but also a $\{2\}$-inverse (or outer inverse), in the sense that there exists an element $b \in R$ such that $bab = b$. We generalize the known characterization of regular elements of the ring $\mathbb{Z}_n$ of residue classes modulo $n$ towards two directions. First, we describe the elements of $\mathbb{Z}_n$ which admit a $\{2\}$-inverse. Secondly, we characterize regular elements of some matrix rings over $\mathbb{Z}_n$.

This is based on a joint work with Septimiu Crivei (Babeş-Bolyai University, Cluj-Napoca, Romania).

References

Let $B \to A$ be a homomorphism of Hopf algebras, and let $1$ be an algebra. We consider the induction from $B$ to $A$ of $1$ in two cases: when $1$ is a $B$-interior algebra, and when $1$ is a $B$-module algebra. Our main results establish the connection between the two inductions. The inspiration comes from finite group representation theory, and some constructions work in even more general contexts.

References

Relatively divisible and relatively flat objects in exact categories
Septimiu Crivei
Faculty of Mathematics and Computer Science, Babeș-Bolyai University, Cluj-Napoca, Romania

crivei@math.ubbcluj.ro

Many important characterizations of rings are of homological algebra nature. For instance: a ring $R$ is semisimple if and only if every short exact sequence of right (or left) $R$-modules splits; a ring $R$ is von Neumann regular if and only if every short exact sequence of right (or left) $R$-modules is pure; a ring $R$ is right pure-semisimple if and only if every pure short exact sequence of right $R$-modules splits. These examples exhibit three classes of short exact sequences in a module category, namely split short exact sequences, pure short exact sequences and all short exact sequences. Such classes of short exact sequences share some properties, that can be considered as axioms for a more general notion, called exact structure (Keller [4], Quillen [6]) on an additive category. Exact structures on additive categories are suitable for developing homological algebra in categories more general than module categories or abelian categories (e.g., see [1] for an exhaustive account).

We introduce and study relatively divisible and relatively flat objects in exact categories. For every relative cotorsion pair $(\mathcal{A}, \mathcal{B})$ in an exact category $\mathcal{C}$, $\mathcal{A}$ coincides with the class of relatively flat objects of $\mathcal{C}$ for some relative projectively generated exact structure, while $\mathcal{B}$ coincides with the class of relatively divisible objects of $\mathcal{C}$ for some relative injectively cogenerated exact structure. We exhibit Galois connections between relative cotorsion pairs in exact categories, relative projectively generated exact structures and relative injectively cogenerated exact structures in additive categories. We establish closure properties and characterizations in terms of approximation theory. We also present several applications to module categories.

This is based on joint works [2, 3] with Derya Keskin Tütüncü (Hacettepe University, Turkey).

References

Formalizing a relational model of concurrent programs in a dependently typed environment

István Donkó†, Ambrus Kaposi‡, Melinda Tóth§
ELTE, Eötvös Loránd University
Faculty of Informatics
Department of Programming Languages and Compilers
†isti115@inf.elte.hu ‡akaposi@inf.elte.hu §tothmelinda@elte.hu

Sequential programming languages have already been formalized in dependently typed programming languages, such as for example Agda or Coq, but the computer based formalization and verification of concurrent programs is still in its early days. The goal of our research is to formalize a relational model [4] that describes the behavior of distributed concurrent programs in a theorem prover system. Our long term goals include the formalization of the material of the subject titled "Specification and Implementation of Distributed Systems", which serves as a core part in the Computer Science education at Eötvös Loránd University in Budapest.

Since our goal is not the introduction of a new way to address the problem of creating correctness proofs for parallel programs, rather the adaptation of an existing system for formalized implementation, instead of discussing the theoretical methods, here we focus on a more practical approach and explore what others have achieved in the field of computer based verification.

Motivation  Software plays a critically important role in the life of modern societies. More or less everybody interacts with computer programs on countless occasions during our everyday lives, most of the time probably not even noticing. For example, just paying with a credit card while shopping, being able to call someone with our mobile phones, or even as mundane tasks, as operating modern home appliances, like washing machines or microwave ovens requires interaction with software. Problems occurring in these example situations mostly just cause inconveniences, but if we take the more critical scenarios into account, such as for example software running on an airplane, or keeping a nuclear power plant safe, we can see that programmers have an even bigger impact.

Borrowing from Robert C. Martin we can define the beginning of programming around the work of Alan Turing, since he was the first one, who wrote code for machines in the sense that we would recognize today. In his time, he described the future possibilities of his vision with the following sentences in a lecture to the London Mathematical Society [6].

"In order to supply the machine with these problems we shall need a great number of mathematicians of ability. These mathematicians will be needed in order to do the preliminary research on the problems, putting them into a form for computation."

He stated the need for mathematicians for the precise formalization of problems. As we can see today, his approach is necessary for building critical systems, the correctness of which can decide between life and death, we need to have formal strategies to verify behaviors of programs under all circumstances instead of just observing them for the most likely situations.

Introduction  There are lots of existing means for confirming the adherence of simple sequential programs to their specifications, ranging from formal verification procedures carried out on paper to contracts built into programming languages, that can be checked and enforced automatically, either via static code analysis, or during runtime by monitoring different values. Reasoning about

---

4The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16- 2017-00002). The research is part of the "Application Domain Specific Highly Reliable IT Solutions" project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme.
parallel programs is a lot more complicated, but this complication also serves as an explanation for the need to do so, since concurrency is often a result of multiple systems working together, in which case it is a lot easier to make mistakes because of the unpredictable order of execution of instructions. Several different ways are known to approach formal proofs of correctness for concurrent programs. For example multiple specific methods can be seen in [2]. What we chose to base our research on is the material of the subject titled "Specification and Implementation of Distributed Systems".

Our formalization does not follow the material exactly, but we tried to stay as close to the original notation as possible, as we also have intentions to later further expand this project to cater for educational usage for example as part of the practical courses. We consider this to be a valuable opportunity for creating a teaching tool that can greatly aid the understanding of the subject for students. The use of proof assistants has already been successfully introduced in several other classes [5], which helps making this idea seem quite feasible.

Background We used type theory (namely, the Agda [1] implementation of it) as the main tool for creating and verifying our formalization, which – by being an expressive alternative foundation for mathematics – enables the formalization of constructive proofs through the connections to intuitionistic logic given by the Brouwer–Heyting–Kolmogorov interpretation. After formalizing a model by defining its types and their elements, one can express statements and theorems in forms of new types, the instances of which can be thought of as proofs for them. This is due to the so called "propositions-as-types" paradigm, formally known as the Curry–Howard isomorphism.

Results The main results of our work include the fully formalized version of a big core part from the original model [3] we built our research around, which is precise enough for computer based typechecking. This consists of a definition for a language which contains parallel conditional assignments as well as several types of predicates and statements that can be used to describe specifications for programs written in this language. On top of these foundational constructs we have developed proofs for several generic lemmas and some bigger theorems. We have also formalized a parallelized version of the bubble sort algorithm and the verified some of its properties.

References

[3] István Donkó. Formalizing a relational model of concurrent programs in a dependently typed environment. Paper at the Student Association Conference, Faculty of Informatics, Eötvös Loránd University, May 2020, Received 1st prize.
Blaschke-products and Hyperbolic Geometry

Tamás Dózsa, Ferenc Schipp
Department of Numerical Analysis, Eötvös Loránd University
dotuaai@inf.elte.hu, schipp@inf.elte.hu

In our presentation we discuss a generalization of the Poincaré disk model of Bolyai’s geometry and the physical (optical) interpretation of our results. The geometry of the model is provided by Blaschke-functions. The hyperbolic lines (the sections of circles inside the disk $\mathbb{D}$, which are perpendicular to the torus $\mathbb{T}$) can be given as the Blaschke-images of the interval $(-1,1)$. They can also be given as the solutions to the variational problem:

$$\min_{\Gamma} \mathcal{J}(\Gamma), \quad \mathcal{J}(\Gamma) := \mathcal{J}(\Phi) := \int_{t_1}^{t_2} \frac{|\Phi'(t)|}{1 - |\Phi(t)|^2} \, dt,$$  \hspace{1cm} (3)

where

$$\Gamma := \{ \Phi(t) : t_1 \leq t \leq t_2 \}, \quad \Phi(t_j) = z_j \quad (j = 1,2)$$

is a parametrization of the simple smooth curve $\Gamma$. The hyperbolic line segment connecting the points $z_1$ and $z_2$ is given as the solution $\Phi_0$ to problem (3) as $[z_1, z_2] = \{ \phi_0(t) : t_1 \leq t \leq t_2 \}$. The previous statement is a simple consequence of the identity

$$\frac{|B'_a(z)|}{1 - |B_a(z)|^2} = \frac{1}{1 - |z|^2} \quad (a, z \in \mathbb{D})$$  \hspace{1cm} (4)

regarding Blaschke-functions.

According to the optical Fermat-principle, for a refractive index of $v(z) := 1 - |z|^2 \quad (z \in \mathbb{D})$, the solution to (3) describes the path of a light-beam traversing from $z_1$ to $z_2$.

In our presentation we discuss a generalization of the variational problem (3) and describe its solutions. Instead of $\frac{1}{1 - |z|^2} \quad (z \in \mathbb{D})$, we consider density functions of the form

$$\sigma(z) := \frac{|B'_a(z)|}{1 - |B_a(z)|^2} \quad (z \in \mathbb{D})$$  \hspace{1cm} (5)

where

$$B_a(z) := \prod_{j=0}^{n-1} B_{a_j}(z) \quad (\mathbf{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{D}^n, z \in \overline{\mathbb{D}})$$  \hspace{1cm} (6)

denotes a Blaschke-product.

We construct the inverse images of the $n$-fold mapping $B_a$ and consider their geometric properties as well as related numerical algorithms. In the special case of $n = 2$, we provide the inverse image branches in an explicit form using the square root function. Transformation of the hyperbolic lines with the inverse images yields the solution to the generalized variational problem. Using these solutions we introduce an $n$-fold geometry on $\mathbb{D}$.

---

5EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies – The Project is supported by the Hungarian Government and co-financed by the European Social Fund.
References


Graph-based duplicated code detection with RefactorErl

Zsófia Erdei, Melinda Tóth, István Bozó
ELTE, Eötvös Loránd University
zsanart@inf.elte.hu, toth.m@inf.elte.hu, bozo.i@inf.elte.hu

Previous studies show, that a significant fraction (between 7% and 23%) of the code in a typical large software system consists of code clones [2]. By code duplicates we mean syntactically or semantically similar snippets of code, that occur more than once in the source code. Code duplicates are generated by various reasons such as code reuse by copying existing fragments of code (copy-and-paste programming). While such cloning is often intentional and can be useful in many ways, generally in the long term the presence of duplicate code makes software maintenance more difficult and as a consequence more expensive. For example by reusing existing functions with slight changes in variables or data structures, not only do we increase the chance of bug occurrences, but also if an instance of duplicate code is changed, its clones have to be modified as well. Duplicated fragments can also significantly increase the work to be done when adding new features to our software, and make code quality analysis and code comprehension more tedious as well.

Several tools exist [1, 2] using different approaches and methods that can be used to identify code duplicates within the source code. Such tools are called duplicated code detectors. The majority of these tools focuses on a concrete language, but there are also some attempts of creating general tool for finding code duplicates in any type of source code. These methods are mostly based on pattern matching on the raw source code mostly using a sliding window algorithm, or analyzing a sequence of tokens generated from the source code. These methods can be used to identify completely identical snippets of code, but they cannot be reliably used to detect code clones that have been modified. The more refined tools for identifying code duplicates exists mostly for mainstream languages, in the context of functional programming, only a few of them are available.

In this paper we give a graph based algorithm [6] which uses the Semantic Program Graph generated by the tool RefactorErl [4, 5], a source code comprehension and refactoring tool for the programming language Erlang [3], to find different types of code clones in the source code. The presented algorithm was able to efficiently detect not only textually identical code fragments (Type I. [2]) but also copied fragments with further modifications such as changed, added or removed statements, in addition to variations in identifiers, literals, types, whitespace, layout and comments (Type II., Type III.).

References


The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00033, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications). The research is part of the ”Application Domain Specific Highly Reliable IT Solutions” project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme.


[6] Zsófia Erdei: Gráf alapú ködduplikátum felismerés a RefactorErl segítségével, Paper at the Student Association Conference, Faculty of Informatics, Eötvös Loránd University, December 2019, Received 2nd prize.
About spatially homogeneous space-time models

Szabolcs Levente Fancsali, János Szenthe
Department of Computer Algebra and Department of Geometry, Eötvös Loránd University
nudniq@cs.elte.hu

Based on observations, we can assume that our space-time is spatially homogeneous and spatially isotropic on a large scale. However, the terms ‘spatially homogeneous’ and ‘spatially isotropic’ should be precisely defined. There is at least one counter-example (the so called Gödel space-time) that shows that the intuitive concept of ‘spatial homogeneity’ cannot be a generalization of the concept of ‘homogeneity’.

So, at first we select the most suitable definition of spatially homogeneous space-times from among several possible ones:

**Definition 7** The Lorentz-manifold \((M, g)\) of dimension \(n + 1\) is spatially homogeneous if it has an isometry group \(G\), whose orbits are spatial sub-manifolds of dimension \(n\).

Then we prove that the homogeneous spatial hypersurfaces of spatially homogeneous space-times (in this above sense) are the level surfaces of a universal time function:

**Theorem 8** If the total isometry group of the Lorentz-manifold \((M, g)\) of dimension \(n + 1\) has a Lie-subgroup \(G\) such that the orbits of \(G\) are spatial sub-manifolds of dimension \(n\) (i.e. \(M\) is spatially homogeneous) then there exists an observer field \(\Sigma\) such that these orbits are rest spaces with respect to \(\Sigma\) and such there exists a cosmic time function \(t\) such that these orbits are the level surfaces of \(t\).

We then show that a spatially homogeneous space-time is locally a product manifold (of ‘space’ and ‘time’):

**Theorem 9** If the Lorentz-manifold \((M, g)\) is spatially homogeneous (with the isometric group action \(G \times M \rightarrow M\) whose orbits are spatial hypersurfaces) then for each point \(p \in M\) there exists an open interval \(I\) such that a suitable open (tubular) neighbourhood of the orbit \(G(p)\) is diffeomorphic to the product manifold \(I \times G(p)\).

We examine whether we can infer spatial isotropy even though we can only observe a light-cone pointing to the past. The exact definition of ‘spatial isotropy’ should also be select carefully:

**Definition 10** Let \((M, g)\) be a spatially homogeneous Lorentz-manifold of dimension \(n + 1\) such that the orbits of the isometric group action \(G \times M \rightarrow M\) are spatial hypersurfaces of dimension \(n\). Then \((M, g)\) is said to be ‘spatially isotropic’ if \(\forall p \in M\) the linear representation of the stabilizer subgroup of \(p\) contains the total orthogonal group of the tangent space \(T_p G(p)\).

We show that if space-time is not only spatially homogeneous, but also these homogeneous hypersurfaces are isotropic, then these three-dimensional hypersurfaces have a constant curvature (if space-time is four-dimensional):

**Theorem 11** If the spatially homogeneous Lorentz-manifold \((M, g)\) of dimension four is also spatially isotropic then the homogeneous hypersurfaces \(G(p)\) are Riemannian manifolds (by induced metric) with constant curvature.
Then we calculate the $E \overset{\text{def}}{=} \text{Ric} - \frac{1}{2} \sigma \cdot g$ Einstein tensor of the spatially homogeneous space-time $I \times_\phi S$ produced as a ‘warped product manifold’.

$$E(\hat{U}, \hat{U}) = 3 \left( \frac{\dot{\phi}}{\phi} \right)^2 - \frac{1}{2} \tilde{\sigma} \langle dt | U \rangle^2$$

$$E(\hat{U}, \hat{Y}) = 0$$

$$E(\hat{X}, \hat{Y}) = 2 \left[ \frac{\ddot{\phi}}{\phi} + 2 \left( \frac{\dot{\phi}}{\phi} \right)^2 \right] \tilde{g}(X, Y) + \left[ \tilde{\text{Ric}}(X, Y) - \frac{1}{2} \tilde{\sigma} \tilde{g}(X, Y) \right]$$

where the last brackets contains the Einstein tensor of the Riemannian manifold $(S, \tilde{g})$.

We determine the conditions under which space-time can be produced as a warped product (at least locally):

**Claim 12** If a space-time is a spatially isotropic product manifold such that each spatial fibrum has positive curvature (or each fibrum has negative curvature or each fibrum has zero curvature) then this space-time is a warped product.

**Remark 13** But there exists such warped product that is not spatially isotropic.

Finally, for some specific spatially homogeneous and isotropic space-time models, we show how the Einstein tensor computed for warped products and the extra conditions applied to the stress-energy tensor by physical intuition together transform the Einstein equation into ordinary differential equations.

**References**


Positivity Preserving Numerical Method for an Extended Ross Model for Malaria Propagation

István Faragó, R. Mosleh

Budapest University of Technology and Economics Budapest, 1111 Budapest, Egry József s. 1., Building H, Hungary
rmosleh@math.bme.hu

Malaria is an infectious and fatal disease transmitted to humans through the bites of infectious female Anopheles mosquitoes. This phenomenon heavily depends on climate factors including temperature, altitude, rainfall and relative humidity. It means that high temperatures, rainy seasons and raised humidity would increase distribution and transmission of malaria. It is also widely spread out in subtropical regions such as Africa, Asia, Latin America and some parts of Europe like Hungary, Austria, and Italy (c.f.[1,2]). Malaria is an ancient disease and since there is no effective vaccine, people still have to cope with it.

To get a better insight of malaria and reduce the impact of it in the world, some mathematical models of malaria have been constructed such as Ross, Ross-Macdonald, delayed Ross-Macdonald, Anderson and May models and others. The Ross model as a SIS-type model is discussed in constant population sizes for humans and mosquitoes and the solutions are densities in the interval $[0,1]$ (c.f.[3]). In this study we consider an extension of the Ross model numerically. The extended Ross model rectifies some shortcomings of the Ross model and is interpreted as a SEIR-type model in nonconstant population sizes for humans and mosquitoes through a non-linear seven-dimensional system of ordinary differential equations. Qualitatively, the solutions of the extended Ross model are positively invariant in some specific intervals (c.f.[4]) and also there are two equilibrium points, disease-free and endemic points, which are globally stable under some restrictions (c.f.[5]). Numerically, since applying the $\theta$-method for the extended Ross model is rigid to prove the positivity preservation for the solutions, a semi $\theta$-method which is a sort of nonlocal discretization is used as a remedy to solve the extended Ross model. In this study we mainly focus on the explicit Euler method.

The research findings indicate the numerical solutions of the extended Ross model are positive and invariant in some intervals and the upper bounds of the solutions for humans and mosquitoes are the maximum value of the initial total population and disease-free total population for humans and mosquitoes respectively. It means that the solutions of the extended Ross model denote a kind of extension of the solutions of the Ross model. Moreover, both the continuous and the corresponding numerical models of the extended Ross model preserve the basic qualitative properties of the solutions and numerical examples are given to confirm the theoretical results.
References


[2] Lillian L. M. Shapiro, Shelley A. Whitehead, Matthew B. Thomas, *Quantifying the effects of temperature on mosquito and parasite traits that determine the transmission potential of human malaria*.


Number expansions can be constructed in many different ways. One of the most natural way is to consider a lattice, a linear operator acting on it and a finite digit set describing the expansions. This paper deals with simultaneous number system constructions over the Eisenstein lattice. The authors present theoretical and algorithmic results as well.

The Eisenstein integers are complex numbers of the form \( \eta = a + b \omega \) where \( a, b \in \mathbb{Z} \) and \( \omega = -\frac{1 + \sqrt{3}i}{2} = \exp(2\pi i/3) \) is a cube root of unity. They form a triangular lattice in the complex plane, or in a different view, they form a commutative ring of the algebraic integers in the third cyclotomic field. Equivalently, we can consider the Eisenstein integers as linear operators of the form

\[
M_1 = \begin{pmatrix} a & -b \\ b & a & -b \end{pmatrix}
\]

acting on \( \mathbb{Z}^2 \). Let us define the digit set by

\[
D = \left\{ \langle x \bullet x \bullet x \rangle : x \in \bigcup_{d_2 \in D_2} \bigcup_{d_3 \in D_3} M_1 M_2 d_3 + M_1 d_2 + D_1 \right\}
\]

where \( M_2 = \begin{pmatrix} a + 1 & -b & \omega b \\ b & a + 1 & -b \end{pmatrix} \) and \( M_3 = \begin{pmatrix} a + 1 & -b - 1 \\ b + 1 & a - b \end{pmatrix} \) are linear operators, and the digit sets \( D_i \) are adjoint type digit sets over \( \mathbb{Z}^2 \) with the operators \( M_i \), respectively and \( \bullet \) denotes the concatenation. The research investigates the number system property of the systems \((\mathbb{Z}^6, M_1 \oplus M_2 \oplus M_3, D)\).

References


Efficiency Improvement of Adaptive Random Forest using Principle Component Analysis for Mining Data Stream

Hayder K. Fatlawi, Attila Kiss
Eötvös Loránd University, Budapest, Hungary, Faculty of Informatics, Department of Information Systems
hayder@inf.elte.hu, kiss@inf.elte.hu

Abstract

The mining of data stream focuses on adapting the trained classifier in a continuous process while new stream elements have arrived. This process has many constraints in comparison with batch data mining such as; limitation of available memory, fast response for each new instance, and forgetting the old instances to adapt to the new ones. While typical random forest as an ensemble technique uses random sampling to train many classifiers, Adaptive Random Forest ARF uses adaptive sliding window ADWIN and Heoffman bound to handle the streaming process [1].

Although the reduction of the required time and space due to mining a limited number of instances in a specific time, the high dimensionality still represents a challenge for stream mining techniques. Principal Component Analysis PCA is one of the most popular methods for reducing the number of attributes based on the Eigenvector and Eigenvalue concept [2]. This work aims to improve the efficiency of ARF using PCA, this improvement ensures that the time of PCA step shouldn’t delay the response of the classifier and it should preserve the accuracy of the classification process.

The implementation of the proposed method included utilizing Massive Online Analysis MOA framework and Python within Anaconda framework. Three medical datasets are used to verify the performance which are EEG Eye state, Hypothyroid, and Breast Cancer datasets with 15, 25, 11 attributes respectively including the class. The performance of ARF with PCA overcame typical ARF with the first dataset EEG Eye state in which the CPU cost was reduced by 55% and 80% for RAM cost reduction while the accuracy of classification was 98% in both cases. In the experiment of Hypothyroid dataset, the reduction of CPU cost was 33% and 48% for the RAM cost reduction with preserving the accuracy at 99%. The Breast Cancer dataset had 8% reduction for both CPU and RAM costs.

Acknowledgment

The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

References


Theoretical and simulation results for a 2-type network evolution model

István Fazekas¹, Attila Barta²

Department of Applied Mathematics and Probability Theory, University of Debrecen
¹fazekas.istvan@inf.unideb.hu
²barta.attila@inf.unideb.hu

Network theory is one of the most popular research topics of our age. It studies both real-life networks and theoretical models. Networks are described by graphs. The nodes of the network are the vertices of the graph and the connections are the edges. One of the most famous models is the preferential attachment model introduced by Albert and Barabási. It is a discrete time model (that is the evolution events occur at time \( n = 1, 2, \ldots \)) and it describes connections of two nodes. The meaning of connection can be cooperation or any interaction. Therefore the connections of more than two nodes are also important. For example, Backhausz and Móri in [2] describe three-interactions, Fazekas and Porvázsnyik in [4] \( N \)-interactions, or Fazekas and Perecsényi in [3] star-like connections. Continuous-time network evolution models seem to be more difficult but more realistic models than the discrete time ones. In [7] a continuous-time branching process is applied to govern the evolution mechanism. In [1] we extended the results of [7] for 3 interaction models. There we applied the general theory of branching processes [5].

We shall study the following random graph evolution model. At the initial time \( t = 0 \) we start with a single object, it can be either an edge or a triangle. We call this object the ancestor. This ancestor object produces offspring objects which can be also edges or triangles. Then these offspring objects also produce their offspring objects, and so on. The reproduction times of any object, including the ancestor, are given by its own Poisson process with rate 1. We assume that during the evolution, the reproduction processes of different objects are independent. The reproduction processes of the triangles are independent copies of the generic triangle’s reproduction mechanism. Similarly, the reproduction processes of the edges are independent copies of the reproduction mechanism of the generic edge.

We present the general results on our model. These are the survival function and the mean offspring number. We show asymptotic theorems on the number of triangles and the number of edges. All of them have magnitude \( e^{\alpha t} \) on the event of non-extinction, where \( \alpha \) is the Malthusian parameter. To prove our results we apply general theorems on multitype branching processes [6], [8], [9]. And last we present some simulation results supporting our theorems.

References

Acknowledgements

This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.
Asymptotic results for contaminated runs of heads

István Fazekas, Suja Michael Ochieng
Faculty of Informatics, University of Debrecen; fazekas.istvan@inf.unideb.hu, pga7iy@mailbox.unideb.hu

In this paper we consider the length of consecutive heads interrupted by several tails in different locations in the usual coin tossing experiment. So let \( p \in (0, 1) \) be the probability of head and \( q = 1 - p \) the probability of tail.

We toss the coin \( N \) times independently. Let \( T \geq 0 \) be a fixed integer. We shall focus our study on the precisely and at most \( T \)-contaminated (in other words \( T \)-interrupted) runs of heads having length \( n \). A \( T \) contaminated head run is essentially a moving sum of \( T + 1 \) adjacent iid geometric random variables.

One of the most famous classical results of the topic is the following theorem by Erdős and Rényi [1]. Consider the case of a fair coin, that is let \( p = 1/2 \). Let \( \mu(N) \) denote the length of the longest pure head run during \( N \) independent coin tosses. Let \( 0 < c_1 < 1 < c_2 < \infty \) be arbitrarily chosen. Then for almost every elementary event \( \omega \) there exists a finite \( N_0 = N_0(\omega, c_1, c_2) \) such that

\[
[c_1 \log N] \leq \mu(N) \leq [c_2 \log N]
\]

if \( N \geq N_0 \). Here \( \log \) denotes logarithm to base 2 and \( [\cdot] \) is the integer part.

It is known that the properties of longest (pure head or pure tail) runs are used to test random number generators, see e.g. [4].

Földes in [2] studied the case of the fair coin toss. She proved asymptotic results for the distributions of the number of pure head runs, the first hitting time of a run having fixed length, and the length of the longest run (pure head or pure tail run). Then she announced the extensions of the above mentioned results for \( T \) contaminated runs of heads, but she did not give any proof of these results. The aim of our paper is to extend the results of Földes [2] to the case of possibly biased coins and to give detailed proofs.

Gordon, Schilling, and Waterman in [3] used extreme value theory to find the asymptotic behaviour of the expectation and the variance of the length of the longest precisely \( T \)-contaminated head run.

Móri in [5] also studied the longest precisely \( T \)-contaminated head run. He presented the limiting distribution for the first hitting time (without proof) and the so called almost sure limit theorem (see Corollary 5.1 in [5]).

We emphasize that, contrary to [5] and [3], our main results concern the case of at most \( T \)-contaminated head runs and not the precisely \( T \)-contaminated head runs. We shall investigate the asymptotic results for the distribution of the number of at most \( T \) contaminated head runs, the first hitting time of such a run and the length of the longest such run.

Our main condition is the following. Let \( N \to \infty \) and \( n \to \infty \) so that

\[
\frac{Nq^{T+1}p^{n-T}n^T}{T!} \to \lambda > 0,
\]

where \( \lambda \) is fixed. We remark that condition (8) implies that \( N/n \to \infty \).

We define the number of at most \( T \)-contaminated runs of heads having length \( n \) as follows.

Let

\[
\eta_i = \eta_i^T(n) = \begin{cases} 1, & \text{if there are at most } T \text{ 0 values among } X_i, \ldots, X_{i+n-1} \\ 0, & \text{otherwise} \end{cases}
\]
Now let
\[ \xi^T(n, N) = \sum_{i=1}^{N-n+1} \eta^T_i(n) \] (10)
be the number of head-runs being at most \( T \)-contaminated and having length \( n \). The following theorem shows that the distribution of \( \xi^T(n, N) \) converges to compound Poisson distribution.

**Theorem 14** Let \( T \) be fixed. Let \( N \to \infty \) and \( n \to \infty \) so that the condition (8) above is satisfied. Then for the generator functions we have
\[
\lim_{N \to \infty} E\left(z^{\xi^T(n, N)}\right) = \exp \left[ \lambda \left( \frac{qz}{1 - pz} - 1 \right) \right].
\]

Let \( \tau^T(n) = \min\{N : \xi^T(n, N) > 0\} \). \( \tau^T(n) \) is the first hitting time of the run having length \( n \) and containing at most \( T \) tails. We show that the appropriate normalised version of \( \tau^T(n) \) has exponential limiting distribution.

**Theorem 15** Let \( T \) be fixed. Then for any \( 0 < x < \infty \)
\[
\lim_{n \to \infty} P\left( \frac{\tau^T(n)}{n} T! q^{T+1} p^{n-T} \leq x \right) = 1 - e^{-x}.
\]

Let \( \mu^T(N) = \max\{n : \xi^T(n, N) > 0\} \). \( \mu^T(N) \) is the length of the longest run of heads containing at most \( T \) tails. The following theorem describes the accompanying distribution of \( \mu^T(N) \).

**Theorem 16** Let \( T \) be fixed. Then for any integer \( k \)
\[
P\left( \mu^T(N) - \lfloor \log N + T \log \log N \rfloor < k \right) =
\exp \left( -q^{T+1} p^{(k-T-(\log N + T \log \log N))/T!} \right) + o(1).
\]
Here \( \log \) denotes logarithm to base \( 1/p \), \( \lfloor . \rfloor \) is the integer part while \( \{ . \} \) is the fractional part.

**References**

Identifying Vulnerable Software Modules with Static Analysis and Version Control

Anett Fekete, Zoltán Porkoláb
Eötvös Loránd University, Faculty of Informatics
Department of Programming Languages and Compilers
{afekete, gsd}@inf.elte.hu

Large software projects inevitably contain vulnerabilities. Maintenance risks can derive from multiple sources, such as overly tight coupling between modules or files, architectural flaws, the usage of antipatterns etc. Several research has already presented some type of metric in order to quantify software program vulnerabilities, such as [3, 2, 1]. These papers serve as good basis for this research which aims to utilize prior research results in determining vulnerability metrics with including version control information as a refining factor.

Nowadays, version control is available and used even in the development of a small project, since it makes programming more flexible and safer. Repositories in popular version control systems, such as Git and SVN contain plenty of information about the software project. This information is often implicitly stored, requiring deep analysis [4] to be brought to surface. Through investigating version control data, we can learn a lot about the structure of the software as well as its development history, from which we may conclude the architectural and practical decisions behind the software. Also, we can discover liabilities in the present state of the software which - if corrected - may spare money, time and human resources in the future.

In this paper, we present a method of identifying vulnerabilities in software modules with the usage of version control data. We determine whether a file poses maintenance risk, and to what extent, by investigating the measure of coupling between the examined file and any other one, and the frequency of modifications affecting the file. The metric of vulnerability is determined according to the following categories:

- If a file does not have many dependants and is rarely modified, it poses no maintenance risk.
- If a file has numerous dependants but is scarcely modified, it poses slight maintenance risk.
- If a file does not have many dependants but is frequently modified, it poses slight maintenance risk.
- If a file has several dependants and is frequently modified, it poses high maintenance risk, thus it is deemed a vulnerability.

In our research, we also attempt to quantify software maintenance risks by taking the amount of coupling between files and modules and version control data (mainly the frequency of committed modifications for each file) into consideration. Utilizing the extracted information might help in debugging, correcting architectural flaws and avoiding future execution failures in the software.

We conducted multiple case studies on large open-source software using the CodeCompass code comprehension software. CodeCompass is capable of identifying dependencies between files in a software, let those be header inclusions, token usages, object instantiations etc., thus it is a decent tool for quantifying coupling in a software project. The test projects include libgit2, GoogleTest, and CodeCompass itself.

References


Enhancing Apparent Personality Trait Analysis with Cross-Modal Embeddings

Ádám Fodor
Department of Artificial Intelligence, Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
foauaai@inf.elte.hu

Automatic personality prediction is essential to machine understanding of human behaviour, social relations, human resourcing and for high-quality human–machine interactions. One of the most studied model to describe personality is the Big Five personality traits [1]. The theory identifies five factors: EXTraversion, NEUroticism, AGReeableness, CONscientiousness and OPEnness. Each personality trait represents a range bounded by two extremes, e.g., for extraversion, the two polar ends are extreme extraversion and extreme introversion.

Audio-visual personality trait prediction has become of high interest [2] due to high quality databases released in the challenges ChaLearn: First Impressions V1 and V2 [3]. In this study, we used the extended and revised dataset (V2). The dataset contains 10,000 video clips extracted from more than 3,000 different YouTube high-definition videos of people mostly facing and speaking to a camera.

Although, multimodal systems present a gain compared to monomodal systems, it rises a number of challenges as well. For example, the selection of which modalities to include in multimodal systems, deriving a correct framework to fuse them, or the cutting down the propagation of errors from noisy, missing or underrepresented data. One characteristic of the dataset is that the data distribution is unbalanced with fewer extreme samples, which are equally or more important in several use cases, including medication.

Multimodal fusion approaches often hardly consider complex intra- and inter-modal interactions and lack robustness in case of noisy or missing modalities [4]. Due to these challenges, an increasing number of studies were conducted to transfer knowledge across domains or modalities [5, 6]. To solve such issues, embedding methods have been proven useful for integrating such inter-dependencies. It has been found that similarity and correlation of semantic information retrieved from real data can be represented using deep metric learning in an embedded feature space [7, 8].

Our contributions are listed below:

1. We propose a general-purpose learning framework to extract modality-invariant embeddings from multiple information sources with a siamese network, which emphasize the extreme examples and implicitly improve the multimodal fusion process.

2. We extended the Multi-Similarity loss [9] to handle multiple apparent personality trait class labels simultaneously, besides using various input modalities. In case of non-extreme examples one or more modalities contain inadequate information to aid the deep embedding process. In this regard, we modified the online hard sample mining procedure to only consider extreme samples as anchor.

3. Although, samples having lower and/or higher personality trait values are less frequent in the database, the prediction of their precise values is highly desired in various situations where an extreme personality trait is of relevance. We show that cross-modal embedding enhance the prediction of the Big Five personality traits in the extreme cases.
To our best knowledge, this is the first work that introduce cross-modal embedding for personality trait prediction.

Our aim is to create a shared coordinate space, transforming the audio, video and text descriptors in a semantically relevant way with a siamese network. Triplet-based loss functions are designed to encourage positive examples as close as possible to the anchor, and negative examples to be apart from each other over a given threshold. The cross-modal embedding are expected to emphasize the two extreme poles trait-wise. Then we use a multimodal deep neural network (DNN) that combines features from visual, acoustic and textual clues, along with the embedding vectors to predict apparent Big-Five personality traits using short video clips from the ChaLearn challenges.

The performance score for comparing different methods is the '1-Mean Absolute Error' (1-MAE). The baseline multimodal DNN produced score of 0.9094, and we improved it to 0.9127 with our extension.

Acknowledgment

The research has been primarily supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002). We would like to thank Prof. András Lőrincz for the guidance throughout this research.

References


Automated defense against side-channel attacks

Dávid Fonyó, Péter Burcsi
Eötvös Loránd University (ELTE), Department of Computer Algebra
fonyodav@inf.elte.hu, bupe@inf.elte.hu

Cryptography is the most essential component for securing systems and communications and it is generally used in embedded systems used for industrial and defense applications. The most frequent cryptographic operations such as encryption/decryption, message-authentication, and digital signatures rely on keys that must be kept in secret within a device and protected from revealing.

Modern cryptographic algorithms, when used with appropriate-sized keys, are designed to resist all known attacks where the attacker can observe or manipulate the inputs or outputs of the algorithm, but does not have any other information about the secret key or the execution of the algorithm. In practice, however, an attacker who has access to the device that is performing a cryptographic algorithm can easily obtain additional information about the operation, beyond just the inputs and outputs. For example, even a remote attacker can obtain an approximation of the time needed to perform the operations. Moreover, if an attacker is physically close to the device, he could also measure the power consumed by the device or its EM emissions while it is working. These additional sources of information about cryptographic operations are known as side-channels.

In the 1990s Kocher et al. [1, 2] showed that these kind of side-channels contain enough information to easily extract the secret key from naïve implementations of all cryptographic algorithms. They also introduced some basic techniques for protecting cryptographic implementations from such attacks.

Time-based attacks form one of the most important classes of side-channel attacks. They exploit small differences in execution time to extract secret information from systems. Commonly discovered sources of timing leaks include data dependent execution times, early exits or cache access times. There are many examples of possible timing attacks against cryptographic algorithms. Kocher in [1] introduced timing attacks against standard cryptosystems such as Diffie-Hellman, RSA, and DSS. Cache collision timing attacks against AES executing on modern processors were demonstrated by Bonneau and Mironov in [3]. Brumly and Boneh [4] demonstrated that timing attacks against networks were possible remotely too. These examples show that cryptographic algorithms and protocols may be vulnerable to timing attacks, even when operating on modern hardware.

If we have a cryptographic algorithm given by its source code, we can determine the sequence of the cache accesses dependent on the secret key. We can slightly modify this sequence by adding new entries. These do not correspond to any meaningful instruction in the algorithm, however they might reduce the information leakage. So with the help of some fake cache accesses we can create a countermeasure against cache attacks.

In this paper we consider cache-based side-channel attacks and introduce a method to automatically defend against them. We developed a machine learning model which builds protection for any cryptographic algorithm vulnerable to cache attacks.

References


Train multi object tracker on similar instances

Áron Fóthi
Department of Artificial Intelligence, Eötvös Loránd University, Budapest
fa2@inf.elte.hu

The two fundamental computer vision methods for behaviour analysis of laboratory animals are instance segmentation [1] and tracking [2]. The accuracy of these methods should be almost perfect because uncovered mistakes during tracking may lead to wrong overall conclusions. We aim to solve this problem with our Cluster RCRNN model. The model is based on the Mask R-CNN [3] architecture, we replaced its mask head with a cluster head to able to separate multiple instances from a single region of interest (ROI), and extended it with a recurrent neural network (RNN) layer to handle temporal sequences. Our model was initially trained with a naïve training method and achieved comparable results. In this paper, we aim to exceed these results and minimize the possibility of failures during tracking by applying various advanced training techniques.

RNNs [4] are well suited for handling sequential data, like video, because they utilize hidden states, where information about past frames can be stored. Despite the fact that the RNN architecture is powerful, it may cause two major problems: exploding gradients and vanishing gradients. During training, our model suffered from exploding gradients which refers to the problem that the gradients get too large during back propagation. This makes the model unstable [5] and causes inferior performance. To alleviate this problem, we studied the effect of the following methods:

- Gradient clipping [6]: This is the most common solution against exploding gradients. During backpropagation if the gradients become too large, we simply rescale it.
- Learning rate reduction: By lowering the learning rate, exploding gradients occur less frequently but it leads to sub-optimal training time.

We have achieved the best results when both method was used: we set the max norm of the gradient clipping to 1 and reduced the base learning rate to 0.001, and to eliminated the drawbacks of low learning rate, we pre-trained the network on 1 length sequences.

We trained our model on two datasets, a larger one which only contained automatically annotated samples and a much smaller one which was annotated by humans. The automatically annotated dataset was gathered by a software that could only annotate videos where the rats were separated so the segmentation was possible by a simple computer vision algorithm. However, the number of these frames was almost infinite, in practice the usability was limited in terms of training. By augmentation we made various training examples but the models trained on these artificial images were lacking behind compared to those that were trained using real images. The smaller, hand annotated dataset contained 17 clips. This alone would not be enough for standalone training, but it was sufficient for fine tuning [7] our models after they were pre-trained on the automatically annotated samples.

By solving the exploding gradient problem and pre-training on artificial images, our results improved drastically: Our model reached 0.667 Average Precision (AP) and 0.369 Average Recall (AR), compared to 0.376 AP and 0.259 AR of our initial model, which is considerably better than the 0.392 AP and the 0.245 AR reached by MaskTrackRCNN [2] on synthetic data. After fine tuning the models on the hand-annotated dataset, we evaluated them by counting the number of track switches. The original model made 5 mistakes, while by using our proposed training method this number was reduced to 2.

These results proved that our Cluster RCRNN model is able to achieve high accuracy in real world tasks with attentive training strategy.
Acknowledgment

Áron Fóthi was supported by the Hungarian Government and co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002: EFOP-3.6.3-VEKOP-16-2017-00002, Integrated Program for Training New Generation of Scientists in the Fields of Computer Science) and the ELTE Institutional Excellence Program of the National Research, Development and Innovation Office (NKFIH-1157-8/2019-DT) and by the Thematic Excellence Programme (Project no. ED 18-1-2019-0030 titled Application-specific highly reliable IT solutions) of the National Research, Development and Innovation Fund of Hungary, respectively. We thank Árpád Dobolyi and Dávid Keller for the database. We thank László Kopácsi for his careful help in composing this paper.

References

[1] Hsu, Yen-Chang and Xu, Zheng and Kira, Zsolt and Huang, Jiawei 2018 Learning to cluster for proposal-free instance segmentation International Joint Conference on Neural Networks IEEE


A Pragmatic Static Analysis Method with An Applied Model of Formalized Concurrency

Endre Fülöp, Norbert Pataki

Department of Programming Languages and Compilers
Eötvös Loránd University
Budapest, Hungary

gamesh411@gmail.com, patakino@elte.hu

Concurrency in high-level programming languages is supported by many built-in and library structures [6]. These allow the programmer to encode the control flow in a manner that is easier to understand and reason about informally compared to low-level assembly and strict formal descriptions [2]. However, the complexity of multiple solutions and the need for performance lead to solutions that can result in unpredictable behaviour [5]. One common way to formalize concurrency and to express the set correct executions allowed by the semantics of a programming language implementation is the relation model of memory access events [1].

Dynamic analyses of multithreaded C++ applications are available. For instance, 3 has a checker regarding multithreaded problems [3]. However, dynamic analysis of multithreaded applications can return non-deterministic result. Programmers need a strict and deterministic solution to detect problems in multithread programs.

In this paper, the already formalized memory model of C++ is looked upon from a practical point of view. Real-world software solutions employ various concurrency-related structures. Non-concurrent language semantics in case of C++ further aggravate the issue of getting meaningful results in such codebases [4]. The authors introduce a practical detection schema for concurrency constructs of the C++11 memory model, investigate the computational complexity of algorithms used to generate and check the sets of relations that describe the appropriateness of a software implementation. A comparison to dynamically typed language Python3 is also given to highlight the advantages and disadvantages of concurrency and the regarding static analysis methods in both programming languages.

References


Modeling Resource Allocations in Cloud Deployment with P Colonies

Endre Fülöp¹, Norbert Pataki¹, Csaba Rotter²

¹Department of Programming Languages and Compilers, Eötvös Loránd University, Budapest, Hungary
²Nokia Bell Labs, Budapest, Hungary

gamesh411@gmail.com, patakino@elte.hu, csaba.rotter@nokia-bell-labs.com

P colony is a mathematical model of computation [5]. Many different variants of P colonies have been developed. Basically, in P colonies, multisets of objects are used to describe the contents of agents and the environment. This environment is shared among the agents and is both a communication channel for the agents and storage for the objects. The environment has a significant role in the agent’s synchronization. The objects are processed by a finite set of programs associated with the agent. Each agent has the same number of objects during the functioning of the agent community. Agents’ program can change the objects present at their disposal and can exchange some of their objects with objects present in the environment. A formal definition of P colony is available [4].

P colonies can be used in different simulations and modeling [2]. Approaches for 3 operations and other simulations are available [3]. Moreover, the application of P colonies can improve the specification and simulation of parallel and distributed workflows.

Modern applications take advantage of 1 technologies and are run in virtual machines. On the one hand, using virtual machines improves resource allocations – even in a dynamic way. On the other hand, 1s have hardware limitations, and applications cannot be started infinitely. Virtual machines resources defined in terms of vCPUs, size of memory, IP addresses, volume, etc. Every 1 has its limitation regarding these resources.

In this paper, we introduce an approach to utilize P colonies for modeling resource allocations and application deployments in 1s. We present our simulation software that is able to visualize the modeling processes of a specified P colony. We show how a P colony can be defined in our system and how the mapping of P colony and 1 resources can be determined. Moreover, we define P colonies whose aim is 1-based resource allocation.

References

Sound classification with transfer learning
Napsugár F. Gáti, Attila Kiss
Department of Information Systems, ELTE Eötvös Loránd University, Budapest, Hungary
h93579@inf.elte.hu, kiss@inf.elte.hu

Abstract
Machine learning is used nowadays in several fields, one of them is speech recognition. Controlling devices with our speech, recognising the music we hear, creating subtitles for a video based on the audio, we use more and more features related to speech and voice recognition. In general, we are glad to use such technology - in those cases when it provides extra functionalities - however, it may be bothering when we would rely on machine learning applications and they do not work correctly. For example, if we call a customer service which uses machine learning to understand what our problem is, we really would like to get connected with the right person. Among the different fields of machine learning image recognition is the most commonly used on a daily basis. Especially in real life scenarios; even security systems apply this technology. The purpose of our research is to combine speech recognition with image recognition.

Image recognition has grown so effective, that we can find a lot of already trained and well working models. During the process of learning it is self-explanatory for human learners to transfer knowledge from one task to another. We can recognise relation between two problems and use this information in benefit of a new task. Transfer learning is a sub-field of machine learning and it works similarly. The main idea is to use the knowledge learnt by a model for solving another problem. Collecting labeled data is time consuming, as well as training the model. Transfer learning can save us a lot of time and work, while using a pre-trained model can improve our algorithm effectiveness. A survey\cite{2} in 2009 defined the notion of transfer learning and since then, we can find a great number of research papers on the different use cases of this methodology. Prior to applying the transfer learning, we must consider the relation between the source and target domains, and between the source and the tasks. The more related the problems are, the more helpful a model could be for the new task.

In this paper we are going to explain how to transform a sound classification problem into an image classification problem. We will compare a deep learning model trained from scratch with a neural network model which starts the learning from a pre-trained model. The first version of Speech Commands dataset \cite{3} will be used for the training and fine-tuning, the second version which contains more data, will be used for evaluation. MobileNet\cite{1} model was trained on natural images, the features learnt from these images could be useful to improve the model for recognizing different spoken words.

Acknowledgment
The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

References
\begin{enumerate}
\end{enumerate}
On some subclasses of holomorphic functions whose derivative has positive real part

Eduard Ștefan Grigoriuc
Department of Mathematics, Faculty of Mathematics and Computer Science, Babes-Bolyai University, Cluj-Napoca, Romania
eduardgrigoriuc@yahoo.com

Starting from MacGregor’s paper [5], we discuss about normalized holomorphic functions whose derivative has positive real part. This class, denoted $R$, is also important since any function in $R$ is univalent (in view of a result due to J.W. Alexander, see [5]). For details about class $R$ and univalence, one may consult [1]. Regarding to the class $R$, in the first part of the paper, we present a general distortion result (some upper bounds for the modulus of the $k$-th derivative). The second part of the presentation contains some remarks on the functions whose derivative has positive real part of order $\alpha$, with $\alpha \in [0, 1)$. Here we present some coefficient estimates for the class $R(\alpha)$, a growth and distortion theorem for the same class and a general distortion theorem for the class $R(\alpha)$.

References

System security is a major concern of current software engineering. Attackers exploiting weaknesses in the system can cause major financial and non-material damage. Due to the enormous amount of legacy and newly written code and the large extent of possible vulnerabilities it is natural that software engineers are looking for advanced engineering practices to detect possible weak points of the system. Coding conventions, code reviews, intensive testing, as well as applying static and dynamic analysis tools are among the preventive measures.

However, the price factor, especially the maintenance cost is not the same for all these methods [1]. Writing and maintaining comprehensive tests for a large software system is time consuming, expensive and reaching the full test coverage for real world applications is almost impossible. Code review – when applied – is a great way to avoid security related bugs in the early phase of the development. However, thorough code review requires experts who are familiar both with the problem domain and the details of software security. Such experts are not always available and their manual code review process is also time consuming. Thus the code review process soon become a bottleneck of the development and companies tend to avoid it.

Contrary to testing and dynamic analysis methods static analysis works at compile time, based only on the source code of the system, and does not require any input data. It is a popular method for finding bugs and code smells [2]. Static analysis unifies the advances of code review and the automated tools: they can multiply the knowledge of experts coded into certain “checkers” – small programs to detect a certain vulnerabilities and execute them many times avoiding the bottleneck of the availability of the experts. Static analysis is most popular for programming languages with static type system, like C, C++ and Java. A number of commercial [3, 4] and open source tools [5, 6, 7] exist with considerable large developers’ community to support these languages [8].

The situation is, however, a bit different for dynamically typed programming languages. Python, one of the most rapidly emerging programming languages with dynamic type system. Python is very attractive implementing Machine Learning and Cloud based applications among others, and it is quite a long time the third most popular language by the Tiobe index [9]. Python has a rapidly increasing static analysis support [10, 11] and we can distinguish between generic and security specific 5 tools. Programs like PyLint, Pyflakes are used to detect various programming errors and code smells, while e.g. Bandit and PyT are more specialized for recognizing security related vulnerabilities. Most of these 5s utilize the Abstract Syntax Three (AST), however there are new directions to apply symbolic/concolic execution too.

In this work we refer to the Fortify Taxonomy to get examples of some of the most common security vulnerabilities present in Python. Using these examples, we initiate a discussion about the existing python static analysis tools’ capability of catching the given security vulnerabilities. Among others, we evaluate the various tools over error types like SQL injection, header manipulation, issues with Dynamic Code Evaluation, Command injection, Resource injection and File uploads.

Based on our evaluation it seems that currently no single static analysis tool can give enough safety against vulnerabilities, therefore the actual best option developers have to protect their code base is to apply multiple tools simultaneously. Even though this solution increases the cost factors, catching security issues in the early phase of the development can significantly decrease the overall maintenance cost of the software.
<table>
<thead>
<tr>
<th>#</th>
<th>Security vulnerability</th>
<th>Bandit</th>
<th>PyT</th>
<th>PyLint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SQL Injection</td>
<td>+/-</td>
<td>+/-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Header Manipulation: SMTP</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Dynamic Code Evaluation: Unsafe Pickle Deserialization</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Cross-Site Scripting: Content Sniffing</td>
<td>-</td>
<td>+/-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>Command Injection</td>
<td>+/-</td>
<td>+/-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>Resource Injection</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>File Disclosure: Django</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>File Upload</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: Python static 5s capabilities to detect security vulnerabilities

References


Discovering Shared Variables for Comprehension of Multithreaded C++ Programs

Attila Gyén, Norbert Pataki
ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Informatics, 3in Research Group, Martonvásár, Hungary
gyenattila@gmail.com, patakino@elte.hu

Today’s software needs to meet more and more demands in parallel with the ever-increasing industry and user expectations, preferably with as little extra hardware load as possible so that their execution time does not increase or increases only minimally and it is very important to maintain their consistency. For this reason, the purely sequential programs alone are not strictly out of place because of extra computing capacity growth. In general, there are two possible approaches to this problem:

1. Purchasing bigger and more powerful hardware – often involves high costs of integrating and may involve expanding the server farm.
2. Writing parallel programs – requires more attention from developers side, sometimes they are written by more experienced programmers, there is no necessary need to purchase new additional physical elements.

The second solution is often preferred, but an organization may not always be able to afford to move an existing complex program to a more modern environment. But in most cases, this proves to be a more effective solution.

Fortunately for developers, more and more programming languages support writing parallel programs. The strength of parallel programs is also one of its weaknesses, which is the use of shared memory space. To do this, the following things need to be recognized:

1. High-level programming language instructions do not count as elementary levels at the processor machine code level.
2. Thread switches occur between two (elementary, atomic) machine code instructions.
3. Thread switches occur at unpredictable moments.

However, compilers, static analyzers and code comprehension tools are typically not aware of multithreaded executions. This means that the programmers have to deal with the problems come from concurrency by their own. For instance, determining that a given variable can be modified by several different threads in a non-deterministic order is not a trivial task and there is currently not a written language element or compiler switch that can clearly determine whether there is a risk.

Our goal is to develop a tool that can be used to determine as accurately as possible whether different threads use a shared memory area. Our motivation is twofold. Firstly, this tool supports code comprehension of multithreaded programs. Inexperienced programmers may understand the architecture and design choices of multithreaded C++ applications deeper. Secondly, this comprehension can help the developers when debugging and fixing incorrect multithreaded applications because a shared variable can be read or assigned from different threads corruptly. To determine this, it is necessary to define the variables used by the program and whether their values can be modified by different threads running in parallel.

Initially, we prefer the C++11’s standard thread class, focusing on the standard C++ programming language. The Clang compiler has a switch to generate the abstract syntax tree.

---

7The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).
(AST) and then filter out data from AST that may be relevant to us, such as whether the variables were passed by value or reference to the function called in the thread or whether the variables are used within the function called by the thread and how they are used, such as changing their value with a ++ operator or appearing on the right of a value assignment.

During development, we encountered various different difficulties, such as using huge amount of AST dump generated by Clang, which in turn is essential if we want to create threads. Furthermore, we need to filter out what is important to us from these many lines, which is only a very small fraction of all this. Then, using different tokens used in AST, the remaining code must be parsed and the data filtered out, such as function and / or variable declarations, input parameter type, parameter passing.

The goal is to get to the point where this tool cannot only work with the `std::thread` class, but also be able to recognize a thread launched from lambda expressions, for example, or even examine the threads of third-party libraries, and then get there in the future to able to know the threading of not only for C++, but also other programming languages and to point out the memory areas of the program that can be modified in a non-deterministic way for parallel programs and cause errors.

References


Synthesizing Same-Language Summaries for Symbolic Execution

Gábor Horváth, Norbert Pataki
ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Informatics, 3in Research Group, Martonvásár, Hungary
xazax@caesar.elte.hu, patakino@elte.hu

In inter-procedural static analysis, the analyzer considers the calling context when evaluating a function call. This approach makes it possible to find bugs that span across multiple functions. For example, the same object might be deleted in two distinct procedures. In order to find those issues, the analyzer engine requires information about both the calling context and the callee. Unfortunately, the implementation of the callee might only be available in a separate translation unit or module [4]. In these scenarios, the analyzer either makes some assumptions about the behavior of the callee (which may be unsound) or conservatively creates a program state that marks every value that might be affected by this function call. In this case, the marked value becomes unknown, which implies a significant loss of precision.

To mitigate this over-approximation, a common approach is to assign a summary to some functions. Each time the implementation is unavailable, use the summary to analyze the effect of the function call. Note that an analyzer engine might prefer summaries over implementations even when implementation is available to improve the analysis’ run-time performance. These summaries are, in fact, approximations of the function implementations that can be used to model some behavior of the called functions in a given context [2]. Those summaries can be synthesized from the source code during a multi-pass (or on-demand) analysis or written manually by a developer [5]. The contents of a summary usually depends on the analysis. In abstract interpretation, we can sometimes derive the summaries from the transfer functions. In symbolic execution, it is common to use (precondition, postcondition) pairs for paths in the function [1]. It is not settled, however, what contents should a symbolic execution summary have [3]. The design space of representing summaries is enormous. We can use a SMT formulae, state machines, graphs, tables, same language representation and more. We could even come up with hybrid representation. The same language representation can be textual or an intermediate representation of the language like token stream or abstract syntax tree (AST). The most proper way to represent summaries remains an open question.

This paper describes a method for summarizing C/C++ functions automatically in C/C++ itself to support symbolic execution. It is challenging to use source language representation efficiently due to the compilation model of C/C++. We propose an efficient solution that circumvents this problem.

References


*The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).


Fractional operators in relativistic quantum mechanics: the square-root Klein–Gordon equation

Ferenc Izsák, Béla J. Szekeres

Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University
H-1117, Budapest, Pázmány P. stny. 1/C, Hungary

Department of Numerical Analysis, Faculty of Informatics, Eötvös Loránd University H-1117,
Budapest, Pázmány P. stny. 1/C, Hungary

izsakf@caesar.elte.hu, szekeres@inf.elte.hu

Abstract

In this work, we investigate the square root Klein–Gordon equation. To complete the results of [1], we prove that the equation describes spinless particles and the Hamilton operator of the system is symmetric and positive. We also provide the equation on bounded domains, especially for periodic boundary conditions on multidimensional cubic domains. Finally, we construct a convergent numerical scheme based on the Matrix Transformation Method (MTM).

The Klein–Gordon equation with mass parameter $m$ and potential $V : \mathbb{R}^n \to \mathbb{R}$ for the wave function $\Psi(t,x) : \mathbb{R} \times \mathbb{R}^n \to \mathbb{C}$ reads as

$$\left(\ih \frac{\partial}{\partial t} - V(x)\right)^2 \Psi(t,x) = \left(m^2 c^4 - \hbar^2 c^2 \Delta + V(x)\right) \Psi(t,x), \tag{14}$$

where $\hbar$ is the reduced Planck constant and $c$ is the light speed. This is an important equation in quantum mechanics, being the first attempt to link the theory of relativity and quantum physics. But it fails to fulfill the postulates of quantum mechanics. One way to resolve this failure is the method of Dirac. The Dirac equation describes the spin-$1/2$ particles like the electron. Our goal was to take the square root of the Hamiltonian in another way to get a Lorentz-invariant equation (corresponding to the theory of relativity) and to satisfy the postulates of quantum mechanics. We propose the following initial value problem for $\Psi$ in the Schwartz space $S$:

$$\begin{cases} 
\ih \frac{\partial}{\partial t} \Psi(t,x) = \left(\sqrt{m^2 c^4 I - \hbar^2 c^2 \Delta} + V(x)\right) \Psi(t,x), & t > 0, x \in \mathbb{R}^n, \\
\Psi(0,x) = \Psi_0(x) \in S, & x \in \mathbb{R}^n, \\
||\Psi(t,\cdot)||_0 = 1, & t > 0,
\end{cases} \tag{15}$$

which is called the square root Klein–Gordon equation. The spatial operator on the right-hand side of (15) is investigated in the following.

**Theorem 17** Let the operator $\sqrt{m^2 c^4 I - \hbar^2 c^2 \Delta} : S \to S$ defined by

$$\sqrt{m^2 c^4 I - \hbar^2 c^2 \Delta} \varphi := \mathcal{F}^{-1} \left(m^2 c^4 + \hbar^2 c^2 |\vec{Q}|^2\right)^{1/2} \mathcal{F} \varphi, \tag{16}$$

where $\vec{Q}$ is an $n$-length multiindex and all of its elements are 2. With these

$$\left(\sqrt{m^2 c^4 I - \hbar^2 c^2 \Delta}\right) \left(\sqrt{m^2 c^4 I - \hbar^2 c^2 \Delta} \varphi\right) = \left(m^2 c^4 I - \hbar^2 c^2 \Delta\right) \varphi, \tag{17}$$

and the operator $\sqrt{m^2 c^4 I - \hbar^2 c^2 \Delta}$ is symmetric and positive.

To complete the results of [1], we prove that equation (15) describes spinless particles, moreover, one can interpret the probability density function $\rho$ in the usual way with $\Psi \overline{\Psi}$. Finally, we investigate the
equation on (multidimensional) cubic domains with periodic boundary conditions. For simplicity, we assume that $\Omega = (-\pi, \pi)$ and we have to solve the following initial value problem for $\Psi \in H_1$

$$
\begin{align}
\Psi(0, x) &= \Psi_0(x) \in F_1, \quad x \in (-\pi, \pi), \\
\|\Psi(t)\|_{L^2(-\pi, \pi)} &= 1, \quad t > 0.
\end{align}
$$

We prove the following statement for the numerical approximation of (18).

**Theorem 18** For any $\Psi_0, V \in H_2([-\pi, \pi])$. Let $\Psi(t, \cdot)$ the solution of (18) and $\Psi_h(t, \cdot)$ the solution of the semidiscretized problem using the MTM. Then the following inequality holds

$$
\|\Psi(t, \cdot) - \Psi_h(t, \cdot)\|_0 \leq hC\|u_0\|_2.
$$

We will also confirm our results with numerical simulations.

**Keywords:** fractional Laplacian, fractional derivative, Matrix Transformation Method, Square Root Klein–Gordon equation

**References**

Region-Based Distributed Hash Table for Fog Computing Infrastructure

Mohammed B. M. Kamel a,b, Peter Ligeti a, Christoph Reich b

a Department of Computer Algebra, Eotvos Lorand University, Hungary
b Institute for Data Science, CC and IT-Security, Hochschule Furtwangen University, Germany
mkamel@inf.elte.hu, turul@cs.elte.hu, christoph.reich@hs-furtwangen.de

Abstract

In this paper, Region based Distributed Hash Table (RDHT) is proposed that can used in fog computing infrastructure to create an overlay of fog nodes divided logically into multiple regions based on their physical locations. RDHT generates a single overlay and it can be generated without specific organizing entity or location based devices.

Introduction

Distributed Hash Tables (DHT) create an overlay by assigning a seemingly unique identifiers to the participating nodes. These identifiers are generated by feeding parameters of the nodes (e.g. IP addresses) to a hash function, and the output is used as the identifiers of the nodes. Depending on the identifier, each node is located in a specific location in the overlay. Due to the randomness feature of the hash functions, the output of the relatively close parameters in the input range might not be close in the hash space. While this property is required to ensure the random and uniform distribution of nodes and the stored data in the overlay, but adopting the original DHT technique in fog computing infrastructure might cause that two adjacent fog nodes be resided in two far locations in the overlay. As a result, while adopting DHT in resource discovery [1, 2] removes the centralized entity, but might map the close fog nodes to distant nodes in the resulted space. Therefore, this issue causes latency in communication of fog nodes.

Comparing to cloud computing infrastructure that sends the traffic to a centralized cloud data center, the fog nodes in the fog computing infrastructure try to keep the data as close as possible to the origin source of data. Therefore, while adopting DHT as a structured p2p scheme in fog nodes has some advantages such as scalability and functionality without involving any centralized entity, but DHT might cause the data to be stored in a far fog node. The main contribution of our paper is proposing a Region-based Distributed Hash Table (RDHT) overlay to keep the data as close as possible to the origin of the data. The proposed model generates a single overlay, can operate without specific location based devices, and add no extra local overhead comparing to traditional DHT overlays.

Region Based DHT

The RDHT overlay consists of $2^g$ regions with a maximum of $2^d$ nodes in each region. A node in the DHT is identified by a $(g + d)$-bits identifier. The first region is called the public region which nodes can join regardless of their locations. In addition to that, a new node joins local regions in the
RDHT based on its location. The regions are created by feeding the information of the locations to a hash function. The locations can be represented by any unique form of information such as human-readable names of regions or a specific prefix of latitude/longitude data. The remaining d-bits of the identifier of a node are generated by hashing the information of the node (e.g. its IP address). Figure 1 illustrates RDHT.

![Figure 1: Region-based Distributed Hash Table](image)

The pair of key/value is stored in the RDHT either in the public region or the specific region based on the location of the data. Storing the pair of data in the specific region that the issuer node belongs to guarantees that the pair of data will be stored in a node that physically belongs to the same region of location. This ensures that the data is stored in a close fog node without adding additional overhead to maintaining the store and lookup processes. The put interface takes three input parameters: key, value and region, and store the value in a node in the public region or a close node resides in local region based on the request of the issuer. The get interface takes a key and a region parameters and retrieves the specific value corresponding to the given key from one or more regions based on the request of the client. Similar to traditional DHT overlay, the public region with $n_p$ nodes requires maximum $\log(n_p)$ steps to reach the required value. The region based lookup with $n_i$ nodes in the region $i$ requires maximum $\log(g.n_i)$ steps to reach the required valued in an overlay with $2^g$ regions.

References


Semantic Encoder Tasks for the Hungarian Language

Attila Kántor\textsuperscript{1}, Attila Kiss\textsuperscript{1}, László Grad-Gyenge\textsuperscript{2}

\textsuperscript{1} Eötvös Loránd University, Budapest, Hungary, Faculty of Informatics, Department of Information Systems
\textsuperscript{2} Black Sheep Intelligence

atis06|kiss@inf.elte.hu, l@bsi.ai

Natural Language Processing (NLP) is a sub-field of Artificial Intelligence (AI) which algorithms are able to gather fruitful knowledge by extracting information from human-written and spoken text. NLP involves speech recognition, text processing and several other unsolved problems. The applications of NLP algorithms directly affect our life through for example personal assistants, recommender systems and chatbots.

The recent advances in the field of semantic encoding show a considerable progress and also draw attention to such techniques. The primary goal of these methods is to map text of human-written natural language into a semantic vector space. By semantic we mean that the vectors assigned to the text should represent its meaning. Semantic encoders are typically based on recurrent neural networks and are trained on several tasks also involving transfer learning. These methods are eager on training data, and due to the lack of labelling are typically trained on auto-annotated datasets. However, in the case of widely spoken languages, explicitly annotated data is also available, e.g., the SNLI [1] dataset. Since semantic representation algorithms are frequently applied tools of NLP, the most accurate results are achieved through methods published by companies researching and developing software products in these fields (e.g. BERT [2], InferSent [3] and USE [4]), but similar techniques are also receiving a lot of attention in academia.

In this paper, the training and the performance of a semantic encoder is presented for the Hungarian language. Hungarian is a not widely spoken language, so the amount of available linguistic resources is limited. Since the introduced method is applicable in the case of Hungarian, it could be adopted to any small or medium language. The paper also discusses the training data, the concrete BERT-like training tasks and the BiLSTM based neural network architecture of the encoder. Moreover, a novel Hungarian dataset for sentiment analysis is also introduced regarding the evaluation process of the trained model.

Acknowledgment

The project has been supported by the European Union, co-financed by the European SocialFund (EFOP-3.6.3-VEKOP-16-2017-00002).

References


A calculus of single substitutions for simple type theory

Ambrus Kaposi, Norbert Luksa

Department of Programming Languages and Compilers, Eötvös Loránd University
{akaposi|luksan}@inf.elte.hu

Church’s simple type theory (or simply typed lambda calculus) is the main foundational language for describing type systems for programming languages. Polymorphic type systems [3] and dependent type systems [6] both build on the foundation of simple type theory. Traditionally [4], simple type theory is described as a relation on preterms together with an operational semantics given as a binary relation on preterms. Typing and operational semantics are usually related through preservation: if a term has type $A$ and is related to another term by the operational semantics, then this new term also has type $A$.

Algebraic descriptions of type theory [2] were recently back-ported to simple type theory using the term simply typed category with families (CwF) [1]. In an algebraic description, there are only well-typed terms which are quotiented by definitional equality, hence type preservation is trivial. In this work we investigate an alternative description of simple type theory, which does not build on a category naturally giving rise to parallel substitutions, but on single weakenings and single substitutions. In our calculus, parallel substitutions are not even mentioned. Our calculus is close to the usual informally written syntax of simple type theory. The main substitution operator takes terms $t : \text{Tm}(\Gamma \Theta B \Theta+\Delta)A$, and $u : \text{Tm}\Gamma B$ and results in a substituted term $t[\Gamma,u,\Delta] : \text{Tm}(\Gamma ++\Delta)A$ where the $++$ operator concatenates contexts.

Compared to simply typed CwFs, our calculus has fewer operators (four fewer as we don’t need to talk about parallel substitutions) but six more equations (as we need to describe interactions of single substitutions and weakenings). Contextual CwFs have the property that every context is built out of a finite number of types. We show that models of our calculus are equivalent to contextual simply typed CwFs with a base type and function space. We expect that our calculus is close to calculi with only normal forms such as hereditary substitutions [5]. We plan to investigate its generalisation to dependent types, obtaining a telescopic variant of CwFs.

References


Modelling to Program in the case of Workflow Systems
Theoretical background and literature review

Meriem Kherbouche, Bálint Molnár

Information Systems Department, Eötvös Loránd University of Budapest, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary

(meriemkherbouche, molnarba) @inf.elte.hu

Abstract

The Model to Program approach is a recent scientific research direction that has its basis in the previous researches. Several sets of tools have been developed in the past decade that can be applied in novel way to program generation from models. This paper will explore the research questions, approaches, and the related literature. The future research direction will be outlined within a well-defined framework.

The development of the semantics that describes the underlying operational semantics achieved higher accuracy in the case of UML Activity Diagram, Business Process Modeling Notation, Workflow description. These results makes it possible that preparation of program generation from models can be more precise. The interpretation and validation of expressions in logical languages advanced and became tractable in practical view. Therefore, the model checking is feasible at level of semantics. Besides, the logical validation, the mapping and matching of the syntactic element that may need for a full-scale program generation can be carried out by using modern algorithm.

The used sources: [1], [3] [2]

Acknowledgement

The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and by no. ED_18-1-2019-0030 (Application-specific highly reliable IT solutions) program of the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme

References


Our objects of interest are generative autoencoders, that is, autoencoders where the decoder can be used as a generator when samples from some prior distribution are fed to it. More specifically, we are interested in the Wasserstein Autoencoder (WAE) class of deep learning models [5]. Tolstikhin et al’s original formulation states that if the pushforward of the true data distribution is equal to the prior, then the Wasserstein distance between the true data distribution and the generated distribution is equal to the reconstruction error. Thus, by applying Lagrangian relaxation, the loss function for these models is the reconstruction error plus a penalty term for matching the aggregated posterior to the prior. This penalty term can be seen as a statistical test verifying that the pushforward is indeed close to the prior.

In all incarnations of this idea that we are aware of, such as Adversarial Autoencoders [2], WAE-MMD [5], MMD nets [1], or Sinkhorn Autoencoders [3], the input for this statistical test is the latent image of the minibatch (latent minibatch for short). However, as it is already hinted by experiments published by Rubinstein et al. [4], the size of the minibatch may strongly affect the performance of the model.

We argue that in the WAE class of models, the minibatch size (which is typically in the range of 50-200), is not large enough compared to the dimension of the latent space, which means that the statistical test is too weak to guarantee a good match between the pushforward and the prior distribution. We propose an optimal transport-based generative model from the Wasserstein Autoencoder family of models, with the following innovative property: the optimization of the latent point positions takes place over the full training dataset rather than over a minibatch.

Our baseline model is the Sinkhorn Autoencoder [3], which operates on the latent embedding of the mini-batch. We compare our Global Sinkhorn Autoencoder model with the “local” Sinkhorn Autoencoder baseline on natural and synthetic datasets, on several evaluation metrics. One highlight of these comparisons is that the global model consistently and significantly outperforms the baseline model on a semi-supervised classification task. Below we describe this setting in more detail.

The MNIST dataset can be seen as a balanced mixture of 10 disjoint image datasets, one for each digit. Thus it is reasonable to consider a 10-mixture of Gaussians as the latent prior distribution for such a dataset, as was already done by [2]. (Setting the pairwise KL-divergences of the Gaussians to a high value, reflecting the fact that the class ambiguity of the data distribution is small.) In this setup, a goal we might set for our encoder is that the pushforward of the data distribution of a single image class should be close to one of the Gaussians. Conversely, the decoder pushforward of a single Gaussian should be close to the data distribution of a single image class. We define several evaluation metrics that quantify this beneficial property,
and we demonstrate that the Global Sinkhorn Autoencoder outperforms the baseline Sinkhorn Autoencoder on each of these metrics in various settings.

References


The Compressed Program Dependence Graph  
Attila Kiss, Ádám Kiss  
Department of Information Systems, ELTE Eötvös Loránd University, Budapest, Hungary  
kiss@inf.elte.hu, akiss@student.elte.hu

Abstract  
Systems and programs can be represented with graphs. With graph properties during static analysis, we can find the critical parts of the system without even running a single line of code. Static analysis has many advantages compared to runtime analysis: our analysis is not dependent on the input and we can cover a much larger part of the possibilities. This input-invariancy is a very important key feature when we test complex systems and programs.

The Program Dependence Graph[2] is a powerful framework for such analysis[1], and was proven for debugging[4]. With that tool, we can trace back a variable’s value and detect possible logical errors described by Fosdick and Osterweil[3], find parts of the code which does not affect the output variable[4].

However we have seen the importance and usage of component-based analysis[5], often we might need a larger analysis on the full program, with each of its components, but originally the model did not support side-effects and recursion, which are common nowadays.

Functions with side effects have obvious effects on parameters, so we must represent these effects in the PDG for data-analysis. In this case, we must insert the whole function for accurate analysis - which makes the graph so large - or only note function call - which leads to a less accurate analysis. Now we represent a method for building a compressed PDG for a large system, which contains the same useful data but has far fewer edges compared to the substitution method.

This new representation method - called Compressed Program Dependence Graph - can handle analysis on both smaller components and the whole system too. It is also compatible with every method created for the Program Dependence Graph, while fixes some of the main drawbacks of the original idea. We can now represent side effects and recursion too with this representation method.

In our idea, we represent each called function once regardless of the number of calls. This can be achieved by ”bridge edges” with special IDs in our representation model for each function call. These edges indicate inherited dependence from a previous function call. We will access the called functions’ PDGs from different call positions, and show how can we do it by only generating each function’s PDG once and still tracing the data- (and control-) dependence with lemmas. Also, we will present a method for representing dependence in recursive functions using bridge edges.

Acknowledgement  
The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

References  


Evaluation of a Recursion Aware Complexity Metric

Smiljana Knezev†, Gordana Rakić†, Zoran Budimac†, Melinda Tóth‡, István Bozó‡
†University of Novi Sad, ‡Eötvös Loránd University (ELTE)
dmi.317m.19@student.pmf.uns.ac.rs, goca@dmi.uns.ac.rs, zjb@dmi.uns.ac.rs, tothmelinda@elte.hu, bozoistvan@elte.hu

This paper looks into how existing and the newly developed Overall Path Complexity [9] metric deal with growing complexities of modern programming languages and projects. More and more programming languages are becoming hybrid, using multiple paradigms and it is not rare for several different technologies to be used within a single project. Recursive solutions are widely accepted and used, yet not many complexity metrics take recursion into account. The Overall Path Complexity (OPC) [9] metric was designed to address this problem while being language/paradigm independent. The novel approach of the OPC metric is looking into length of (recursive) invocation chains at the statement/expression level. The OPC metric is introduced as an extension of Cyclomatic Complexity (CC) [5] with the use of Inlined Cyclomatic Complexity (ICC) [9], Length of Recursion (LOR) [9] and Recursive Complexity (RC) [10]. Goal of this research is to evaluate how the OPC metric behaves in the context of Object Oriented language such as Java and to compare the results with results from [9] where Erlang projects were evaluated.

OPC metric was already integrated in the RefactorErl tool [1]. Both small examples and bigger open source projects were evaluated with the use of the OPC metrics for Erlang programming language [9]. Results in [9] showed that in comparison to Halstead (Volume, Effort, Difficulty) Metrics [3], Unique Complexity Metric [6], Maintainability Index [2] and Cognitive Metric [4], the OPC metric looked into new information that has not been evaluated before. In contrast to [9] paper where behavior of OPC metrics was evaluated for a functional programming language (Erlang), here we look into how the metric behaves in the context of an object-oriented language, Java. To achieve this, similar methodology as in [9] was used.

In our work the OPC metric was integrated in the SSQSA platform [7]. One of the main goals of SSQSA platform is to provide consistent static analysis across multiple programming languages [7]. The SSQSA platform has a unique source code representation called enriched Concrete Syntax Tree (eCST) [8] which represents a union of concepts used in abstract syntax tree (AST) and concrete syntax tree (CST). ECST trees are further used as an entry point for other tools within the SSQSA platform. One such tool is enriched General Dependency Network (eGDN) generator [11], [12] which produces software network representing relationships between software entities. EGDN network can be used to generate all possible call chains. With the use of eCST trees and the eGDN network there is enough information to calculate all the mentioned complexity metrics at the static level. Furthermore, all algorithms for calculating metrics and other analysis are implemented on the eCST representation providing consistent results across projects using different languages/paradigms. Calculating metrics and preforming analysis on eCST trees also allows for easy extension of the framework in supporting new programming languages.

Currently, SSQSA platform supports Java 5 projects. Eight Quick Sort algorithm implementations in Java were chosen to be analysed within SSQSA platform. These one class implementations were evaluated using OPC metric and the results were compared with values obtained by using Halstead (Volume, Effort, Difficulty) Metric, Unique Complexity Metric, Maintainability Index and Cognitive Metric. Correlations between the OPC metrics and each of the above mentioned complexity metrics was calculated. It was found that correlation between OPC and...
Maintainability index was negative. Correlations with other complexity metrics values were found to be varying from 0.2 to 0.4. Similar results were observed in [9]. Since this is a small sample, some deviations from results in [9] were expected.

The low correlation between the OPC and other complexity metrics showed that there is new information for Java 5 projects that has not been evaluated by the other complexity metrics so far. This research shows that besides a functional programming language, the OPC metrics brings new information for an object-oriented language too. The research concludes that the OPC metric is indeed language and paradigm independent.

As mentioned before only small sample of one class projects was evaluated. Further research needs to be done to see how bigger Java projects behave. Also there is work being done on supporting newer Java versions for the SSQSA platform. Since SSQSA platform is language/paradigm independent future work will focus on how other languages supported by the platform behave in the context of OPC and other complexity metrics.

References


Diagnosing vulnerabilities with static analysis

Ábel Kocsis, Zoltán Gera, Melinda Tóth
Department of Programming Languages and Compilers
Faculty of Informatics
ELTE, Eötvös Loránd University
abelkocsis@caesar.elte.hu, gerazo@inf.elte.hu, toth_m@inf.elte.hu

Poorly-designed programs often cause errors and failures, which may lead to an irreversible disaster, such as the Ariane 501 satellite launch [1]. A software which is not designed thoroughly is usually more vulnerable, so attackers may manage to steal personal data from customers much more easily. To avoid these mistakes, there are numerous static analyser tools which are able to check the programs without executing them. This is crucial, since the earlier an error is discovered, the easier and cheaper it is to fix it [2]. However, at first sight, it is really hard for a programmer or a company to tell which tool to use.

In this research, we aim to answer this question by comparing the most popular open-source static analyser tools for C and C++ language. Different comparative studies have been conducted in the field of static analysers before [3, 4]. However, these papers do not consider other factors than the results of a few analysers.

A previous study [5] has examined the reasons for using and underusing static analysis tools. These reasons are tool output, collaboration, customisability and result understandability. Since our study concentrates on open-source tools, two other aspects of comparison have been added: method of analysis and collaboration. To get a full picture of each tool, testing for the most common vulnerabilities have been added to the factors. These aspects have made it possible to examine the tools in-depth and to spot the key features of each tool, analysis or a wide range of available checkers, respectively.

Eleven of the most popular static analysis tools [6], have been compared. These tools are the following: Clang Static Analyzer, Clang-Tidy, Cppcheck, Facebook Infer, Flawfinder, Ikos, RATS, Smatch, Splint, SVF and Yasca. The tools differ from each other in a number of important characteristics. These programs are quite diverse; there are vulnerabilities which can be diagnosed with them, and there are many which may be impossible to spot. In the end of the research, a recommendation has been stated which tool may be suitable for general use and which should not be used.

LLVM Clang Static Analyzer and Clang-Tidy have been found to be one of the best analyzers in this comparison. After that, four new checkers have been developed for Clang-Tidy to diagnose hardly-recognisable vulnerabilities connected to multithreaded C and C++ programs; and another checker to improve Clang Static Analyzer.

References


The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002), and by the UNKP-19-1 New National Excellence Program of Ministry for Innovation and Technology. The research is part of the "Application Domain Specific Highly Reliable IT Solutions" project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme.


[6] Ábel Kocsis: Programok sebezhetőségének felismerése statikus kódelemzéssel, Paper at the Student Association Conference, Faculty of Informatics, Eötvös Loránd University, May 2020, Received 1st prize.
A Static Analysis Approach for Modern Iterator Development

Dániel Kolozsvári, Norbert Pataki
ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Informatics, 3in Research Group, Martonvásár, Hungary
kolozsvari.dl@gmail.com, patakino@elte.hu

The C++ Standard Template Library (STL) is the most familiar library based on the generic programming paradigm. STL provides containers that store a collection of elements and manage the underlying memory allocation (e.g. vector). STL includes many algorithms that are functions for different purposes that operate on range of elements (e.g. count_if). Iterators make connection between the algorithms and containers, therefore the set of algorithms and the set of containers can be extended parallelly in a non-intrusive way [2].

However, STL evolves with every new C++ standard. New algorithms, containers have become standardized in the recent years. Moreover, some elements of the library become deprecated, obsolete and removed [4].

The std::iterator class template had been in the C++ since beginning and has been deprecated in the C++17 standard [3]. This class template’s purpose was to specify the traits of an iterator [5]. Typically, it was a base class of many standard and non-standard iterator class to provide the necessary traits [6]. However, the usage of iterator is straightforward and fits into the object-oriented programming paradigm. Many non-standard containers offer custom iterators because of the STL compatibility [1]. Using this base class does not cause any weird effect, therefore usage of iterator can be found in code legacy.

There are some reasons of iterator’s deprecation. The template parameters of iterator are not explicit, therefore it can be hard to figure out which argument what specifies. A more explicit way to define these inside the custom iterator class with using declarations or typedefs.

On the other hand, this approach is not sophisticated enough, new constructs and tools should be utilized.

In this paper, we present a static analysis approach to assist the development of iterator classes in a modern way in which the iterator class template is not taken advantage of. We utilize the Clang compiler infrastructure to look for how the deprecated iterator classes can be found in legacy code and present an approach how to modernize them.

References


The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).
Gview: Visualising software dependencies in order to support code comprehension

Mátys Komáromi, Melinda Tóth, István Bozó
ELTE, Eötvös Loránd University
Faculty of Informatics
Department of Programming Languages and Compilers
makom789@gmail.com, toth_m@inf.elte.hu, bozo_i@inf.elte.hu

It is always a great challenge to maintain industrial-scale software. It requires to fully understand and be aware of the different components and their connections to avoid introducing software errors. Aiding the process of software maintenance by visualisation is a very timely topic, as it is easier for humans to understand visualised information. In our paper, we introduce Gview, a new tool for interactive graph representation. The presented graph is interactive and utilises the GPU to speed up layout generation. We integrated Gview with RefactorErl. RefactorErl is a source code analyser and refactoring tool that also supports code comprehension for Erlang. The tool represents the syntactic and semantic information in the Semantic Program Graph, containing a massive amount of nodes and edges as input for Gview.

The visualisation of a software maps a software system and its architecture to a visual representation. The created view can be static, interactive or even animated [1]. The visual representation of a software may improve the productivity of developers, as it supports code comprehension, helps to find inconsistencies and to improve the quality. The software visualisation extracts and combines closely related information of the system. The visualised representation is easier to comprehend than gathering the same information manually from the source code.

RefactorErl [2] is a static source code analyser and transformation tool for Erlang. It aims to support the everyday code comprehension tasks of the Erlang developers. Since presenting the semantic information about the source code is quite natural on a graph, we started the Gview project as a new graph visualisation component for RefactorErl. The main goal was to be capable of rendering the huge Semantic Program Graphs [3] as well.

The main contributions of this work is the introduction of the new Gview [4] that is a new interactive graph visualisation tool, and the extension of RefactorErl that utilises Gview. Gview was designed to utilise the GPU resources, to provide different layout generation mechanisms, to support a generic data transfer protocol and an easy to use interface for different tools. We present the integration of Gview with RefactorErl and some use cases. Although, the tool was designed for software visualisation, its usage has no restrictions. The new integration of Gview and RefactorErl made it possible to visualize industrial-size software components that were not possible with the old Graphviz [5] based implementation of the tool.

References
[3] Zoltán Horváth and László Lövei and Tamás Kozsik and Róbert Kitilei and Anikó Nagyné Víg and Tamás Nagy and Melinda Tóth and Roland Király: Modeling semantic knowledge in Erlang for refactoring, In Knowledge Engineering: Principles and Techniques, Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques, KEPT 4th Application Domain Specific Highly Reliable IT Solutions project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-1.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme.

[4] Mátyás Komáromi: Visualising software dependencies to support code comprehension, Paper at the Student Association Conference, Faculty of Informatics, Eötvös Loránd University, May 2019, Received 1st prize.

Indirect electricity consumption calculation in the product manufacturing

Adrienn Koncz, Attila Gludovátz

University of Sopron, Simonyi Károly Faculty; Eötvös Loránd University, Faculty of Informatics

koncz.adrienn@uni-sopron.hu, gla@inf.elte.hu

We have been doing our research since 2016 at a furniture company. At the beginning of our research we created a cyber-physical system at the company, which is capable of storing, collecting and analyzing energy consumption data of the production machines. This system is special, because it can collect not only the energy consumption data of the machines but also the system can compare the energy consumption with production data. The data is collected by sensors, which get into the building management system via the company’s own computer network. In this building management system, we can also perform calculations. After that, all of the collected and the calculated data get into the company’s big database. In the database we analyze the data with business intelligence system, and we present the results to the management and the other employees of the company.

Through the years we have continuously developed the cyber-physical system, and now we can measure not only the energy consumption and production data of the production machines with the system but also we can measure the energy consumption of the other production support equipment (for example compressors, extractors, fans) in terms of utility. In the case of compressors, we can compare the energy consumption data with the performance data, thereby we can assign the amount of compressed air generated by compressors to the energy consumption. As a result of this, we could create a totally new efficiency indicator into the system.

Our other cyber-physical system development was, that we expand the measurements to the extractors, and we examined their energy consumption too and their efficiency. For this we perform calculations in the building management system that we have already mentioned above. In the plants the extractors belong to more production machine, and one extractor has more exhaust shutters. One shutter in the extractor is assigned to one production machine. We can follow up in the building management system, that one shutter how many cubic meters of air can be extracted per hour, which we divide into minutes and we check every minute that the shutter is open or close. If it is open, it means, that the extractor extracts air via the examined shutter, and of course it means active and real energy consumption. The electricity consumed by the extractor can be added proportionally (as a shutter) to the electricity consumed by the production machine. The extraction is essential on some machines to the actual production implementation, for example, when planing. This process was necessary because the production managers wanted to charge the energy consumption of the extractors to the production machines, thereby they could make direct costs out of indirect costs.

The building management system, which collects the data via sensors, has its own development interface and its own programming language, which is similar to C programming language. We defined scheduled procedures to solve the task in the development environment. These scheduled procedures are the following: we check the condition of the shutters every minute (they are open or close), and we sum up of the opening time of the shutter every 10 minutes. After that, we gear to the open minutes and the permeability of the shutter, and according to this ratio we are able to add to the energy consumption of the extractors to the energy consumption of the production machines. This process is necessary because the consumption of extractors accounts for about 30% of the factory’s total electricity consumption. The extractors generate indirect cost, which can be built during the manufacturing into the final price of the product as direct cost. These calculations and results we store in database too, and we continuously analyze them with the business intelligence system. These calculations and results help to the company’s management to make more effective decisions thus they achieve a more optimal production operation at the company.
References


A Self-Supervised Method for Bodypart
Segmentation and Keypoint Detection of Rat Images

László Kopácsi

Department of Artificial Intelligence, Faculty of Informatics, Eötvös Loránd University,
Budapest, Hungary

laszlo.kopacsi@gmail.com

Keypoint detection and instance segmentation is a critical problem in understanding interactions
between agents on the scene. To address, we have to tackle the difficulties arising from heavy
occlusions of objects. This task requires large set of carefully annotated samples, although there
are solutions that can make annotation faster [1], they still require some form of manual labeling.

We provide a method which can automatically annotate rats without any human effort. An
initial segmentation of objects is achieved via foreground-background segmentation, from which
we annotate keypoints and parts of samples that can be easily separated using various computer
vision techniques. Given the data and their corresponding generated labels, deep neural networks
can be introduced. We use the Mask R-CNN [2] architecture as both the bodypart segmentation
and the keypoint detection (including the instance segmentation) can be addressed with it. To
handle occlusions we study different augmentation methods used in the video segmentation
literature [3, 4].

We evaluate the final method on hand annotated samples using metrics introduced in the
COCO benchmark [5]. We started with an initial average precision (AP) of 53.22% on instance
segmentation, 48.91% on keypoint detection and 9.38% on bodypart segmentation and by training
deep models with various augmentation techniques, we achieved 57.13%, 44.65% and 28.21%,
respectively.

Our contributions are listed below:

• We propose a computer vision based method for automatically annotating keypoints and
  bodyparts from foreground-background segmentation.
• We study the contributions of various augmentation methods used in the video segmentation
  literature.
• Finally, we train deep models on the generated labels in a self-supervised manner to handle
  the heavy occlusions present in the database.

Our results show that automatic annotation of rat images is possible without the need of
manual labeling. The method presented here can be used to analyze the interaction between
instances and to speed up the annotation process if more precise labels are needed. Our method
can be easily extended to handle video sequences via bipartite matching and by incorporating
optical flow estimated by deep neural networks [6]. However, a more detailed analysis of temporal
extension techniques and fusion of the keypoint and the bodypart models are required to achieve
a system that can track rats and study their interactions in an automated fashion.

Acknowledgements

The research has been supported by the European Institute of Innovation and Technology. We
would like to thank Prof. András Lőrincz for the guidance throughout this research. We would
also like to thank Árpád Dobolyi and Dávid Keller for providing the database.

References

dreaming for video object segmentation.” International journal of computer vision (2018):
1–23.

[4] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-
Hornung. “Learning video object segmentation from static images.” In Proceedings of the
IEEE conference on computer vision and pattern recognition (2017): 2663–2672.

Piotr Dollár, and C. Lawrence Zitnick. “Microsoft 5: Common objects in context.” In Eu-

flow using pyramid, warping, and cost volume.” In Proceedings of the IEEE conference on
Detecting lifetime errors of std::string_8 objects in C++

Réka Kovács, Gábor Horváth, Zoltán Porkoláb

Department of Programming Languages and Compilers
Faculty of Informatics, Eötvös Loránd University
Budapest, Hungary
rekanikolett@caesar.elte.hu
xazax@caesar.elte.hu
gsd@inf.elte.hu

std::string_8 [8] is a 8-like data structure that enables fast and cheap processing of read-only strings in C++. Due to its wide applicability and performance enhancing power, std::string_8 has been very popular since its introduction in the C++17 standard. However, one of its often overlooked characteristics may lead to serious memory management bugs: dangling std::string_8s. The problem is that the lifetime of a std::string_8 is not tied to the lifetime of the 8ed string object in any way. It is the user’s responsibility to ensure that the 8 is only used while the 8ed string is live and its contents are unchanged.

In order to avoid memory errors as early as possible, experts recommend using static analysis tools during development [7]. Most of them run in text editors and IDEs with as little effort as clicking a button, and some of them are able to diagnose sophisticated programming errors. One of these tools is the Clang Static Analyzer [5], an open-source tool that comes packaged together with one of the most widely used C++ compilers, Clang [6]. The Clang Static Analyzer is a symbolic execution engine [3] that performs a path-sensitive analysis of a program by tracking program states and building a model of possible execution paths. It is actively used and maintained by tech giants like Apple and Google.

One of the great strengths of the Clang Static Analyzer is its extensibility - developers can use the infrastructure of the tool to find new classes of bugs by writing a new module called a checker. A recently introduced error-finding module called cpp.InnerPointerChecker [4] tackled a very similar issue to the std::string_8 problem outlined above: dangling raw pointers to std::string buffers. Raw pointers pointing to the inner buffer of a std::string are often used for cheap, no-copy string processing similarly to std::string_8s, and they can cause the same kind of memory errors.

Although raw pointers to std::string buffers are very similar in spirit to std::string_8s, they are processed differently by the tool. While raw pointer handling is hard-wired into the engine, std::string_8s are complicated data structures that it does not understand, and thus std::string_8 operations need to be recognized and explicitly modeled in the engine.

This paper describes the details of a module that finds programming errors caused by the incorrect use of std::string_8. The work entails 'teaching' the tool to 'understand' std::string_8 operations, defining steps to detect lifetime errors, constructing user-friendly diagnostic messages, and performing an evaluation of the checker.

References


Random program generation via Lambda terms
Dániel Leskó, Máté Tejfel
Department of Programming Languages and Compilers, Eötvös Loránd University
{ldani,matej}@elte.hu

In modern software technology, testing is still an essential and core part of the development cycle and will be in the coming years too, despite other slowly emerging approaches like proof assistants, formal verification approaches, etc. Thus the need for high-quality test data is still an important issue for every software development team. For "simpler" case there are a lot of mature, automated solution like QuickCheck [1], however, if the expected test data is quite complex and has to fulfill several internal invariants, like if your test data should be a program itself for testing compilers, source code transformation or analyzer tools, one can find much fewer options and off-the-shelf solutions.

A general, two-phase approach will be presented to tackle this problem. The first phase aims to generate random, typed λ terms in closed, β-normal form. The generation is done in a top-down manner and guided with expected size and type information. The generated set of λ terms has a uniform distribution over all correctly typed terms with a given size, ensured by a counting approach [2].

The second phase translates these random λ terms into high-level programs, based on a grammar describing the exact language constructs of the targeted high-level programming language. As the first phase of the generation is also guided by these language constructs (more precisely only with their types) the translation from λ terms to high-level programs is always complete and successful.

This work was originally focused on functional programs and their random generation, but the concept – as a small example will show – could work for imperative languages as well. From the implementation point of view, this solution could be wrapped to form a QuickCheck generator, which would result in easy usability with test frameworks [3].

References
Distributed Storage Pattern

Jianhao Li, Viktória Zsók
Department of Programming Languages and Compilers
Faculty of Informatics
Eötvös Loránd University
lijianhao288@gmail.com, zsv@inf.elte.hu

In this paper, the principles of a new distributed storage pattern are specified. By this novel architecture designing, we provide the system with new features for distributed storage. The proposed pattern is a generalization of the distributed storage system introduced in the master 3 [3] implemented using the Golang language [4] and the RabbitMQ [5] (which supports the AMQP 0-9-1 Advanced Message Queuing Protocol [1]). The pattern specification, implementation and extension possibilities are provided with features and usages.

There are four types of servers in this project. The web server communicates with the clients and it uses the distributed storage services by making HTTP requests to the API servers. The API servers offer the interface of this distributed storage service. The data servers store the file fragments. The monitor server inspects the status of the data servers, records and solves some of the problems in the system.

In this pattern the following principles are valid: the data servers are divided into several clusters. A hard disk space limit can be set up before the data servers run. The file to be stored in this system is cut into fixed number of fragments or into fragments of fixed size so that the data servers can cooperate to store them. When we get or delete a file, the operation of the fragments should be concurrent for efficiency reasons. The get process merges the fragments into the original file using a method defined by the programmer.

The system has stream duplicate mechanism. Based on the file stream which is provided by the web server, the API server can store multiple streams with the same content, at the same time in each cluster of data servers. The system can detect a damage of file fragments of a cluster and it can recover the file according to the content of other clusters with complete file fragments.

The put process uses request and promise system. The data servers in one cluster share a queue of the RabbitMQ which is bound to a RabbitMQ exchange of type direct. Each request in the queue should be received by only one data server of a cluster at the same time. The API server sends store request of a file fragment to a cluster of data servers. The requests are received by the data servers of a cluster in a balanced way.

After receiving a request, the data server will estimate the free hard disk space to decide whether to give a promise to the related API server or to put this request back to the queue. The promise contains a promise token and the address of this data server. The API server that got a promise can use the token of it to make HTTP put request of the file fragment to the related data server. After the data server stored the file fragment, it will discard the related promise. The communication is reduced or avoided (e.g. the status report messages from the data servers) when there are no tasks to be done by the system; therefore, the system behaves lazily.

The get process uses cluster broadcast locating system. The data servers have their own private RabbitMQ queues for receiving the locate requests from the API server. In order to enable the API server to send locate messages at the same time to all the queues of data servers in a cluster, the queues are bound to a RabbitMQ exchange of type direct with same routing key. The broadcast only happens in a cluster instead of all the data servers, so that the system can also achieve work load balance between clusters during the get process.

In the following, the main features of the new generic distributed storage pattern are given. Lazy. This pattern makes the distributed storage system lazier, it does not use the 2 mechanism [2] to reduce the messages’ traffic, especially when there are no tasks. The API servers do not need to maintain the information of all the data servers, and they do not need to select a data server in a balanced way to store the file. They just send the fragment store requests to
the cluster queue and wait for promises until a timeout. Instead of periodic status report, the
data servers will report their status only on demand.

**Space usage is controllable.** It can happen that the computer running a data server would
not allow that all the hard disk space can be used for storage. Therefore, in order to regulate
the space usage, a hard disk space limit must be set up before the data servers run. After a
request for fragment storing is received, the data server checks the free space availability. If it
has enough space for the fragment, it makes a promise to the request sender API server. After
sending the promise, the related space is reserved before the fragment is received from this API
server.

**Cooperation in storage distribution.** The data servers cooperate in a distributed manner
to store a file. They have hard disk space limits, and the file may be stored only if all the free
space can be saved in all of the free capacities offered by the data servers. Thus, the data servers
cooperate in storing the fragments of a big file.

**Data safety.** In this distributed storage pattern, there are several clusters of data servers.
The file is stored in at least two distributed copies in each cluster. When the API server is
reconstructing the file from the fragments, it may be found damaged in one of the clusters.
Therefore, the API server will try reconstructing based on fragments stored at another cluster.
If on the other cluster the file is not damaged, the API gets the reconstructed file and starts
another goroutine to recover the file in the cluster with the damaged file. If the file is damaged
in all the clusters, then the system cannot recover the file.

**Load balance.** In this pattern, a request and promise system is implemented based on
balanced queue consuming mechanism which is tested in the 3 [3]. In this request and promise
system, the API server sends store requests to the data servers. All the data servers in one cluster
share one RabbitMQ queue. The store request sent to this cluster arrives in this queue. The
data servers in this cluster consume the store requests from this queue in a balanced round-robin
way because each data server has a consumer connected to this shared queue [6]. For example,
suppose there are six fragments to be stored in a cluster. If there are two data servers connected
in the cluster, then each data server stores three fragments. If there are three data servers in
the cluster connected, each data server stores only two fragments.

**Concurrent operations.** The get and delete operations on the fragments are executed
currently. Each operation is processed in a separate goroutine. The number of goroutines
that run at the same time is constrained.

References


Based on Multi-level Heartbeat Protocol, 2009 IITA International Conference on Control,

Eötvös Loránd University, 2020.


Real-time web-based visualization of the Radon transform

Levente Lócsi

Department of Numerical Analysis, Eötvös Loránd University
locsi@inf.elte.hu

The Radon transform is the mathematical tool behind nowadays widely used medical imaging techniques for diagnostic purposes such as Computer Tomography (CT). It was introduced by Johann Radon in 1917 [3], more than a hundred years ago [4]. The basic idea of the transform is to calculate the integral of a given real function defined on the plane along lines of the plane, corresponding to the adsorption of X-rays travelling through segments of the human body from different directions. In practical applications the inverse Radon transform shall be used to reconstruct the original function (or medical image) from the measured line integral values. Radon already showed that this is mathematically possible, and it is closely related to the Fourier transform.

If someone wanted to see examples for the Radon transform on their computer screens, to study it in case of some simple images to get familiar with its basic properties, one needed to write at least some short programs in Matlab (with the Image Processing Toolbox), Octave or Mathematica etc. to utilize their built-in commands for the transform and for visualization. (And/or do the calculations using pen and paper.) This is slow and cumbersome for most introductory purposes.

Our project to be presented aims for a web-based solution, where one could easily create a simple image consisting of some basic objects (rotated ellipses and rectangles, such as in the case of the well-known Shepp–Logan phantom [5] or the recently introduced Bognár lung phantom [1]) and modifications are immediately reflected on the transform. This kind of visualization already existed e.g. in case of the two-dimensional Fourier-transform [6], and is in-line with trends also in educational tools emphasising quick feedback and interaction with students [2]. We see that with some trade-off concerning the details of the transform shown, suitable accuracy and (at least near) real-time interaction is achievable, orders of magnitude faster than using earlier solutions.

References

Extracting deep control flow graphs from P4 syntax trees
Dániel Lukács\textsuperscript{a}, Gergely Pongrácz\textsuperscript{b}, Máté Tejfel\textsuperscript{c}
\textsuperscript{a, c}Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
\textsuperscript{b}Ericsson Hungary Ltd., Budapest, Hungary
\textsuperscript{c}3in Research Group, Eötvös Loránd University, Martonvásár, Hungary
\textsuperscript{a}dlukacs@inf.elte.hu
\textsuperscript{b}Gergely.Pongracz@ericsson.com
\textsuperscript{c}matej@inf.elte.hu

Introduction Our overall research aim is to statically derive execution cost and other metrics from program code written in the P4 programming language \cite{4}. For this purpose, we extract a detailed control flow graph (CFG) from the code, and then turn this into a full, formal model of execution, to extract properties – such as execution cost – from the model \cite{2}. P4 has several features beneficial to program analysis, e.g. the small state space and the lack of loops construct, function stacks, and exceptions. While CFG extraction and analysis is well researched area, details are dependent on code representation and therefore application of textbook algorithms (often defined over unstructured code listings) is often non-trivial. Our aim is to present an algorithm for CFG extraction over P4 syntax trees (ST). During the extraction we create direct links from the CFG to the P4 ST; this way we can access all information in the P4 ST during CFG traversal. We call these overlay graphs deep CFGs. We are utilizing a graph query language to take advantage of graph databases (GDB), but also for compactness and to enable formal proofs of algorithm correctness. We plan to publish both the algorithm and the correctness proof in the full paper.

Problem domain We define CFG extraction for P4 STs uploaded to a GDB. We complete the CFG in the same GDB, so we can associate nodes between the ST and the CFG (e.g. we can link statements in the ST to blocks in the CFG) P4 code has a similar structure to code in popular imperative languages, except that P4 has no loop constructs, and unstructured flows (exceptions and jumps). For now, we consider simplified STs built only from blocks, two-way conditionals, and statements. Blocks can nest any number of that three. Two-way conditionals nest exactly two elements, each either a block or a conditional. Statements are leaves. More complex STs (such as those resulting from the official P4 syntax \cite{4}) can be reduced to this representation by introducing auxiliary edges. We consider the resulting CFG correct if (i) it is a rooted directed connected graph, and (ii) we can reconstruct the traversed ST from the CFG. The latter condition ensures that the ST and the resulting CFG are semantically equivalent.

Approach to CFG extraction The basic idea of our CFG extraction algorithm is that we simply traverse the ST in bottom-up direction, and transform each block nodes and conditional nodes separately to a sub-CFG. At the same time we link to the appropriate statement nodes of the ST as well. Going upwards, we create CFGs that encompass the sub-CFGs created beforehand from lower-level nodes. The CFG created for the top-most block of the function is the full CFG.

Having a graph query notation with high-level semantics simplifies both the algorithm and the proof text. Below, we present the top-level definition of our algorithm in Gremlin \cite{3}. Gremlin is the programming language of a graph traversal machine (GTM) with a memory consisting of a graph, a stack, a key-value-store, and other high-level elements.

\[ \text{cfg} = E_{\text{nest}, \text{true}, \text{false}} \circ \text{inV} \circ \text{order} \circ \text{byId}_{\text{desc}} \circ \text{assoc} \circ \text{addV}_\text{block} \circ \text{addP} \circ \text{map}_{\text{handleChildren}} \]

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).
The following steps summarize the semantics of this traversal. 1. Clear the stack, and push all nesting or conditional edges of the ST onto the stack; 2. Replace the edges in the stack with their destination vertex; 3. Order the stack so that vertices lower on the ST are traversed earlier; 4. Assign the vertex in the stack to a named register; 5. Create an entry vertex for the CFG and put it onto the stack; 6. Send an edge from the CFG vertex and tree ST vertex, and replace the stack top with the edge. 7. Apply a custom routine to create a sub-CFG using the statements, nested blocks and conditionals under this ST vertex.

Given that the algorithm is in the form \( f \circ g \circ h \), and Gremlin has a simple state space, we can use the traditional approach to prove program correctness. Let pre-condition \( P \) be that the graph is a proper ST, and post-condition \( Q \) be that the output is a correct CFG. To prove \( \{ P \} f \circ g \circ h \{ Q \} \), we show \( \{ P \} f \{ Q_f \} \land \{ Q_f \} g \{ Q_g \} \land \{ Q_g \} h \{ Q \} \) for some \( Q_f, Q_g \) intermediate conditions. The intermediate conditions regard the current state of the GTM (stack, names, graph, etc.).

**Related work** In our earlier work [2] on P4 cost analysis, we analysed probabilistic cost formulas produced through CFG traversals. For early experiments, we leaned on an abstract CFG representation such as the one produced by the official P4 compiler, P4C. We are now at a point where we need a more refined CFG representation to perform in-depth cost analysis incl. call semantics, basic statements, etc. Unfortunately, P4Cs CFG is not intended as a stand-alone representation, and our aims would have required additional parsing and analysis resulting in complex software architecture. In our solution, we can always access relevant facts in the ST during CFG traversals.

For conciseness, we omit individually referring the standard textbooks on compiler optimization and static analysis concerning CFGs over low-level code. Instead, we mention one work on CFG extraction with focus on soundness. Amighi et. al. [1] solves the more difficult problem of producing CFGs that make exception flows and stacks explicit in Java bytecode CFGs. They prove that the extraction is correct by establishing the existence of a simulation relation between states induced by the bytecode instructions and states induced by the extracted CFG.

**Conclusion** In this abstract, we described our use case, and introduced a simple, graph query-based algorithm for extracting CFGs from P4 STs. One of our goal was to improve upon existing high-level CFG implementations and create detailed CFGs for P4 as an overlay on the P4 ST to provide direct access to facts in the ST. Since GDBs are ideal for storing large attribute graphs, we are representing the CFG extraction as a graph query, and take advantage of this formalism to propose an approach for proving its correctness. In future work, we plan to use the CFG extraction as the major part of a larger P4 model extraction algorithm we will use for cost analysis.

**References**


Randomized dispersion of mobile robots – theory and experiments

Tamás Lukovszki, Péter Vadász
Eötvös Loránd University, Faculty of Informatics
{lukovszki,pevad95}@inf.elte.hu

In swarm robotics, a large number of autonomous mobile robots cooperate in order to achieve a complex goal without any central control. The robots of the swarm are simple, cheap, and they have limited computational abilities. Robot swarms can achieve high scalability and fault tolerance. In this work, we investigate the problem of dispersion of robots in an unknown area. We use the model introduced by Augustine and Moses [1]. In this model the area is represented by an arbitrary connected undirected graph of \( n \) vertices. Initially, \( k \) robots (\( k \leq n \)) are placed arbitrarily at the vertices of the graph. More than one robot can be at the same vertex of the graph at the same time. The robots are homogeneous and they have limited communication and computing abilities. They operate autonomously. Each robot only sees the robots at the same vertex. The robots at the same vertex can communicate with each other in a limited way. The robots can move on the edges of the graph. The task is to achieve a configuration, where each vertex is occupied by at most one robot. Each robot runs the same algorithm performing Look-Compute-Move (LCM) cycles. In the "look" phase they take a snapshot about their environment which means to discover the robots at the same vertex. In the "compute" phase they compute whether to settle down or move to an adjacent vertex. In "move" phase they move there if needed. In solving the dispersion problem our goal is to minimize the memory used by the robots and the number of LCM cycles required for dispersion.

Kshemkalyani and Ali [2] studied the model of Augustine and Moses and presented three deterministic algorithms working with arbitrary connected graphs. Two of the algorithms assume asynchronous model and all of the algorithms are variants of distributed depth first search (BFS). The time complexity of the algorithms is linear in the number of edges \( O(m) \). Kshemkalyani et al. [3] showed a simpler efficient methods for grid graphs. Molla and Moses [5] presented a random walk-based algorithm to solve the dispersion problem, which is simple and robust. Further advantages over deterministic solutions are that robots do not need a unique identifier, they use less memory and fewer bits for communication. The memory requirement of the robots is \( O(\log \Delta) \) bits, where \( \Delta \) is the maximum degree of the graph. The number of communicated bits is also \( O(\log \Delta) \). The bounds on the cover time of a classical random walk can not be applied for bounding the running time of this algorithm in arbitrary graphs.

In this paper we propose an algorithm which makes the cover time bounds of the classical random walk applicable. Therefore, the dispersion is achieved expectedly in \( O(n^3) \) LCM cycles in arbitrary graphs and \( O(n \log n) \) steps on complete and expander graphs. The number of communication bits and memory bits remain \( O(\log \Delta) \) and the the necessary persistent bits (which must be stored between two LCM cycles) is \( O(1) \). After that we investigate the effect of \( \log \Delta \) bits of persistent memory. We show that on certain graphs the runtime of the algorithm of Molla and Moses is \( \Omega(n) \) times higher than the algorithm with \( \log \Delta \) bits memory.

We evaluate our methods by simulations using our simulation framework. The algorithms are simulated on graph topologies, such as circle, grid graph, complete graph, "lollipop" graph, hypercube, Erdős-Rényi random graph and a modified line (exact definition later). For classical random walks, there are well known bounds on the cover times in these topologies (see [4]). These theoretical results ensure a useful base to compare the different solutions. The simulation results show that the running time of the proposed classical random walk based algorithm matches very well to these bounds. Furthermore, they show that the proposed classical random walk based method achieves a faster dispersion than the algorithm by Molla and Moses in some graph topologies.
References


Semantic Consistency behind Ontology Learning and Mapping for Heterogeneous Data Integration

Chuangtao Ma, Bálint Molnár

Department of Information Systems, Faculty of Informatics, Eötvös Loránd University, Hungary
machuangtao@caesar.elte.hu, monlarba@inf.elte.hu

Ontology-based data integration (OBDI) plays a critical role in heterogeneous data integration, due to the excellent semantic interoperability and rigorous mathematical foundation of ontology [1]. However, the traditional methods for constructing ontology are manual, in which a lot of effort and experience from domain experts are required. Accordingly, ontology learning (OL) was proposed to (semi-)automatically construct ontologies, in which entities and relationships are usually extracted based on the computation and inference. The techniques of ontology learning are classified into four categories: association rule mining (ARM), formal concept analysis (FCA), inductive logic programming (ILP), neural networks (NN) and machine learning [2].

However, the semantic collision and errors will inevitably occur while learning and constructing ontologies from multi-source database (semi-)automatically [3]. Consequently, it will induce the redundancy and inconsistency during the process of ontology-based data integration from heterogeneous database.

To address the above issues, the semantic consistency behind ontology learning and mapping are investigated based on the formal representation language [4]. Considering the semantic compatibility of OWL DL (description logic ontologies) and excellent expressivity of description logics (DLs) [5], DLs are employed to formalize the process of ontology learning and mapping from heterogeneous database. Furthermore, W-Graph is utilized to formalize the semantic mapping for providing data-exchange between database and ontologies, in which the semantic consistency behind ontology learning is analyzed by introducing model checking. Moreover, an example is presented to demonstrate the specific process of formalizing and checking the semantic consistency behind ontology learning and mapping for heterogeneous data integration.

Acknowledgments

This work was supported by grants of the European Union co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002), and by no. ED_18-1-2019-0030 (Applicationspecific highly reliable IT solutions) program of the National Research, Development, and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme.

References

Representation of Product Catalog in a Multifaceted Demand Structure by Hypergraphs
Zsighmond Máriás, Bálint Molnár
Information Systems Department, Eötvös Loránd University of Budapest, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
(zmarias, molnarba) @inf.elte.hu

Abstract
The Bill of Materials can be considered as a traditional data structure in Enterprise Information Systems. However, the most recent development in e-commerce creates a two-pronged problem. Based on the Bill of Material, a Product Catalog should be easily navigable by consumers and manufacturers, logistics companies, and salespersons based on rich product data and taxonomy of products. The structure should serve a multifaceted demand structure. From IT’s viewpoint, the Product Catalog should provide data for the consumer interface and data collection management system.

Both sides can be perceived as complex active documents that interchange information so that for this reason, the interaction should maintain consistency, integrity, validity, and security.

The classic Product Catalogs are defined by a category tree, where leaf categories contain the actual product items enriched with descriptive data, thus creating the complex documents mentioned above. The document schema is based on the document schema defined in the different levels of the category structure. In this representation, it is rather difficult to implement features such as products in multiple categories, multiple occurrence of the same category within the taxonomy, create product document schema independent from categories, and product relations across the whole product catalog. These features are required for the sake of the different purposes of the different systems and users working with the BoM / Product Catalog.

Therefore, in this paper, we suggest a hypergraph-based representation which is rather flexible to overcome the regular solution’s limitations and suggest a transformation algorithm to upgrade the regular BoM representation into the hypergraph based one suggested in this paper.

Acknowledgement The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and by no. ED_18-1-2019-0030 (Application-specific highly reliable IT solutions) program of the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme
Quadratures for the numerical solution of fractional differential equations

Gábor Maros

Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University
magabor42@gmail.com

The study of fractional order differential equations has become an important research area in the past twenty years. A large number of real-life observations confirmed the presence of the so-called fractional diffusion. To describe these dynamics, fractional-order differential operators are necessary. One of the most important fractional operator is the fractional Laplacian \((-\Delta)^{\alpha}\). Let \(\Omega \subseteq \mathbb{R}^d\) a bounded domain and let \((\lambda_1; \varphi_1), (\lambda_2; \varphi_2), ... (\lambda_k; \varphi_k), ...\) be the eigenvalue-eigenvector pairs of the Dirichlet-Laplacian operator \(-\Delta_D\) acting on the Sobolev space \(X := H_0^1(\Omega) \cap H^2(\Omega)\). Let \(u = \sum_{k=1}^{\infty} c_k \varphi_k \in X\), where \((c_k)_{k \in \mathbb{N}}\) are the appropriate coefficients. Then the fractional power \((-\Delta_D)^{\alpha}\) for \(0 < \alpha < 1\) can be expressed by

\[ (-\Delta_D)^{\alpha} u = \sum_{k=1}^{\infty} \lambda_k^\alpha c_k \varphi_k. \]

This operator is non-local, which means that in the numerical simulation of fractional problems full matrices arise. This results in longer computation times and reduced efficiency compared to the local case when \(\alpha = 1\). Dealing with the full matrices arising from the discretization of non-local operators is a challenging topic. One can explore their structure, which makes possible to develop an efficient solution of the corresponding linear problems. Such efficient algorithms to solve fractional-order differential equations have been proposed: methods based on the best rational approximation (BURA) [2], quadratures using a resolvent-type characterization of the matrix-exponential [1], matrix-transformation methods [3]. We introduce new numerically efficient quadratures, that are based on the semigroup approach. The method can be applied generally for any discretized operators, e.g., finite elements, finite differences. We also compare the computation times and the approximation errors of the proposed schemes to the already established methods.

References


Group graded Morita equivalences induced by wreath products

Virgilius-Aurelian Minuță
Babes-Bolyai University of Cluj-Napoca, Romania
Faculty of Mathematics and Computer Science
Department of Mathematics
minuta.aurelian@math.ubbcluj.ro

An important strategy used in the verification of the local-global conjectures of the representation theory of finite groups is to reduce the statements to some stronger ones about simple groups. In order to obtain such reduction theorems the language of character triples and of the relations between them has proven useful, as surveyed in the recent results of Britta Späth [3, 4, 5]. One aspect of her research shows that a new character triple can be constructed via a wreath product construction of character triples [5]. This, together with the fact that there is a link between character triples and a special type of group graded Morita equivalences, as presented in [1], provides the motivation for checking whether group graded Morita equivalences can be induced by wreath products.

In this presentation, we will obtain group graded Morita equivalences for tensor products and wreath products, as presented in [2], while also keeping in forefront the context given by character triples.

References

Some properties of the fixed point equation with Meir-Keeler operator

Mădălina Moga
Faculty of Mathematics and Computer Science, Babeș-Bolyai University Cluj-Napoca
madalina.moga@math.ubbcluj.ro

Abstract. In this talk we will recall the notion of Meir-Keeler operator and we will give some examples. Using a characterisation theorem given by Lim we will present several stability properties (data dependence, well-posedness in sense of Tykhonov, Ulam-Hyers stability and Ostrowski stability property) of the fixed point equation $x = f(x)$, where $f : X \rightarrow X$ is a Meir-Keeler operator and $X$ is a complete metric space.

References

Fully Dynamic Strong Connectivity and Reachability in Digraphs

Gregory Morse, Tamás Kozsik
ELTE Eötvös Loránd University, Budapest, Hungary, Faculty of Informatics, 3in Research Group, Martonvásár, Hungary
morse@inf.elte.hu kto@elte.hu

Although computing all-pairs reachability and strongly connected components (SCCs) have well-known algorithms [1, 2], less focus has been given towards their incremental and decremental variants and more specifically towards finding a way to bring them fully online so that both operations can be done without data structures too specific towards edge additions or edge deletions.

In the context of compilers and static analysis tools, the control flow graph (CFG) often needs to make reachability queries or identify SCCs when evaluating, refactoring or optimizing code.

An attempt at moving towards an efficient, deterministic and precise fully online algorithm is thus made based on utilizing ideas of well-known non-incremental algorithms [3, 4]. In this paper, efficient data structures for storing SCCs are used [5] but transitive closure is represented in a manner that requires propagation. For dense graphs and sparse graphs which are expected to contain many SCCs or large SCCs, it can be expected to have good performance. The incremental case will achieve better performance than the decremental case given that Union-Find [5] is used as opposed to Even-Shiloach (ES) trees [6]. Intuitively, adding edges can merge SCCs into a larger one and thus the removal of an edge has to effectively perform a discovery process.

Although this algorithm does not achieve state-of-the-art bounds on the incremental [7] or decremental [8] cases for SCCs, for digraphs, no publication has attempted to address the fully online problem. For all-pairs reachability, many algorithms are known but only in this isolated context. As well the best known algorithms are probabilistic or randomized without decrease of their time complexity and not precise and deterministic as is needed in the given context of CFGs which tend to be sparse and contain a relatively low number of vertices.

The algorithm will be similar to and based on the offline SCC-based reachability algorithm that it is based on maintaining SCCs in $O(n)$ space in $O(n+m)$ time and adds an additional term of $O(n^2)$ for the space and $O(mn+n^2)$ worst case time for reachability propagation where $n$ and $m$ are the number of vertices and edges in the graph respectively. For total update time on insertion and deletion, the presented algorithm has the same storage space and worst case time complexity, yet in practice the cases in which the worst case would occur are rare.

Consider the example of random graphs with $\frac{2n}{3}$ edges inserted, a single offline computation performed, and finally all of the edges being deleted. As the number of vertices (and thereby edges) in the graph increases, the incremental case for a sequence of $\frac{2n}{3}$ edge insertions is only slower by approximately a constant factor $a_1$ from the final single offline computation. Likewise, the decremental case for a sequence of $\frac{2n}{3}$ edge deletions will have an increase in time of about $a_2 n$ where $a_2$ is a constant. This comparison is in fact generous to the offline algorithm as a graph with this many edges will possibly and very probabilistically contain only one large component. Without considering pathological cases, it will perform much faster than repeated offline computations after each edge modification in the graph. Testing on random sparse graphs including some with certain property constraints as well as actual CFGs taken from selected Windows and Linux programs was performed to evaluate the algorithm.

References


Acknowledgement

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).
Experiments in Computational Evolutionary Perception
Mirtill Boglárka Naghi, David Iclanzan
Sapientia Hungarian University of Transylvania
Faculty of Technical and Human Sciences Târgu-Mureș
naghi.mirtill@gmail.com

Introduction
Several researchers in vision science agree that the primary function of perception is to create an accurate picture of the world around us [5, 3]. Another school of thought argues that perceptions constitute a species-specific “user interface” that guides fitness enhancing behavior in a niche [1]. According to American cognitive scientist Donald D. Hoffman, evolution favored the perception of a fitness-based truth, instead of the objective reality [2]. The question whether evolution favors veridical perceptions was also investigated with the help of genetic algorithms (GAs) [2]. Building upon the problem first studied by Mitchell and detailed in [4], the authors evolve with the help of a GA a perception-decision-action loop for an agent in a simple 2D 10x10 grid, surrounded by walls. The GA evolves a creature whose goal is to collect as many resources as possible within 200 moves. They find that besides it evolving effective foraging strategies, it also evolves interface perceptual strategies. In this paper we analyze if the representation strategy used in the previous studies is scalable to a 3D grid and compare it to the performance of different control strategies obtained by neuroevolution.

Neuroevolution of Control Strategies
We hypothesized that neural networks by providing robust and scalable data compression, their neuroevolution qualitatively (not just quantitatively) outperforms direct representation, used in previous experiments [2, 4]. Neuroevolution can capture more data at reduced costs, and by using memory, it can find qualitatively better strategies. To test our aforementioned hypotheses we implemented a simple three layer neural network topology (19-31-18 neurons), that uses ReLU as the activation function. We used a GA to evolve the weights of the network to map the agents perceived environment (input layer) to the best action in the output layer, signaled by the highest activation obtained here.

Results and Discussion
In our first attempt, we reproduced the experiments [2] done by Dr. Hoffman. We tested the mutational rate with different values, examined where the achieved results peaked for larger populations, and experimented with three different fitness functions: one described in the article by Hoffman (0, 1, 3, 6, 9, 10, 9, 6, 3, 1, 0), one whose values are twice the previous one, and one that had a smaller distribution. Subsequently, we applied the function described in the article for further experiments. Our results, summarized in fig. 2 were consistent with those obtained earlier on by Hoffman. We then concluded that the direct representation which was used in the previous experiments cannot function effectively for larger dimensions, as the cost of representation increases exponentially with each new “direction”. In 3D considering two adjacent squares it requires a gene length of $3^{13} = 1 594 323$. The length of the gene sequence which encodes our neural network is 1 128. Even if we encode more information than in the direct representation, the cost of our network is still a fraction of that (more precisely 0.075%). As shown in fig. 3, the neuroevolution approach qualitatively outperforms the direct representation:

6https://melaniemitchell.me/ExplorationsContent/RobbyTheRobot/
the latter performing quite poorly. In the next attempt, we introduced the memory neurons. Evolved control strategies use the signs stored in the memory cells to navigate through the map. The use of memory did not lead to better performances. Yet, if the agent has to pay a cost penalty proportional to the number of directions it sees, the memory-using version is not affected at all, as it does not rely on its vision, but instead on its memory (neuroevolution regularly blinds them). A 3D agent will perform similarly to its 2D relative when penalized.

**Conclusion**

Experiments conducted using genetic algorithms by Donald Hoffman and Melanie Mitchell were reproduced with similar results and some aspects were analyzed more in-depth. Simulation results had clearly shown that direct representation used in previous studies is not scalable to a 3D grid. Further experiments were performed using a neural network, which proved to be a robust and scalable data compressor, obtaining effective control strategies. The neural network developed through neuroevolution surpasses direct representation in both cost and performance in higher dimensions. The agents did indeed develop an interface strategy in both 2D and 3D environments.

**References**


A representation of quaternionic Blaschke group
Zsuzsanna Nagy-Csiha, Margit Pap
Eötvös Loránd University, University of Pécs
ncszsu@gamma.ttk.pte.hu, papm@gamma.ttk.pte.hu

Let us denote by $D$ the unit disc, $D := \{z \in \mathbb{C} : |z| < 1\}$ and by $T$ the unit circle. The Blaschke-functions are defined by

$$B_{(b,\epsilon)}(z) := \epsilon \frac{z - b}{1 - \overline{b}z} \quad (b \in D, z \in \mathbb{C}, z \neq 1/\overline{b}, \epsilon \in T).$$

Let us denote the set of the parameters by $B := D \times T = \{(a = (b,\epsilon) : b \in D, \epsilon \in T)\}$. In the set of the parameters $B$ let us define the operation induced by the function composition in the following way: $B_{a_1} \circ B_{a_2} = B_{a_1 \circ a_2}$. The set of the parameters $B$ with the induced operation is called the complex Blaschke group.

The representation of the Blaschke group on the complex Hardy space of the unit circle $H^2(T)$ is defined by

$$(U_{a^{-1}} f)(z) := \frac{\sqrt{e^{i\theta}(1 - |b|^2)}}{1 - \overline{b}z} f\left(\frac{e^{i\theta}(z - b)}{1 - \overline{b}z}\right) \quad (z = e^{it} \in T). \quad (19)$$

In [2.] and [3.] Pap M. and Schipp F. introduced and studied the properties of the representation given by (19) and the properties of the induced voice transform

$$(V_g f)(a^{-1}) := \langle f, U_{a^{-1}} g \rangle \quad (f, g \in H^2(T)).$$

Let $\mathbb{H}$ be the set of quaternions, and $S = \{q \in \mathbb{H} : q^2 = -1\}$. A function $f : \Omega \rightarrow \mathbb{H}$ is said to be slice regular, if for all $I \in S$ its restriction $f_I$ to $\Omega_I$ is holomorphic, i.e. it has continuous partial derivatives and satisfies

$${\overline{\partial}} f(x + yI) := \frac{1}{2} \left( \frac{\partial}{\partial x} + I \frac{\partial}{\partial y} \right) f_I(x + yI) = 0.$$

The quaternionic Blaschke group was introduced and studied in [1.]. This extension depends on three parameters. We consider a subgroup of this group, the so called slice regular Blaschke group $(A_H, \circ)$.

We introduce the representation of the slice regular Blaschke group $(A_H, \circ)$ on the slice determined by $I$ to the slice regular Hardy space in terms of the slice regular composition and slice regular product on this space, and we study the properties of the introduced representation. As in the complex cases, the representation for the quaternionic Blaschke group also turn out to be unitary.

The aim of our research is to extend the results connected to the voice transform of the Blaschke group on the complex Hardy space obtained in [2.] and [3.].

References
Correctness of a High-level RCU Implementation
Gergely Nagy, Zoltán Porkoláb
Eötvös Loránd University
gergeley.a.nagy@gmail.com, gsd@elte.hu

As the hardware and software development industries are reaching the limits of single-threaded CPU performance, the need for continued improvements had us search for other solutions, namely highly concurrent generic-purpose programming. Since such programming disciplines are generally considered hard, it has been in the focus of researchers and industry parties. Numerous advancements have been made in recent years, starting from new computational models to a wide adoption of generic-purpose GPU programming. Abstracting away complexity has always been a major intention in these new models, trying to keep complexities hidden from programmers. We also believe it has to be a major consideration for every new approach.

Concurrent programming with classical mutex/lock techniques does not scale well when reads significantly outnumber writes. A locking scheme named read-copy-update (RCU) allows reads to continue while a write operation is in progress by providing a classical space-time trade-off. Several low-level implementations have been created for systems programming languages such as C and C++, but none at higher-levels of abstractions. We have previously implemented the RCU pattern in Scala and have analyzed its performance characteristics by implementing a concurrent hash map. We have modified the non-thread safe hash map container in the Java standard library by simply guarding the internal data structures by the RCU locking scheme. We have ran an extensive benchmark suite to test how this version compares to the standard concurrent hash map with a varying ratio of readers and writers. We have observed that the RCU hash map has comparable performance characteristics when reads are more frequent than writes. We have also tested how the performance of these data structures scales by increasing the number of concurrent accesses.

Correctness of any library is a key consideration. While we can ensure correctness to some degree by writing comprehensive unit tests, formal proves add a significant benefit. As multi-threaded programming is considered a complex subject, so is any formal framework that can be used to prove correctness of concurrent programs. Several methodologies have been developed that help with proving correctness, including the extended temporal logic, the Owicki-Gries model and ones using the happens-before relation. The memory model that is available on the JVM has used the happens-before relation to establish the guarantees of its low-level synchronization constructs, thus it was the natural choice for our purposes as well.

The happens-before relation guarantees for two actions that the results of the action are visible to the other action even if they are run in different threads. The Java Language Specification (JSL) defines the happens-before rules of several low-level actions, including statements in a single thread, locking a monitor, reading and writing volatile variables and starting and joining threads. It is easy to see that the relation is transitive, meaning that if actions A and B as well as B and C are in relation, A and C are also necessarily so.

In this paper we have used the happens-before relation to prove the correctness of our previous RCU implementation. We have built on the JSL rules and established the happens-before rules for a read-write lock. This is necessary as our main RCU implementation uses a read-write lock. Then we defined for rules that would guarantee the correctness of our lock. Furthermore, one of the rules are used to show that the key feature of RCU – write operations can be done while reads are not being interrupted – is also guaranteed. We also show why our implementation is free of deadlocks. These two aspects guarantee that the high-level RCU will not only not corrupt the data, but it also guarantees the progression of operations.
Color image analysis and recognition using orthogonal quaternion Zernike moments

Zsolt Németh, Gergely Nagy
Department of Numerical Analysis, Eötvös Loránd University
birka0@inf.elte.hu, nagygeri197@gmail.com

Image moments and moment invariants are widely used in applications for pattern matching [6], image recognition [1], or to extract useful features from images [9] in general. Most of the classical moments were originally defined for single-channel, grayscale images in the literature, however extending these techniques to multichannel ones is an important and generally unresolved problem. Conventionally, for color images either RGB decomposition or grayscale conversion was used in order to utilize the methods defined for grayscale images. This may lead to loss of information, for example in the case of grayscale conversion (where the average over the channels is taken) some color information can be lost.

More recently, the algebra of quaternions has been used to extend the single-channel methods to color images. For example, quaternion Fourier-Mellin moments have been introduced as an extension of the conventional Fourier-Mellin moments [5], as well as the quaternion Zernike moments as an extension of the conventional Zernike moments [2].

The Zernike functions are a system of orthogonal functions defined over the unit disk. Using these functions as a basis for series expansions proved to be useful because of certain inherent invariance properties. Zernike moments, and by extension quaternion Zernike moments are defined by these functions.

Considering a digital image as a discrete sampling of an image function defined over a continuous domain, the need arises to discretize the computation of these moments. One important property of the discretization of these methods defined by orthogonal functions is to preserve the orthogonality over the discrete system, so as to avoid redundancy and achieve high robustness with respect to noise.

The conventional method for discretizing quaternion Zernike moments (used by Chen et al. [3]) consists of uniformly distributed points over the unit disk. This method does not achieve discrete orthogonality thus decreasing the robustness of the moments.

In this talk, we propose a novel method for the discretization of quaternion Zernike moments over the unit disk: a points system is defined on the unit disk, over which the Zernike functions extended to quaternions are discrete orthogonal. This improves the robustness of computing quaternion Zernike moments by decreasing the error introduced by discretization. Our construction is based on the results of Pap and Schipp [8] for the real valued case.

The new method is compared to the one used by Chen et al. [3]. For the tests, image sets from the Columbia Object Image Library [7] and the Amsterdam Library of Object Images [4] were used. The image reconstruction capabilities of both methods are also compared, and we find that the proposed method decreases the error of reconstruction significantly.

The recognition capabilities for rotated, scaled and translated (RST transformed) images with varying levels of either Gaussian or salt-and-pepper noise are also studied. We find that with respect to Gaussian noise the new method achieves significantly better rates of recognition, even for images with high noise values. For salt-and-pepper noise no significant difference can be found between the capabilities of the methods. Additionally, we also show that by decreasing the number of points used for discretization, the new method is able to achieve similar results as the original method with high number of points, but the computational need to obtain these results is much lower using the new method.

References


A new bootstrap resampling scheme for INAR processes with trend

László Németh, András Zempléni

Department of Probability Theory and Statistics, Eötvös Loránd University, Budapest
lnemeth@caesar.elte.hu, zempleni@caesar.elte.hu

Stationary integer valued autoregressive (INAR) processes are suitable to model numerous real life time series. However, these processes are usually not stationary, but contain seasonality or trend. In this talk we investigate the Poisson distribution-based INAR processes containing trend in the innovation parameter. Our main aim is to introduce an appropriate bootstrap resampling method, suitable for investigating e.g. the distribution of the parameter estimates in this case.

The definition of an INAR process of order \( p \) is the following:

\[
X_s = \alpha_1 \circ X_{s-1} + \alpha_2 \circ X_{s-2} + \cdots + \alpha_p \circ X_{s-p} + \varepsilon_s,
\]

where \( \alpha \circ X_h \) is a realization of \( \text{Bin}(X_h, \alpha) \) and \( 0 < \alpha_1, \ldots, \alpha_p < 1 \). The \( \varepsilon_s \) innovation process is integer valued, independent and identically distributed. A typical choice is the Poisson distribution with parameter \( \lambda \) that we also use in our analysis. In this talk we investigate the first order, INAR(1) processes.

In order to define an INAR(1) process with trend, we fixed a \( \lambda_0 \), as the parameter of the innovations for \( X_1 \), and added a linear trend \( \tau \). The structure of the process is then:

\[
X_s = \alpha \circ X_{s-1} + \varepsilon_s,
\]

where \( \varepsilon_s \) is Poisson(\( \lambda_0 + \tau \cdot (s-1) \)) distributed.

Our approach is based on the following steps, to be introduced in more detail in the first part of the talk.

1. Estimate the trend in expectation by a linear regression using an appropriate method, denote it by \( t \). By subtracting the estimated trend, one can construct a detrended sample \( X^d = X^d_1, X^d_2, \ldots, X^d_n \) where \( X^d_i = X_i - t \cdot (i-1) \). Since \( X^d \) can have non-integer values, the detrending procedure can not be modelled by an INAR(1) type series. However, as the detrending procedure does not distort the residual structure of an AR process (by [Yue2003]) the autocorrelation remains similar as in the original series, therefore we can estimate \( \alpha \) by \( \hat{\alpha} \) using the Yule-Walker equations. This allows for estimating \( \tau \) as well.

2. We apply a special detrending procedure, from which after a "retrending" procedure we receive a bootstrap time series with similar properties as the original one. We call this procedure the transformation bootstrap.

In the second part of the talk we compare the properties of the proposed transformation bootstrap to other well-known bootstrap procedures. Our simulation study shows that despite the differences in the distributions, our method is able to estimate suitable confidence intervals for the trend coefficient \( (\tau) \) and the autocorrelation parameter \( (\alpha) \). For the \( \lambda_0 \) parameter the parametric bootstrap is more efficient.

Finally, we test our method on the dataset of the most significant volcanic eruptions of the past 100 years.

References

Comparing epidemiological models with the help of visualization dashboards

Boróka Olteán-Péter
Sapientia Hungarian University of Transylvania
boroka.oltean@ms.sapientia.ro

In 2020, due to the COVID-19 pandemic, various epidemiological models appeared in major studies [1., 2., 3., 4.], which differ in terms of complexity, type, etc. In accordance with the hypothesis, a complex model, which takes into consideration more parameters, is more accurate and gives more reliable results than a simpler one.

In this paper we study three different epidemiological models: a SIR, a SEIR and a SEIR-type model. Our aim is to set up differential equation models, which rely on similar parameters, however, the systems of equation and number of parameters deviate from each other. A visualization dashboard\(^7\) is implemented through this study, and thus, we are able not only to study the models but also to make users understand the differences between the complexity of epidemiological models, and ultimately, to share a more specific overview about these that are defined by differential equations [5.].

In order to validate our results, we make a comparison between the three models and the empirical data from Eastern Italy and Wuhan, based on the infectious cases of COVID-19. To validate our results, we calculate the values of the parameters using the Least Square optimization algorithm.

References


---

\(^7\)https://seir-visualisation.vercel.app/
On a special class of fractional type set-valued functions
Alexandru Orzan, Nicolae Popovici
Department of Mathematics, Babeş-Bolyai University, Cluj-Napoca
orzanalexandru@yahoo.com, popovici@math.ubbcluj.ro

Vector-valued ratios of affine functions have been introduced and studied by Uriel G. Rothblum in 1985, in the framework of finite-dimensional spaces. These functions preserve the convexity of sets under direct and inverse images. Recently, Alexandru Orzan proposed a similar concept for set-valued functions and proved that they also preserve the convexity of sets. It is our aim to characterize this class of set-valued functions and to further investigate the preservation of convexity under lower and upper inverse images.

References
Existence and uniqueness theorems for ordinary
differential equations with mixed initial and
boundary value conditions

Zsolt Páles
University of Debrecen
pales@science.unideb.hu

The aim is to investigate the solvability of the ordinary differential equation

\[ y'(x) = f(x, y(x)) \quad (x \in I) \]

for Banach space-valued functions \( y : I \to Y \) satisfying the condition

\[ A_1 y(\xi_1) + \cdots + A_n y(\xi_n) = \eta, \]

where \( Y \) is a Banach space, \( I \) is a real interval, \( f : I \times Y \to Y \) is a continuous function, \( A_1, \ldots, A_n : Y \to Y \) are bounded linear operators, \( \xi_1, \ldots, \xi_n \in I \) and \( \eta \in Y \). Existence and uniqueness theorems are obtained for this general setting.
Lightweight, length invariant models and dimensionality reduction in respiratory disease detection
Tamás Pál, Bálint Molnár, Ádám Tarcsi
Information Systems Department, Eötvös Loránd University of Budapest, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
(evwolcheim, molnarba, ade) @inf.elte.hu

The detection of respiratory diseases has been an important field of study of respiratory illnesses that are responsible for millions of deaths yearly. Machine learning offers a plethora of methods to preprocess, analyze, and classify such recordings. Approaches that have reduced computational demand are preferred to achieve shorter processing time.

Two deep learning models are proposed that are length-invariant and have simpler neural network topologies. With length invariance, the processing time is shortened, as splitting the recordings into equal-sized segments is not necessary anymore. Moreover, extracted spectrograms of the records can be reduced in dimensionality by calculating aggregated values along the time axis and using efficient methods like PCA or tSNE \[1\]. Mel Frequency Cepstral Coefficient (MFCC) spectrograms were extracted.

The first deep model is a lightweight dense network that receives as input feature vectors from aggregated spectrograms. Inputs of different dimensionality are compared. The second model is inspired by the 1D MaxPooling architecture by Phan \[3\] that introduce through the use of global max-pooling layers length invariability into the model. An extra hidden layer and other minor modifications are added that increased the classification performance in the case of this data-set. 2D spectrograms are used as input for this model.

The respiratory sound database contains 920 annotated breathing recordings so that this database includes the symptoms of 7 classes of diseases or records that constitute as healthy. The data-set was created by a Portuguese and Greek research group \[2\]. The data were collected from 126 patients so that these samples extend over through all age groups, namely children, adults, elderly. The data-set is also heavily imbalanced. The proposed neural networks are systemicall investigated on the before-mentioned data-set (full - 8 class version) and analysed according to the metrics of the discipline. The results are compared with other works on this data-set.

Acknowledgement
The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and by no. ED-18-1-2019-0030 (Application-specific highly reliable IT solutions) program of the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme

References
On a Dirichlet problem for the Darcy-Forchheimer-Brinkman system with application to lid-driven porous cavity flow with internal square block

Ioan Papuc
Faculty of Mathematics and Computer Science, Babes-Bolyai University
ioan.papuc@math.ubbcluj.ro

Using the potential theory technique we prove the existence and uniqueness of a weak solution of a Dirichlet boundary value problem for the Darcy-Forchheimer-Brinkman system in a bounded Lipschitz domain in $\mathbb{R}^n$ $(n = 2, 3)$, when the boundary data belongs to a $L^2$-based Sobolev space. A numerical investigation of the flow of a viscous fluid through a two dimensional lid-driven porous cavity with a solid square block is also performed. The effect of the Darcy and Reynolds numbers as well as the dimension and position of the internal obstacle on the flow behaviour is analysed.
Univalence properties of an integral operator

Oana Părvă, Daniel Breaz

Babeș Bolyai University
Department: Complex Analysis - Mathematics
oanamaria.trif@yahoo.com, breazdaniel@yahoo.co

Most studies refer to analytical functions defined on the inside of the unit disk, so we decided to study an integral operator that contains a univalent function and is part of the class of functions defined on the outside of the unit disk, the less studied part of analytical functions.

We started from the integral operator introduced by Daniel Breaz and Narayanasamy Seenivasagan in the paper "The univalent conditions for an integral operator on the classes $S(p)$ and $T_2$":

$$F_{\alpha_1, \alpha_2, \ldots, \alpha_n, \beta}(z) = \{\beta \int_0^z t^{\beta-1} \prod_{i=1}^n [f_i(t)]^{\frac{1}{\alpha_i}} dt \}^\frac{1}{\beta} \in S,$$

and we take into account that $f_i(t)$ is in the class of univalent functions on the unit disk.

From the paper of G. Kohr and P. Mocanu "Capitole speciale de analiză complexă" we find out that between the $S$ class (univalent functions on the unit disk) and the $\sum$ class (univalent functions on the exterior of the unit disk) there are some links, such as:

(i) Let $f \in S$ and $g(\varsigma) = 1/f(1/\varsigma)$, $\varsigma \in \Delta$. Then $g \in \sum$ and $g(\varsigma) \neq 0$, $\varsigma \in \Delta$.

(ii) If $g \in \sum$ and $g(\varsigma) \neq 0$, $\varsigma \in \Delta$, then $f \in S$ where $f(z) = 1/g(1/z)$, $z \in U$.

So let be the function $g_i(t) = \frac{1}{f_i(t)} \in O_1$, with $g_i(t) \neq 0$; $t \in O_1$, where $O_1$ and $O$ are subclasses of $S$.

After we apply the following transformations:

$$t \rightarrow \frac{1}{t} f'(t)$$

and we get

$$dt \rightarrow -\frac{1}{t^2} dt$$

We form the integral operator:

$$G_{\alpha_1, \alpha_2, \ldots, \alpha_n, \beta}(z) = \left[ -\beta \int_0^z t^{-1-\beta} \prod_{i=1}^n \left( \frac{t}{g_i(t)} \right)^{\frac{1}{\alpha_i}} dt \right]^{\frac{1}{\beta}}$$

In this paper, we determine conditions on $\beta, \alpha_i$ and $g_i$ such that the integral operator $G_{\alpha_1, \alpha_2, \ldots, \alpha_n, \beta}$ is univalent on the exterior of the unit disk for two subclasses of analytical functions.

References


Sleep states detection using Halfwave and Franklin transformation
Y. Paul, S. Fridli
Department of Numerical Analysis, Faculty of Informatics
Eötvös L. University, Budapest, Hungary
yash@inf.elte.hu, fridli@inf.elte.hu

Introduction Sleep is a significant part of person’s life and individuals used to sleep one-third of their entire life. Our research is motivated by the fact that there are large number of disorders like insomnia, breathing disorders, wake-sleep disorder sleep movement disorder found in human beings. Every sleep state has different group of neurological and physiological features and correct identification of these features along with their states are important for diagnosis and the better treatment for such sleep disorders [1]. Sleep classification process is not a standardized one, i.e. different experts have different criteria to mark a specific period of sleep. Usually sleep scientists make classifications by using visual method to predict or decide in which state the patient is for a specific time [2].

Motivation The aim of our paper is to develop an effective algorithm by processing and combining various human biological signals. To this order we extend our method [3] originally developed and successfully applied for epilepsy seizure detection to the case of sleep stage classification in EEG (electroencephalograph) signals. The main challenge to automatic sleep phase detection is heterogeneity. This means that people around the world have different cranial structures which effect the patterns in the signal. For example 10 percent individuals don’t produce alpha rhythm during stage W (wake) and 10 percent create just a constrained or limited alpha beat. This justifies the combination of other signals with EEG in order to improve the results.

Proposed method Our method is a so called hybrid one, i.e. we utilize both time and frequency domains to extract components for the feature vector. The main idea is to use the piecewise linear function model for data reduction. Piecewise linear functions are of low complexity that still preserve the characteristics of the signal relevant in context of stages of the sleep. To this order we took the so called half wave method and modified it according to our need in the time domain. The so called frequency domain is generated by Franklin transformation. The Franklin system an orthogonal system which is strongly related with the well-known Haar system, and its elements are piecewise linear functions. Then the feature vector elements are generated from the piecewise linear models in both domains. We applied this technique for synchronized EEG, blood pressure, and respiratory signals.

Database The database that we used is the MIT-BIH Polysomnographic Database (Physionet,https://www.physionet.org/physiobank/database/slpdb/). It contains 16 annotated records with a total length of more than 80 hours. We note that we used only one EEG channel in our method. This way we have lower computational complexity than the multichannel methods.

Classification After having the feature vector constructed we applied the K Nearest Neighbor algorithm (KNN) for classification. We found that it performs better in this task than other often used classification algorithms like Support Vector Machine (SVM), Artificial Neural Network (ANN) etc. This agrees with the conclusions of other authors in this topic. We note that the sets of sleep stages in the databases are not balanced. This means that the numbers of tuples of different classes are significantly different. Therefore, before applying the classifier, the class imbalance problem needs to be addressed otherwise results would be biased. In the proposed method we applied an advanced form of Synthetic Minority Over-Sampling Technique (SMOTE) [4] to address the issue of class imbalance. It is called Adaptive Synthetic Sampling Approach for Imbalanced Learning [5].
**Results** Tests were performed on 15 records (out of 18 records) since three records were not accessed due to technical problems. The total duration of selected 15 long term EEG, Blood and Nasal signals with six different sleep classes is more than 60 hours. First we have optimized our method then we have carried out comparison tests. It turned out that the proposed method shows better performance in terms of an average accuracy than state of the art methods. Our algorithm achieved an average sensitivity, specificity, accuracy and false alarm rate of 98.35% and 97.32%, 96.96%, 0.029 respectively for two randomly selected classes, 96.62% and 97.10%, 93.94%, 0.030 for 4 random selected classes and 96.13%, 98.33%, 93.84%, 0.016 for all six classes.

**References**


EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies - The Project is supported by the Hungarian Government and co-financed by the European Social Fund.
Abstract

This paper analyses the current status in Romania’s digitalization process (based on DESI report) and why Romania faces challenges in removing the barriers that stand between the current status and having good public digital services. A digitalization of public administration has many benefits for the citizens, and for the state via its public administration. The digitalization of the public services represents a method to provide to the citizens better and more efficient services, it represents a method to reduce the bureaucracy and to help the citizen to improve the relation with the public institutions. Once the information is available for citizens and they can resolve the issues and they can communicate with the public institutions via digital applications, the public pressure of providing an answer in a timely manner is drastically reduced, the general satisfaction is increased and the efficiency could reach new standards.

On the other hand, the benefits for public administration are huge. Digitalization implies reducing personnel cost, increasing transparency, reducing corruption, better planning and more efficient information retrieval. Having interconnected systems enables finding information with a high speed and most important having correct and accurate information. Understanding the importance of having connected systems, at European level, there was developed a program called ISA² Programme in order to increase interoperability between the systems in the EU countries. The ISA² Programme supports 54 projects focused on developing digital solutions in interoperability area [20]. However, the main digital platforms used in public administration in Romania are not connected. The paper analyses the need of a digital strategy so that all the digital systems are interconnected; interoperability is a major issue in Romania, as the digital systems do not expose interfaces in order to communicate and to exchange data. Moreover, the application systems in Romania use different data patterns in order to uniquely identify the objects (users, persons, companies), and there’s an urgent need for standardization.

The benefits of using and implementing digital applications in public administration are huge, but the statement is correct only with the assumption that digital services are implemented by a competent team, at the correct price. Even if the benefit of going digital are widely acknowledged, there are many barriers to achieving good digital public services in Romania. Due to various impediments, the current digital applications implemented in Romania face a lot of usability problems. This paper analyses the deficiencies that appeared in the main corruption cases in Romania related to implementing digital public services from two points of view: legal side and implementation/support side. On the legal side, the corruption that exists in public administration at all levels is acknowledged by the European Commission and GRECO report, and it’s a major impediment in having a real progress in the digitization area. This paper analyses other deficiencies that appear in the public service applications and provides solutions. From contract stipulations, to overpricing, conflict of interests on the legal side to the structural and architectural problems of the applications on the other side. The applications use old technologies, they are hard to scale, error prone, insufficient tested and they have a lousy performance.

In order to find the best solutions and to provide a methodology two key approaches are analyzed. One approach is to analyze the main deficiencies of the digital systems in public
administration in Romania, as each country has its specificities and only understanding the problems and analyzing the issues, the best solutions can be found. The other approach is to copy the solutions that were efficient in other countries. There’s no need to reinvent the wheel, and the solutions that worked in other countries could work in Romania too. The paper analyses the Romania’s specificities and also the methods and the approach of the countries that managed to digitalize efficiently. A comparison is performed with Estonia’s and Sweden’s process of digitization. However, no matter how well the information is structured, no matter how well the problems and the solutions are detailed, the decisions are taken at the administration level. Only one application cannot make a lot of change and Romania needs a strategy in order to become more digitalize, needs to prioritize the applications, needs to grow them on a modular base and needs to launch transparent auctions for contracting firms to implement and support them. The society can only put pressure by writing articles and interviews, trying to inform about the necessity to digitalize, the need for correct and efficient, corruption free digitalization process.

**What is digitalization**

There are many definitions for digitalization: digitalization is referring to the process of converting information into a corresponding digital format. Digitalization is the process of storing and processing the data, sharing and accessing it around different applications. Digitization is referring to transforming analogue information and encoding it into bytes, arrays of zeros and ones, a format that allows information to be stored, processed, and transmitted by computers.[8] Digitalization is about the integration of digital technologies into everyday life; it’s about using digital technologies to improve and automate business processes and flows; it’s about providing services into a more efficient way.

Digitalization of the public administration services it’s about providing the same services or even better ones, faster and more efficiently, with a lower cost. Moreover, the digitalization is more than digitalizing information, processes and roles. It’s about the customer; respectively the digitalization of the public administration (eGoverment), it’s about the citizens. The citizens in Romania lose a lot of time interacting with the administration institutions due to the bureaucracy. Their interaction with the public administration entities can become easier and more efficient using digital services; filling an online form should take less time than going to the institution to fill in the same form, obtaining information or making an appointment can take only a few minutes.

However, the state and the public administration benefit a lot by digitalizing the services they provide. Digitalization increases transparency, correctness, and as a result it reduces corruption. Modern, digitalized institutions are less susceptible to the frauded, are more efficient and provide better services. A key component in the digitalization process is related to the system’s interoperability as all the systems should be able to exchange data. Information was delivered in paper format, across departments and different institutions. Due to these impediments, it was hard to provide reliable information and statistics to the managers, it was hard to plan the budget or to track the information changes.[6,28] Interconnectivity and interoperability is a key feature of going digital.

The European Commission realized the potential and the benefit of digitalizing the public services in EU, so it started implementing programs such as eGovernment. “Effective eGovernment can provide a wide variety of benefits including more efficiency and savings for governments and businesses, increased transparency, and greater participation of citizens in political life.

As part of its strategy, the European Commission is taking concrete actions for the development of Cross-border Digital Public Services. These include, but are not limited to, the creation of European interoperable platforms such as a common framework for citizens’ electronic identity management (eID), and the fostering of innovation through the Competitiveness and
Innovation Programme (funding Large Scale Pilots and eParticipation projects)”. European Commission pays attention to the systems interoperability, the ISA² Programme supports 54 projects focused on developing digital solutions in interoperability area. The projects are organized in packages which are defined by the annual work programme.[10,17]

**Current status in Romania**

A good barometer to check the Romania’s status is the European index called The Digital Economy and Society Index (DESI) which is a composite index that summarises relevant indicators on the European countries measuring digital performance and tracking their evolution in digital competitiveness. According the DESI, Romania ranks 27th place in 2017 from 28 states, and 26th out of 28 in 2020.[4] The following image presents DESI index from 2020. Romania scores medium on connectivity and good on human capital, but performs badly in all the other indicators. It’s also important to notice that even if the Romanians are using the internet, 86% of them use it mainly to socialize via Facebook, Instagram or Twitter, as the European average is 65% [5 – DESI 2018].

In public sector, there are a lot of companies that offer digital services, the IT&C sector in Romania is a key component as it produces more than 5% of PIB and the percentage is growing. According to a study performed by Romanian National Bank: “Studiu al evoluțiilor sectorului IT&C în România”, the IT&C sector in PIB represented 5.1% in 2016, 5.5 in 2016 [21], to get to 7% in the first semester of 2020[9]. Statistically, the IT&C sector performs overall very well in Romania, but only 10% of Romania’s population have digital competencies [4]. Regarding this aspect, Romania is the last one compared to UE’s countries; there were initiatives such as National Strategy for Competitivity 2014-2020 that proposed to increase the number from “5% in 2013 to 35% in 2020”, but the project did not reach its objectives.

Romania is dealing with an oversized administration, the agriculture’s minister mentioned in an interview that in the agriculture minister there are 14600 people working because of the high level of bureaucracy. [7] Without having any desire to innovate and to go digital, most of the managers in public administration institutions still prefer the old hard copy method. Piles of sheets of paper are stored everywhere in these institutions, as citizens are waiting for digital
services. In digital public services, Romania ranks 28th place from 28 states and it’s almost at half (scores 48) comparing with the average UE value (scores 72). [4]

Current situation in Romania is also due to a number of corruption cases related to the digitalization process

**Going digital - Major corruption cases**

According to a study made by Transparency International, more than €38.6 billion is lost each year in Romania due to corruption. This value reflects the high corruption in the public administration, 68% of Romanian citizens believe that their lives are daily affected by it.[2]

"Additionally, 25% reported having been asked or expected to pay a bribe for a public service they needed at a given time, the second-highest among the EU. Moreover, Romania scored 47 points out of 100 and ranked, after Hungary, the most corrupt EU member state, in the Corruption Perceptions Index released by Transparency International in 2018."

Even if Romania is monitored, it does not fight against corruption and this fact is known in the European Council and is subject of GRECO report.

“Romania has made "very little progress" to prevent corruption in its government, the Council of Europe’s anti-corruption body said in two critical reports released on Tuesday. The reports evaluated Romania’s previous responses to the Group of States against Corruption or GRECO’s recommendations and concluded that Romania had only fully complied with just over a quarter of the recommendations issued in two previous reports".[11]

Romania holds the lines on European newspaper due to its huge corruption cases – in the most famous one, no more than seven ex-ministers were involved. In 2014, the DNA announced that it has started prosecuting nine former ministers in a case known as Microsoft case for overpriced Microsoft licenses for schools. The damage was estimated at almost 67 million USD.[16]

Other huge corruption case was related to the digitalization of the CNAS (National Health Insurance Company) that signed a contract for a digital health system that should have been delivered in two years and should cost 119 million EUR. The project took 6 years to be implemented and costed over 300 million EUR [3]. An analysis of this contract reveals deficiencies in the contract, as the provider did not have any penalties for delaying, the penalties were defined only for CNAS. In case of early contract termination, CNAS should support by itself all the costs. Despite contractual problems, there are a lot of functional problems – as the system was not designed to support a large number of users and it does not performs as expected.

**Deficiencies of the eGoverment applications implemented in Romania**

There are many deficiencies in the public administration’s digital services. The following part of the paper analyses the main problems that appeared in the applications. Using old technologies. Many of the application use old technologies that cannot scale properly and that are already deprecated by the time the people start using them.

*Low availability.* The users the applications encounter many outages, loss of connectivity and high latency. When a page takes more than a few seconds to load, more than a minute to submit a form, more than 10 minutes to complete a flow, the whole digital application is delaying the public service.
Vendor Lock-in. Vendor lock-in is not a new concept, it represents the incapacity of changing the provider for a specific set of products/services. The main reasons for which this situation occurs are the price, the contract (enforced conditions), the provider’s knowhow and the used technology.

Interoperability. The systems in Romania have other major draw-back: interoperability. Different software from different institutions do not communicate with each other, do not exchange or share information. Moreover, different system use different methods to identify the data: for example, a citizen is identified by national ID (CNP) in some systems, ID serial number in other systems, but in the law system (www.portal.just.ro) the search uses as key the name of the file number. Multiple search criteria is not implemented, so searching for a person named “Pop” from Bucharest, implies searching all the files where appear a person names “Pop” and manually trying to identify the person based on other criteria: address and /or CNP. Ideally, a digital system should provide methods to uniquely identify any data or any data related to a person/company. In order to protect the data, there should exist different levels of rights, each system should be analysed to check what data can expose and to whom.

Another critical gap in Romania’s digital infrastructure is the lack of interconnectivity between various solutions. Every institution or department is asking the same set of personal documents from the citizen instead of getting the information automatically by interacting to the appropriate system. All the interaction with the public administration, except for the most basic operations, require a significant amount of paper forms, including the ever present ID paper photo-copies when these could be easily obtained and checked via an online connection.

Insufficient testing. Different types of testing should be performed before an application runs into production. In many cases, volume and stress testing was omitted so then the application is running in productions, it can not cope with the large number of users. The performance problems are due to design and architectural problems (these types of problems are very hard to fix as they imply rewriting parts of application), to underestimating the components (usually can be easy to fix by upgrading them), or due to programming mistakes (there are the easiest to fix).

Due to the deficiencies, the administration applications become nonfunctional and they have been out of order for periods that can spread for a day to a month. One after each other, the main applications fall one by one: CNAS [14], Border Police IT System [12,33] or ANAF (National Agency for Fiscal Administration) IT system [31].

User friendliness. This feature should be correlated to the application’s usage. If the application is a productivity application, used by trained professionals as public officers, the focus should be on effectiveness, efficiency, and on learnability. A metric to roughly measure those features is the amount of time needed for an officer to perform a specific task. If the digital application is for the use of the citizens, a relevant measure is the easiness to find an information or to solve a problem. In this type of applications, easiness should have a larger impact, and the same metric can be used: in what time an untrained person could find a specific document, find and fill a form, so on. For the application that were implemented, the success rate in resolving a problem online is 67

Digital signature and authentication. Besides the user and password authentication, many public institutions accept and require to sign the papers with a digital certificate. The digital certificate should be issued by a Certified Authority, and it was recommended that the certificates should have availability for one year (although technically certificates can be issued for a period in excess of 20 years). There are a couple of companies that offer digital certificates in Romania, but the overall cost is quite big for a country where the medium net salary is around 3300 RON (around 680 EUR) [23] and it represents 4.4% from the salary. The cost of a qualified digital certificate in Romania varies between 27 EUR to 40 EUR per year, depending if the
certificate is issued for a single year or up to three (3) years and whether it includes or not the price of the physical token.

Estonia found a solution to this problem in order to reduce the cost: the digital certificates were included in the ID cards and were issued by a state authority. Moreover, Estonia increased the availability period of the certificates, so they do not need to be reissued yearly.

A similar legal initiative failed to gain required support in Romanian parliament earlier this year. Romania also had other problems regarding the digital certificates used in public administration as some of the institutions did not accept digitally signed papers. However, due to the COVID pandemic, an OUG was released by the government, forcing all the public institutions to accept digital signed papers.

**Estonia’s model**

Estonia managed to implement in a fast and efficient manner its public services in the last decade, offering its citizens other methods to communicate with the state’s public administration. The digitization process started with the awareness that digitalization can bring a lot of benefits, they created a plan, and started to build and implement applications on a modular base.

Estonia’s president Kersti Kaljulaid mentioned in an interview that Estonia was too poor to support a huge administration, so, in order to offer good quality services to its citizens, it had to digitalize. By digitalizing its public services, Estonia gain competitive advantage. All the digital services are offered by the public administration too, they did not give up paper-based communication, but the number of citizens that use old fashioned communication with the state is constantly decreasing. After having implemented efficient digital systems, president Kersti Kaljulaid affirmed: “Estonia offers more transparency and less bureaucracy!” [15] due to its digital systems. Increasing transparency in governance, building trust in digital society, Estonia saves over 844 years of working time each year.[25]

The Estonian dream is to have as little state as possible, but enough to cover the needs of its citizens. Estonia managed to have 99% of its public services available 24/7 [25], and only a few numbers of public services cannot be performed online: marriages, divorces and real-estate transactions.

Toomas Hendrik Ilves, ex-president of Estonia, concluded in an interview that in fact, going digital is not about technologies, human resource or budget allocations, it’s about “political will, policy, laws and regulations, in that order. In order for it to work, you need laws that underpin the system. You want to define digital identity, then set out the regulations to avoid abuses” [22]

**Sweden’s model**

Sweden was chosen as a model due to the fact the country is a digital leader, it ranks 2nd out of the 28 EU Member States with a score of 69.7 in the Digital Economy and Society Index (DESI) 2020.[5] The digitalization process in Sweden started, as expected with a strategy, in the 1990s, when the government’s IT Commission published the country’s first strategy paper. In spring 2017, the government established the Swedish national digitalisation council. The council was formed by leading experts from universities, by experts from private and public sector under the leadership of Minister for Digital Development. The objectives of the council were: [35].

- **Analyze/follow** – to understand and to identify the challenges associated with digital development process
Promote – to promote to the population the digitization process based on the goal of strategy
Propose - to suggest recommendations

In the same year, the parliament votes for Förvaltningslag – a Administrative Procedure Act in order to help the digital communication. The Swedish government launched a new Digital Strategy whose overall goal was for Sweden to become the number one in the world in the use of digitalisation opportunities. In order achieve their goal, the strategy set a number of targets: Digital security, Digital literacy, Digital leadership, Digital infrastructure, Digital Innovation, each one with a specific objective: [29,30]

- To enhance the digital skills of the citizens, so they could actively participate and take part in the digital transformation process.
- To increase the overall digital security, in order to gain the trust of the people that are using digital services
- To encourage the digital innovation by creating proper conditions to create and to implement value added services.
- To improve digital leadership and to measure and follow up activities
- To consolidate digital infrastructure.

Sweden linked the strategy to specific policy instruments and concrete targets, action plans, budget lines and clear responsibilities. Even if the Swedish public sector is digitally mature, ranking 10th in the EU, there are other countries that are progressing faster, and Sweden will invest in what considers the problematic areas: roll-out of broadband and coverage of remaining sparsely populated areas and the small companies (SMEs) digitalization.[5]

However, nothing was possible without political will, good strategy and organization.

Barriers and Solutions

The main barriers raised in the process of digitalizing the public services in Romania are:

- The lack of political will, including laws and regulations
- The lack of a national strategy
- The lack of coordination between public institutions in setting up such services
- The corruption
- The migration of IT specialists from the public sector to the private sector or to other countries;
- The overall lack of digital skills and conservative mentality
- The resilience to change due to a large bureaucratic administration
- The lack of proper infrastructure in selected areas of the country

In the next paragraph each problem will be detailed and solutions will be provided.

The lack of political will and the lack of a national strategy

In a state that has a corruption problem [31], the lack of transparency is wanted as it offers methods to capture the state, and the digitalization seen as a method to increase transparency and efficiency was not a priority. The digitalization’s solutions should come from political area, an Authority for the Digitalization of Romania (“ADR”) was created in order to elaborate and implement a digital strategy, subordinated to Communication's Minister that currently has only 9 projects in implementation at national level [27]. ADR coordinates the systems: elicitatie.ro (SEAP – The electronic system for Public Acquisitions), ghiseul.ro (SNEP – a national system for taxes payment using cards), edirect.e-guvernare.ro (PCUe – electronic unique point.) However, the implemented systems have major deficiencies, as the information is hard to find, and the flows are not structured properly. The first step is to have a coherent national strategy, based on the population and public administration needs. This strategy should analyse the needs and prioritize the applications, splitting them into modules that are easy to be implemented in
parallel. Which application should be implemented first? A set of factors needs to be analysed in order to take a decide and to prioritize the applications that will be implemented first:

- The impact based on the number of users: the citizens/public administration
- The impact based on the flows and efficiency increase – how faster an answer can be obtained from a public administration?
- The impact on the budget – what are the cost implications and what savings (in man hours and/or in money) will be obtained?
- What is the perception among the citizens (do they reject using digital public services)?
- Is the population conservative or opened to new ideas?

The team that elaborates the national strategy should work with the political factors: government and parliament to pass whatever laws and regulations are needed to be enforced in order to allow public services to be digitalized.

The public pressure, the media can help the digitalization’s process, as the political factor reacts sometimes to the society requests.

The lack of coordination between public institutions in setting up such services

The “transversal” strategy between public institutions is lacking, there are two main issues: interoperability and duplicating services. The applications implemented by/for different public services do not communicate with each other, they do not expose interfaces or exchange data, making data retrieval and processing much harder to perform. Some institutions duplicate the services: instead of having one national, efficient payment system for the taxes, there are city halls that implement their own. Its inefficient, time and resource consuming; according to ghiseni.ro more than 100 city halls around the country decided to integrate with it in order to allow their citizens to pay their taxes, as the administration from Iasi decided to implement their own solution for tax payment (https://dfpl.primaria-iasi.ro/index.aspx). These decisions to have separate applications be translated into duplicate code, duplicate functionality, duplicate maintenance, duplicate hardware – payed from the local/national budget.

A national strategy and national level applications could fix all those duplication problems and could also oblige the institutions to implement interfaces in order to exchange data.

The corruption

The corruption, a stated before is a huge problem in Romania, and its manifesting in two ways: the corruption for application implementation (over pricing, vendor lock in, contract stipulations) and the corruption that will be avoided by using digital and transparent applications. The solution for corruption is neither easy, nor straight forward: it requires transparency, laws that penalize corruption and an efficient and correct justiciary system.

The migration of IT specialists from the public sector

According to the report from the Tax Council, in the fourth semester of 2016, the medium gross salary in the budgetary sector was 6574RON, and in private IT sector, it was around 2000 EUR for senior programers (with more than 4 years’ experience) [13,24]. Due to this difference the work force migrates to the private sector, and the solution would be to hire consultants for design and for management and to either increase the gross payment for these categories of works in the public administrations or to externalize the implementation of the applications.

The overall lack of digital skills and conservative mentality

The overall lack of digital skill of the great majority of the Romania’s population, confirmed by the DESI report, combined with a conservative mentality represents a draw back. In Romania,
a part of the population that has a negative reaction towards the usage of health card (needed by the National Health system), the usage of credit cards and against the chip on the identity cards.

The solution is an information campaign, introducing the digital application one by one, in order for the people to time to see the benefits of going to digital and to learn—either from the younger members of the family or from public courses (that should be taught to everyone).

The resilience to change due to a large bureaucratic administration

According to the INS (National Institute for Statistic) the net salary is 38% greater than the salary in the private sector. Romania also has an oversized budgetary administration, that would require a reform. However, the fact is that an important part of the population is working in the public administration and is satisfied by the amount of work versus salary. There’s a resilience because the general perception that many of the public officers will lose their jobs due to digitalization. Based on the other countries’ statistics, the population did not use digital services in huge numbers overnight; it took years to increase the numbers of citizens that use digital services. By an informative campaign, the resilience might lose momentum, and solutions can be found in order to minimize the impact in case some of the public officers should be laid off—professional reconversion, for example. [13]

Conclusion

Romania was not innovative in the digitalization process, in the gov.ro site, the Government General Secretaryship, mentions regarding strategies for education and digitalization—9 documents, the first one related to education and digitalization dates from 2014!! “STRATEGII DOMENIU 08 – EDUCATIE, CERCETARE, DIGITALIZARE” [26]

The digitalization process started late, compared with Sweden’s, with lack of political support compared to Estonia’s political will. None of these factors help Romania to be effective in the digitalization process. Moreover, Romania is facing another major problem: lack of transparency and corruption cases at the highest level. The corruption scandals were present in the actions related to the implementation of public services, most over, the criteria’s mentioned in request for proposal allowed only specific companies, sometimes these companies bided at lowest price and then signed additional contracts to increase the overall price. Moreover, the delivered software had issues and problems, most of the time, it was badly designed, badly implemented and maintained. The flows did not have the client/citizen in the centre, as information is hard to find, the prefilled forms inexistent and the overall people satisfaction is low. As there are multiple causes, there is no one effective solution – each problem should be addressed individually, political will, experts from private companies and teachers from universities should work together in order to put Romania on the digitalization track. Other important aspect would be the existence of an independent measuring/follow up process (performed by international audit companies) that would verify the results and based on their report coercive measures to be taken. These measures could be: from indefinitely forbidding a company to participate in public auctions for digitalization in Romania if it did not fulfil previous engagements, to personal actions against the people/managers that supervised or signed the contracts if they did not perform their duties.

The last resort for digitalization process in Romania, is EU: by its programmes that are implemented at European level, Romanians, as European citizens should have access to digitalization.
References

[7] Ministru Agriculturii: Suntem 14.600 de bugetari care gestionam o agricultura eminamente privată. Mie mi se pare un pic mult,
[13] Consiliul fiscal, Raport Anual 2019,
[14] Dobrescu P., Sistemul cardurilor de sanatate a picat. CNAS: validarea serviciilor medicale se face offline
[16] eGovernment & Digital Public Services,
[17] European Commision, ” ISA² - Interoperability solutions for public administrations, businesses and citizens”,
[18] European Commission , GRECO report- Romania,
[28] José Marcelo A. P. Cestari, Eduardo de Freitas R. Loures, Eduardo A. P. Santos, Yongxin Liao, Hervé Panetto, Mario Lezoche, An Overview of Attributes Characterization for Interoperability Assessment from the Public Administration Perspective;
[29] OECD. “Enhancing Access to and Sharing of Data : Reconciling Risks and Benefits for Data Re-use across Societies”,
[31] Oncu M., Sistemul IT al Vânilor este nefuncţională de azi-dimineată, iar angajaţii lucrează pe hârtie. Traficul este îngreumat în toate vâmile, mai ales în cele mări,
[33] Soica M., Medicii sunt disperaţi! Sistemul cardurilor de sănătate a cărăpat din nou!
Community structures of autism genes on functional interaction graphs support the 3genic background of complex neurodevelopmental disorders

Csaba Pintér, Ábel Fóthi
ELTE IK, Neural Information Processing Group
csabapinter@protonmail.com

Tissue specific gene functional interaction graphs\[1\] are commonly used tools to analyze the genetic background of complex disorders such as ASD\[2\]. According to the recently emerged 3genic theory\[3\], focusing on the direct effect of a certain number of candidate genes is not enough, as all genes expressed in relevant tissues play an indirect role in complex phenotypes. As it adds up from possibly thousands of small indirect effects, through gene co-regulation, understanding the community structures of these so called 'peripheral genes' has become a crucial task.

The purpose of our research is twofold: We want to validate the tissue specific relevance of gene networks, at the same time analyze the applicability of these networks to gather insights about the 3genic nature of ASD.

For each tissue, we worked with two versions of networks: i) complete graph containing the top edges without modification, ii) subgraph containing only edges that connect core ASD genes. We ran Louvain\[4\] community detection on both the complete and subgraph versions, and analyzed the distribution of genes in these clusterings. According to the 3genic theory, we expect the gene distributions (across clusters of complete versus subgraphs) to be more dependent on relevant tissues (e.g. brain) compared to irrelevant ones (e.g. liver, kidney). Our experiments reinforce this hypothesis. To measure the independence, we used Cramer’s V values where the variables are the intersections of the two types of clusterings.

To gain a more sophisticated understanding, besides the core ASD genes, we also ran this experiment with randomly selected genes. Random selection results in a very strong dependence as it samples from the complete graph and the subgraph inherits its structure, yielding clusters that are corresponding well across the two clusterings. However, building a subgraph on an irrelevant tissue from ASD genes significantly reduces the dependency between its clusters and the complete graphs clusters, as expected. When it comes to ASD-relevant tissues, the dependency between the two clusterings does not drop compared to the random sampling, suggesting that the communities of core genes can be extended to the entire graph.

With these findings we reinforced the applicability of the relatively new 3genic theory to ASD research. Using Gene Ontology enrichment analysis\[5\], we gathered insights about the biological processes\[6\] that best characterize the ASD gene communities.

References


[3] EA Boyle, YI Li, JK Pritchard. An expanded view of complex traits: from polygenic to 3genic Cell, 2017


**Acknowledgements**

This project was supported by the Hungarian Government and co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001).
Explicit solution, for \( n = 7 \), to a Markov-type extremal problem initiated by Schur

Heinz-Joachim Rack, Robert Vajda

Dr. Rack Consulting GmbH, Hagen, Germany
and
Bolyai Institute, University of Szeged, Hungary

heinz-joachim.rack@drrack.com and vajdar@math.u-szeged.hu

Abstract. We consider the solution found by P. Erdős and G. Szegő (1942) respectively by A. Shadrin (2014) to a A. A. Markov-type respectively to a V. A. Markov-type extremal problem which was initiated by I. Schur in 1919. In particular, the said problem is to determine, for algebraic polynomials \( P_n \) of a given degree \( n (n \geq 4) \), and for each \( k \in \{1, 2, \ldots, (n-2)\} \), the extremum

\[
\mu^*_n,k = \sup_{x_0 \in I} N_{n,k,x_0}, \quad \text{where} \quad N_{n,k,x_0} = \sup_{P_n \in B_n} |P_n^{(k+1)}(x_0)|,
\]

and \( I = [-1, 1] \) and \( B_n = \{ P_n : |P_n(x)| \leq 1 \text{ for } x \in I \} \). The said solutions are of a general character and it holds that \( \mu^*_n,1 = N_{n,1,1} \).

Concrete solutions to this problem are known for \( n \in \{4, 5, 6\} \). We turn here to the next higher degree \( n = 7 \) and determine explicitly, for \( k \in \{1, 2, 3, 4, 5\} \) the sought-for extremum \( \mu^*_7,k \) as well as the septic extremizer polynomials (which are, except for \( k = 5 \), normalized proper Zolotarev polynomials with certain optimal parameter values).

To this end we deploy the Abel-Pell differential equation and compute Groebner basis to describe the extrema and the extremizer polynomials by means of algebraic numbers. We then compare (with \( T_7 \) denoting the 7th Chebyshev polynomial of the first kind) the concrete constants \( M^*_7,k = \mu^*_7,k / T_7^{(k)}(1) \) for \( k \in \{1, 2, 3, 4, 5\} \) to the corresponding upper bounds as provided by Shadrin, and in particular we compare the constant \( M^*_7,1 \) to the so-called (asymptotic) Zolotarev-Schur constant \( M_{\infty,1} \), see e.g. the book by S.R. Finch (2003) and item A143295 in the OEIS database of integer sequences.

References


Transfer of (dual) CS-Rickart properties via functors between abelian categories

Simona-Maria Radu

Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania

simonamariar@math.ubbcluj.ro; simonamariar@yahoo.ro

Rickart and dual Rickart objects in abelian categories have been introduced and studied by Crivei, Kör and Olteanu [2, 3]. On one hand, they generalize regular objects in abelian categories in the sense of Dăscălescu, Năstăsescu, Tudorache and Dăuş [6]. On the other hand, Rickart and dual Rickart objects generalize to abelian categories Rickart and dual Rickart modules in the sense of Lee, Rizvi and Roman [7, 8]. Going back to module theory, the study of extending modules (also called CS-modules) and lifting modules has been a fruitful field of research for the last decades, due to their important applications to ring and module theory. Recently, Abyzov, Nhan and Quynh have introduced and studied the concepts of CS-Rickart and dual CS-Rickart modules [1].

Motivated by all the above, we have introduced and studied (dual) relative CS-Rickart objects in abelian categories, as common generalizations of (dual) relative Rickart objects and extending (lifting) objects [4]. We have also investigated the transfer of (dual) relative CS-Rickart properties via functors between abelian categories [5]. We have considered fully faithful functors and adjoint pairs of functors between abelian categories, and we have discussed the cases of coreflective and reflective abelian full subcategories of abelian categories as well as adjoint triples of functors. We have presented several applications to Grothendieck categories and, in particular, to (graded) module and comodule categories. Also, we have derived consequences for endomorphism rings of (graded) modules and comodules.

This is based on joint works [4, 5] with Septimiu Crivei (Babeş-Bolyai University, Cluj-Napoca, Romania).

References


Cell-oriented Programming

Dániel Balázs Rátaí, Zoltán Horváth, Zoltán Porkoláb, Melinda Tóth

ELTE, Eötvös Loránd University, Department of Programming Languages and Compilers 

{danielratai, hz, gsd, toth}@inf.elte.hu

In the currently existing distributed architectures, there are different components which are optimized to fulfill different roles (e.g. databases, message brokers, load-balancers) in the whole architecture. Distributed architectures are highly complex [1]. Several competencies are needed to be able to create and maintain such an environment. Furthermore, these systems have strict limitations in performance optimization as well, because the interaction between the components is limited to their interfaces.

Most of today’s organizations are moving aggressively to adopt more agile, efficient software delivery and IT management practices to meet customers’ evolving expectations. While agile development teams are focused on speed and agility, the traditional mantra of IT operations is maintaining application stability, even if that means slowing down the development. To this end, IT operations often employ rigorous change control processes and enforce infrastructure standardization dictates to maintain control. Historically, IT operations have been delivered as a set of shared services, supported by functional disciplines in areas such as data center services, network management, and security [4].

Autonomous teams can select the programming language best suited to their project. This may include older languages like Java and C++, or new platforms and languages to support specific development use-cases or design requirements (e.g., Node.JS, Go, and Rust). In other cases development teams are eschewing traditional relational databases in favor of NoSQL or document style data stores. These could be advantageous in projects where data requirements are indeterminate or evolving and where speed and scalability is more critical than up-front logical design and data integrity. Examples include MongoDB, PostgreSQL, and Cassandra [4].

Moreover, we see the world moving towards a hybrid computing architecture that includes large centralized cloud data centers, smaller regional edge data centers, and even smaller, local edge micro data center sites. This environment presents unique management challenges that, require a cloud-based software management architecture to thrive in this complex ecosystem [5].

Building a centralized cloud-computing based infrastructure can be already a highly complex problem to solve for many enterprises. There are many architectural questions that have to be made respectively to the requirements what the system needs to achieve. Very often huge teams, hundreds of developers or more are working on systems that can solve a single business problem. As 5G is increasing the need for edge computing, the task of a developer is becoming an even more complex problem. Therefore as designing, deploying and maintaining modern distributed systems become more and more complex, the necessity of radically new, simple and effective software solutions might increase drastically.

We propose a new programming approach, Cell-oriented Programming, aiming to simplify the development of complex distributed applications. It has some similarities with Object-oriented Programming (OOP) [2], but instead of objects, it is built on so-called Cells. COP aims to make it possible to create a homogeneous system where all the computer nodes can have the exact same software environment. The nodes can form one single huge logical computer together, which makes the whole architecture much more simple. The goal is to eliminate the need to deal with databases, stream-processing services, CDN-s, load balancers, distributed operating systems [3], etc. COP aims to make it possible to write code to a distributed system
as easy as we would to one single computer and make distributed computing magnitudes more effective, reliable, safe and easier to code.

When we write a standalone application, we do not need to care with fault tolerance, consistency, atomic transactions [6], network throughput, latency, containers, container orchestration, canary deployment, etc. Everything can synchronously run in the memory; we only need to care with the core logic of our application. The essence of the Cell-oriented Programming is what if we would not have to care about all those things which make distributed scalable computing so hard, and it would be as easy or close as easy as writing a standalone application.

Cell-oriented Programming is based on the entities called Cells. Cells have a behavior and a state just like objects in OOP. But just like the living biological cells the Cells in COP have a survival strategy and they are able to move between computing nodes or even spread themselves to several nodes at the same time. This way they can optimize the performance and ensure fault tolerance at the same time. Furthermore, this way the architecture of the overall system adapts automatically respective to the load what the systems gets. Therefore we have an automatically self-organized architecture with a bottom-up logic, instead of a classical manually designed top-down approach. Developers do not need to care with all the architectural questions and decide about the different components which should be used. They can just code as they would write a standalone application, just in a bit different style as we got used to in case of Object-oriented Programming.

References

Services for an Edge-native application

Anna Reale, Benedek Kovács, Melinda Tóth, Zoltán Horváth
ELTE Eötvös Loránd University, Budapest, Hungary
Ericsson, Budapest, Hungary
{anna.reale, toth.m, hz}@inf.elte.hu benedek.kovacs@ericsson.com

Edge Computing has been depicted as the enabler of several novel solutions and applications, leveraging on low latency and user proximity. Edge-native applications will be deployed on infrastructures that abstract the underlying compute, storage, and networking primitives. Developers and operators dealing with this new variety of applications may interact with application programming interfaces (APIs) exposed by Edge providers or just rely on policies and automation provided by orchestrators. The uniqueness of the Edge infrastructure translates into the necessity to define new specific services to expose at the edge site. Nevertheless, it appears obvious how not all services can fit at the Edge; thus, the need to integrate it within the existing fabric of services available at the Cloud. Few efforts have been done in the standardization to describe in details this complex interactions.

The term Edge-native has been applied, in December 2019, by the The Eclipse Foundation to refer to their new vendor-neutral Working Group. In their announcement [3] Edge computing is stated to be not necessarily different from Cloud computing in terms of computational power or software stack (e.g. containers, Kubernetes, and microservices). The major difference stands in the fact that in Edge Applications we care about the location of the used resources and the transparency of the orchestration.

In the recent works on Edge computing, few authors have also introduced this term [5, 6]. Authors in [5] define an Edge-native application as any software custom-designed to take advantage of one or more of the attributes of Edge computing: bandwidth scalability, low-latency offload, privacy-preserving denaturing, and WAN-failure resiliency. We do argue that privacy-preserving is not a characteristic intrinsic to the Edge. Even though adding an extra tier to the network allows data aggregation, and reduces the amount of data that gets to the Cloud. Security and privacy could potentially benefit from the Edge, but remain highly dependent on the application implementation. Nothing forbids an application running on the edge to log in clear every data sent to it.

In [6] an application is Edge-native not only because it is deeply dependent on services that are only available at the edge; but also because it is written to adapt to scalability-relevant guidance. This implies that the application should be able to scale up and down, not only horizontally (ex. growing resource usage on the same machines/tiers) but also vertically: moving from a device/Tier-3 to Edge/Tier-2 to Cloud/Tier-3.

In our work we expand on this definitions identifying five common characteristics for Edge-native applications: Cloud-native, Edge Driven, Mobile, Geo-Localized and User Equipment Dependent.

After defining what an Edge-native application will be, it is more clear that the role of the communication service provider (CSP) will be twofold. Firstly CSPs will be providing services for edge-native applications that facilitate the usage of the Edge network infrastructure. Nevertheless, they may also play a mediator role between Cloud players (Amazon, Azure, Google Cloud...) and Edge users, facilitating the integration with Cloud services. On a very high level, we can differentiate Cloud services in general and communication services. The former represents all the services already available at the Cloud and that can be integrated into the Edge. The latter is specific for CSP, representing their added value and what should be especially provided for Edge-native applications. We furthermore present a possible architecture to enable provision of both cloud and edge services to Edge native applications.

For these reasons, in this article, we would like to focus, on one hand, on the services provided by the CSP to edge-native applications and the way they are exposed. On the other hand, on

9Application Domain Specific Highly Reliable IT Solutions project has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme.
how Cloud services may be introduced in the networking context [1, 4] to provide Network as a Platform for edge-native applications.

The final output is a proof of concept implementation of the full stack providing Edge native services: a demo Edge native application on a 5G Mobile Edge setup.

The work main contributions are the following:

1. A clear definition for Edge Native applications.
2. A collection of communication services needed at Edge site (surveying the standards and integrating with our owns).
3. A guideline on integration of Cloud services on an Edge platform.
4. An architecture proposal in which CSP allows Enterprises to create Kubernetes clusters, between Edge and partner Cloud providers. Our solution is supported by a Virtual Private Network Certificate Authority in charge of generating and spreading certificates to connect the distributed nodes. CSP services are exposed via Rest API on local host, together with Network exposure Function APIs based on ETSI MANO [2]. The CSP manages a pool of accounts from different Cloud provider to allow the Edge Native applications deployed by the Enterprise to leverage both on cloud and edge.
5. A Proof of Concept of our architecture: an edge-native application, personalized advertisements based on mobile location, exploiting different services, and how those are integrated at the Edge.

References


Containers are lightweight and easy to deploy and configure, so containerization has become an essential approach in modern software engineering [1]. There is still a need for a higher abstraction for service which eliminates further configuration and deployment costs [4]. However, a new service layer is introduced which is called Function As A Service (FAAS) [5]. FAAS enables developers to deploy functions as single code functions. These functions are mostly common operations used by multiple microservices in service solutions. However, as standalone functions are deployed, this approach is called serverless programming.

An image thumbnail creator function in a social platform could be taken as an example for a function which can be shared across social pages feed, instant message conversation and file sharing service as well, where the actual service shows a smaller resolution image of a full-sized picture, helping the association in a collection of images and speeds up page/service load.

The ability of deploying standalone functions leverages code reusability and enables further abstractions. As of the time being most known cloud providers offer FAAS on their palette as a reply to the increasing demand. Yet there is no widespread solution for orchestration over this new kind of service instances. When discussing function as a mathematical term the composition naturally gets mentioned [3]. Having these FAAS instances, functional programming paradigm and the need of orchestration in mind, we have started to gather requirements and planning a functional programming language which can be deployed on a distributed FAAS cluster enabling the development of FAAS function instances in composition, having the functions share the same software project source code having a compiler enabling build-time checks on the function compositions and more. However, using functional programming in cloud-based infrastructure has been proposed [2].

In this paper, we propose an approach for developing serverless applications in a seamless way. We define a programming language that is able to describe complex calculations. We developed a compiler that generates the necessary packets for deploying standalone functions in the serverless realm. Our compiler utilizes the Haskell’s constructs of compiler construction.

References

Optimal Scale for the Classification of Immunohistochemically Stained Colorectal Carcinoma Samples

Zsombor Rigmányi¹, Zsolt Kovács², David Iclanzan¹

1. Sapientia Hungarian University of Transylvania
   Faculty of Technical and Human Sciences Târgu-Mureș
2. George Emil Palade University of Medicine, Pharmacy, Science, and Technology
   Department of Fundamental Pharmaceutical Sciences
   rigmanyi.zsombor@student.ms.sapientia.ro

Introduction

Automatic tumor budding evaluation methods, such as the one presented in [3], use hybrid approaches, combining classical image processing methods with machine learning techniques. In a first phase, tumor bud proposals are obtained through color deconvolution, segmentation and feasible size filtering. In a second phase, image tiles containing the tumor bud proposals and their neighbouring region are presented to a convolutional neural network (CNN) for binary classification.

The tile size in pixels is fixed, determined by the architecture of the CNN. For example, AlexNet [1] or GoogLeNet [2] have an image input size of 224-by-224, while larger architectures such as NASNet-Large [4] use a 331-by-331 input size. Studies typically scale the budding proposal and its region into the fixed tile in a way that allows the largest expected budding to fit exactly, fully covering one or both dimensions.

In this study we examine the effects of the scaling factor with regards to the classification accuracy when performing transfer learning on the GoogLeNet architecture. We determine empirically, what scale performs best for tumor bud proposals found in high-resolution, 40-60 gigapixel scanned tissue specimen samples, where the size of a budding can easily exceed the typical input size of CNN-s.

Materials and methods

Four whole-slide immunohistochemically stained tissue specimens of colorectal carcinoma were obtained from dr. Zsolt Kovács, who is working at the Pathology Department of Mures County Emergency Hospital. These cases belong to the archive of the hospital, and could be used for research purposes. Each image is high-resolution, being around 50 gigapixels. Dr. Zsolt Kovács performed the clinical annotation of the budding regions. 3 out of the 4 samples were positive, containing 71 buddings in total. The average budding size was 632-by-620 pixels, therefore most of the buddings would not fit in the input tiles used by CNN-s at the original resolution.

Next, we generated 224-by-224 pixel tiles containing the annotated buddings at different scale levels: 1/2, 1/3, 1/4 and 1/5. Two sets of tiles were generated for each scale factor (exemplified in fig. 5); in the first the centroid of the budding was placed in the center of the tile while in the second set of images, the centroid was placed randomly within the tile. The first set of images is suitable for classification tasks where budding proposals are first obtained by other means, for example color thresholding and segmentation. The second set amends itself for use cases where the classifier is used as a detector, being shown image regions where the budings do not have an expected position.

To obtain a balanced classification dataset, we generated 71 randomly sampled negative tiles at each scale level. Due to the low number of samples, we resorted to transfer learning using GoogLeNet [2]. The last layer of the 22 layers deep CNN was replaced with a layer containing only 2 neurons, representing the positive and negative classes. The images were split in a 80% - 20% proportions into training and testing sets, the mini batch-size was set to 48, the initial learning rate to $5 \times 10^{-4}$ and the maximum number of training epochs to 10. We experimented with two transfer learning approaches: in the first the network was trained normally while in
the second modification of the weights was restricted to the last layer i.e. all the weights of previous layers were “frozen”. In both cases we performed the transfer learning 10 times at each scale level, with a new random split of the images, and computed the classification accuracy. The best and worst accuracy were removed for each set of 10 runs and the mean accuracy and standard deviation was computed from the remaining 8.

Results and discussion

The results are presented in table ???. The larger scale factors of 1/2 and 1/3 perform the best. As expected, on the easier centered budding classification task the mean accuracy is usually higher. The standard deviations are quite high, this is probably due to the low number of available samples for the transfer learning. More experiments are needed to decide on the usefulness of only modifying the weights of the last layer during transfer learning.

<table>
<thead>
<tr>
<th>Frozen weights</th>
<th>Scale factor</th>
<th>NO</th>
<th>1/2</th>
<th>1/3</th>
<th>1/4</th>
<th>1/5</th>
<th>YES</th>
<th>1/2</th>
<th>1/3</th>
<th>1/4</th>
<th>1/5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centered budding Mean accuracy</td>
<td>87.50%</td>
<td>83.04%</td>
<td>67.86%</td>
<td>76.79%</td>
<td>87.50%</td>
<td>87.50%</td>
<td>63.39%</td>
<td>75.45%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>std</td>
<td>2.70%</td>
<td>7.33%</td>
<td>5.40%</td>
<td>3.31%</td>
<td>2.70%</td>
<td>6.61%</td>
<td>3.70%</td>
<td>4.84%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random placement Mean accuracy</td>
<td>81.70%</td>
<td>75.45%</td>
<td>74.11%</td>
<td>76.79%</td>
<td>82.59%</td>
<td>73.22%</td>
<td>74.11%</td>
<td>77.23%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>std</td>
<td>4.84%</td>
<td>7.25%</td>
<td>3.70%</td>
<td>1.91%</td>
<td>3.54%</td>
<td>5.40%</td>
<td>4.16%</td>
<td>5.70%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Classification accuracies for different scale factors.

References


Primality Proofs with Elliptic Curves: Probabilistic Factorization

Gábor Román

Department of Computer Algebra
Eötvös Loránd University
Budapest, Hungary
romangabor@caesar.elte.hu

Hafner and McCurley [3] gave asymptotic for the number of integers $2 \leq x \leq n$, which can be completely factored by a given algorithm using a certain number of arithmetic operations involving integers of $O(\ln n)$ bits. They’ve examined the trial division method and the elliptic curve method, while applying either the Adleman–Pomerance–Rumely test, or the Solovay–Strassen test for primality testing.

We give a similar asymptotic in the case of probabilistic factorization methods for the expected number of complete factorizations of integers having form of $2 \leq fq \leq n$, where $f$ is a $b(n)$-smooth number, and $q$ is a prime above $b(n)$, with $b(n) : N \rightarrow [e, \sqrt{n})$ being a function. This result plays an important role in the analysis of the Atkin–Morain primality test [1, 2], where during a stage of the recursive descend it is crucial to know the number of complete factorizations of integers having the above described form.

References


Stochastic processes on random graphs with multiple type edges

Bence Rozner
ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Science
Department of Probability Theory and Statistics
bence.rozner@ttk.elte.hu

Stochastic processes on (random) graphs have been used to model infectious diseases on large networks (see e.g. [3, 5, 6]). A social network can be modelled by a graph, where vertices represent the individuals in the population, and two vertices are connected if there is a relationship between the two corresponding entities. In order to understand the spread of an infectious disease on the graph, we assign different states to the vertices (e.g. susceptible, infectious, recovered, carrier, exposed and so on). Then, a discrete or a continuous time stochastic process is defined on the phase space of the states of vertices, where the evolution of the process depends on the structure of the underlying graph.

In many applications, the structure of the graph can be extended by various features, i.e. we can assign some kind of characteristics to the vertices or to the edges. In a social network, infectious diseases are spread through human contact. Since the relationships between the individuals can be different in nature, the probability of spreading is also different among different people. In this study, we examine various types of epidemics on random graphs with multiple type edges. More precisely, we assign a type for the edges of the graph, which are chosen from a finite set of possibilities by certain random dynamics. Then, the infectious disease spreads among the vertices of the graph, so that the probability of infection is different on different types of edges. By using stochastic simulations, we examine the behaviour of the spread of epidemics, when there is also a connection between the types of the edges and the parameters of the process.

In some applications, we can control the spread of the disease, up to a certain level, by separating infected individuals in order to slow down the contagion. We can also assign a state to the edges of the graph, i.e. active or inactive. We assume that the virus cannot spread on inactive edges. At this point, it is clear that if all the edges of the graph are inactive, then the epidemic cannot spread further and all the infected individuals will recover in time, but in practice our goal is to slow down the spread of the infection by eliminating as few connections as possible. Again, by stochastic simulations, we examine the effect of separation (or quarantine), which can be considered as a graph with two types of edges, dynamically changing over time.

We also introduce a model in which a group of vertices of the graph represent the medical employees. In some applications, the infected individuals require some kind of medical treatment. In this case, the medical employees (doctors, nurses, etc.) are assigned to the infected individuals. We assume that the medical employees can also be infected, and then, they also require medical treatment. By stochastic simulations, we examine how much capacity is required in the healthcare for the infected patients to receive appropriate treatment under different parameters of the infection.

The underlying graph models that we have used in this study are the multi-type version of the preferential attachment graph and the model of independent edges ([2]), a generalized version of a random graph model with duplication and deletion ([1]), and the multi-type version of the Erdős–Rényi graph.

We will show that the spread of the epidemic depends on the structure of the underlying graph model, and the introduction of the types of the edges (with the different propagation probabilities) or the quarantine can lead to different results. This is a joint work with Ágnes Backhausz.
References


Stability of triangularization of polynomial matrices
Nikolai Ryzhkov, Ildikó László
Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
namezys@gmail.com, ildiko@inf.elte.hu

Polynomial matrices are very important tools in many areas of applied mathematics, signal processing, engineering, and especially in system theory. In spite of their usefulness, we had to learn that surprisingly R-project [1], which is a free software environment for statistical computing and graphics, did not have a package to handle polynomial matrices.

Because of this reason, we decided to develop a package polyMatrix, which after a severe reviewing process was accepted by R community and now is available on CRAN repository (https://cran.1.org/package=polyMatrix).

In this polyMatrix package there are implemented basic algebraic methods, algebraic functions, and of course many matrix manipulation routines as well. It is well known that in many areas like those mentioned above and others, reduction of polynomial matrices into triangular form is also very important. polyMatrix package contains three algorithms to perform triangularization of polynomial matrices: the extended Euclidean algorithm for polynomial matrices triang_Euclidean(), the Sylvester algorithm triang_Sylvester(), and the interpolation algorithm triang_Interpolation(). The extended Euclidean algorithm uses polynomial matrix operations, but Sylvester- and interpolation algorithms allow triangularization of polynomial matrices by using numerical matrix operations instead.

The essential step in both the Sylvester- and interpolation algorithm is a transformation of a numerical matrix into a lower triangular form that can be done by using LQ-decomposition. This LQ-decomposition is based on the built-in QR-decomposition ([2]), which is a widely used standard tool and is based on the Householder method.

This decomposition is an iterative process, which means that a source matrix is processed row by row, and it contains operations like orthogonalization and normalization as well. Because of these arithmetics errors will be accumulated which affects the numerical stability of triangularization. Besides standard machine precision has to be used, that is not all digits are kept. Also, it is very common that during triangularization tail elements are not calculated exactly, that is they are truncated to zero.

Because of these reasons, we decided to do a study to see what are the real values of the zeros in tails. Then to use this knowledge if it is possible to truncate or round those values in heads, which are smaller than the absolute value of the biggest error in tails.

Because R language does not have the ability to calculate with arbitrary precision, we decided to do different studies in Phyton where we are able to control precision (5-digits to 40-digits).

In many cases polynomials contain only a few nonzero coefficient which will result in a sparse numerical matrix. That means that the triangular form will also be sparse. In our studies we examined both sparse and non-sparse matrices as well. First we took a simple example PM, found in the literature [4], which is a sparse matrix, and performed triangularization (T) calculating tail values as well. We found that the zero elements are not always really zero. We compared these error values with values in head elements. We also did the LQ decomposition of this small matrix PM by hand by using rational numbers in order to compare this result with the result given by the algorithm. We found that the absolute value of the biggest error in tails is bigger than the biggest error in head elements.

Based on this we could round small elements in heads to zero (T').

\[
PM = \begin{pmatrix} x - 1 & x^2 - 1 \\ 2 & 2x + 2 \\ 0 & 3 \end{pmatrix} T = \begin{pmatrix} 1.6x - 1.6 & 5 \cdot 10^{-7}(-x^2 + x) \\ 10^{-6}x + 3 & -1 \cdot 10^{-6}x \\ -0.6x + 1.8 & -1.5 \cdot 10^{-6}x + 3 \end{pmatrix} T' = \begin{pmatrix} 1.6x - 1.6 & 0 \\ 3 & 0 \\ -0.6x + 1.8 & 3 \end{pmatrix}
\]

As a next step, we made a study by using Sylvester matrix representation for polynomial matrices from size 3 x 3 till 15 x 15 and degree of polynomials from 3 to 9 in order to compare errors in tails and heads for higher orders and degrees as well. During LQ decomposition taking advantage of the linear dependency of the rows we could find errors in tails, check tail elements if they are zero or not, calculate the biggest error in tails, and compare with errors in heads.

1. [R-project](https://cran.1.org/package=polyMatrix)
2. [QR-decomposition](https://cran.1.org/package=plyMatrix)
3. [Sylvester algorithm](https://cran.1.org/package=polyMatrix)
4. [Interpolation algorithm](https://cran.1.org/package=polyMatrix)
We found that in the case of randomly generated sparse matrices there are always some head elements which can be round to zero.

We also investigated the behavior of randomly generated non-sparse polynomial matrices. In this case, we calculated LQ-decomposition with different precision to be able to determine the error in heads.

We performed this experiment for randomly generated matrices from size $3 \times 3$ till $25 \times 25$ and degree of polynomials from 3 to 9. To get value precision we performed calculation with 5-digit and 10-digit precision. We found that the order of the errors in tails and errors in heads have almost always the same order. Additionally, we checked this property for matrices of size $7 \times 7$, for polynomials of degree up to 40.

Using tail error information we can round to zero some elements which are smaller than the biggest error in tails. Additionally, we can estimate the precision of coefficients of the polynomial. For example $p(x) = (1 \pm 10^{-5})x^2 + (20 \pm 10^{-5})$.

Taking into consideration both results, information about the maximum absolute value of the error in tails can be used in order to simplify the triangular form by truncating small coefficients of polynomials and to estimate the precision of non zero coefficients of polynomials. Using these results we intend to improve our polyMatrix package.

Figure 6: Error ratio $(e_h/e_t)$ vs polynomial degree for matrices of different sizes

References
Networks of Evolutionary Processors with Length-based Communication

Pramod Kumar Sethy, Erzsébet Csuha-j-Varjú

Department of Algorithms and Their Applications
Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
{pksethy,csuhaj}@inf.elte.hu

Networks of evolutionary processors (NEPs) are graphs where each node is represented by an evolutionary processor (a special variant of rewriting systems) and a finite set of strings. Both the evolutionary processors and the finite languages at the nodes are given over the same alphabet. An evolutionary processor is a finite nonempty set of insertion rules, or deletion rules, or substitution (replacement) rules; each rule inserts, or deletes, or replaces only one symbol. Due to their extreme simplicity, these types of rules are of interest for formal language theory and since they can be considered as formal counterparts of point mutations, they play important role in bio-inspired computing, in DNA computing as well. Each node can be be viewed as a cell having a genetic information encoded in DNA sequences (the strings) which can evolve by local evolutionary events (the point mutations). Each node is specialized just for one of these evolutionary operations.

NEPs operate with rewriting and communication steps, performed alternately. By a rewriting step, the rules of the evolutionary processor are applied to the strings at the node according to a predefined protocol. Usually, one rule is applied to a copy of a string and this is done in all combination and these actions are performed in parallel. That is, all the possible evolution events that can take place do actually take place. After the rewriting step, communication follows: the obtained strings are re-distributed among the nodes according to some predefined communication protocol. A computation in a NEP is a sequence of alternating rewriting and communication steps, starting from the initial state of the NEP (at the beginning, each node has an initial set of strings). A finite computation is successful if certain predefined conditions hold on the state of a distinguished node, called the output node.

The notion of a network of evolutionary processors was introduced in [1], and since then the concept has developed to be a research area, studied in detail. Several variants of the original concept have been investigated, among them the so-called hybrid networks of evolutionary processors (HNEPs) where insertion and deletion is allowed only at the left end or the right end of the string [4]. Several models in NEP theory are proved to be computationally complete (Turing equivalent). It has also been demonstrated that NP-complete problems can be solved by variants of NEPs in linear time.

Communication protocols significantly influence the power and complexity of NEPs. Usually, they are given with a filtering mechanism, i.e., input and/or output filters are associated to the nodes that selects those communicated strings that are allowed to enter the node (that will be element of the set of strings at the node). These filters are defined by context conditions, that prescribe the permitted/forbidden patterns (substrings) of the string to be communicated or has been communicated. These conditions are qualitative conditions. The most common variants of filters are regular conditions.

In this paper we introduce new variants of filters, based on quantitative conditions (for example, the lengths of the strings, the Parikh vectors of the strings). These concepts have been inspired by certain models in DNA computing, called length-separating test tube systems [3], where networks of splicing systems use communication protocols based on the length of communicated word. In [3] it is shown that this type of protocol is sufficient to obtain computational completeness, that is, the model is as powerful as the standard variant of test tube systems with context conditions for communication. To compare the effect of communication protocols for NEPs which are based on qualitative and quantitative conditions, we defined two new types of communication protocols. In the first case, the nodes are associated with input/output filters given by sets of Parikh vectors. In the second case, the communication prescribe the difference of the length of the string when it enters and when it leaves the node. It can be shown that in case of both types of communication protocols computational completeness can be obtained by
using some particular variants of NEPs or HNEPs. To prove the statements, we provide simulations of computation by register machines and generation by phrase structure grammars. We also propose several variants of communication protocols based quantitative properties of NEPs and HNEPs for future investigations, and provide examples for demonstrating their interesting properties.

References

Coupled fixed point theorem and fractals on mixed patterns

Levente Simon\textsuperscript{1,2}, Anna Soós\textsuperscript{1}

\textsuperscript{1}Babeş-Bolyai University, Faculty of Mathematics and Computer Science
\textsuperscript{2}Eötvös Loránd University, Faculty of Informatics

simonl@math.ubbcluj.ro, asooos@math.ubbcluj.ro

The main aim of this talk is to underline a coupled fixed point theorem on sets constructed by finite sequence patterns.

A metric space \((X,d)\) and the operator \(T : X \to X\) has the fixed point \(i \in X\) if there holds that

\[ x = T(x). \]

Fixed sets are also defined for the operator \(T : X \to X\) such that \(A \subseteq X\) is a fixed set if

\[ A = T(A), \]

where \(A \in P_{fp}(X)\). If \((X,d)\) is a metric space and \(T : X \times X \to X\) is an operator, then a coupled fixed point is a pair \((x,y) \in X \times X\) such that

\[ x = T(x,y), y = T(y,x). \]

Coupled fixed sets are interpreted for the operator \(T : X \to X\) such that

\[ A = T(A,B), B = T(B,A), \]

where \((A,B) \in P_{fp}(X) \times P_{fp}(X)\) and \(T(A,B) = \{T(a,b) | a \in A, b \in B\}\).

We generate mixed patterns using a graph-directed fractal operator such that each finite sequence of patterns constructs a set of mixed patterns. A distance based the areas of the black squares constructs a complete metric space on the mixed patterns’ set.

As an initial result from \([9]\), we get that the fractal operator has an unique fixed point of the set generated by the mixed pattern.

This talk underlines a coupled fixed theorem interpreted on a set constructed by finite sequence patterns. The coupled fixed theorem is also connected with the construction of Vicsek fractal.

References


Today, more than ever before, our lives are being projected onto the binary canvas. People ranging from data scientists to less sophisticated users are dealing more and more with data in their daily activities. As users' profiles grow increasingly diverse, so do the size and complexity of data stores. However, our data analysis and comprehension capacities do not follow the same curve.

Users often only have a vague idea of what they’re looking for when delving into their data repositories, e.g., scientists searching for patterns in their data. The exploratory goal becomes more and more clear as the user repeatedly goes through the time-consuming process of asking a question, examining the results, and rephrasing the question, often having to switch between database systems and data mining tools. To get the most out of their data sets, users can end up spending a tremendous amount of time on formulating the right questions.

In this paper, we focus on SQL queryable relations and propose a new version of our query “rewriting” technique [1,2], to help users formulate queries while exploring their data. This time, we ask the user to weigh the importance of the query’s predicates. We introduce a new way to compute the negation query, and conduct an experimental study that shows the feasibility of our approach.

Given a user-specified SQL query, the system reformulates it using machine learning techniques. One of our stated objectives in [2] was to generate a balanced learning set for the machine learning algorithm. In this paper, we propose a new approach to generate the negation query in the presence of user-specified predicate weights, this being the main contribution of our current work.

We consider the class of conjunctive queries (in this abstract, we restrict ourselves to single-table queries). Let $Q$ be a conjunctive query issued by the user. $T$ is a database table. $Q$ has a set of $m$ predicates $p_1, \ldots, p_m$, $m \geq 1$:

$$Q = \pi_{\alpha}(\sigma_{p_1 \land \ldots \land p_m}(T))$$ (22)

Tuples in the result set of $Q$ constitute the so-called positive tuple set. A negation query $\overline{Q}$ returns tuples that are not wanted by the user, i.e., the negative tuple set. In [1], we asked the user to specify a discriminative condition to generate the negation query. In [2], we proposed a Knapsack-based heuristic to find the ”balanced” negation query (the size of the negative tuple set should be as close as possible to the size of the positive tuple set). We now ask the user to weigh the importance of the predicates in the query. This input is used in the generation of the negation query. A decision tree algorithm can then be applied on the resulting learning set; the positive branches of the tree can immediately be translated into a new SQL query.

The user orders $Q$’s predicates on several importance levels, with corresponding ”weights”. The first level contains the ”heaviest” predicates, with weight 0, which are definitely negated in $\overline{Q}$; the second level (weight 1) contains the predicate that is considered next when building the negation query, etc. Each level has one predicate, except for the first one, which can have several predicates.

We denote by $P_0$ the set of predicates with weight 0, by $P_1$ - the predicate with weight 1, etc. $\overline{Q}$ is generated as follows. Its initial version, $\overline{Q}_0$, negates all predicates in $P_0$. Subsequent versions ($\overline{Q}_1$, $\overline{Q}_2$, ...) are created by attempting to add the negated predicate on the next level ($P_1$, $P_2$, ...) to the current version of the negation query. As long as the size of the corresponding negative tuple set gets closer to the size of the positive tuple set, this process continues. Predicates in $Q$ are either negated in $\overline{Q}$ or discarded.
The positive and negative tuple sets are loaded into a C4.5 decision tree [6] implementation. A new query is built from the positive branches of the resulting tree.

Data exploration is an extraordinarily appealing research direction, with numerous papers in recent years [3,4,5,7,8].

References

Scheduling with time restriction and clique search

Sándor Szabó
Institute of Mathematics and Informatics
University of Pécs
Ifjúság u. 6
7624 Pécs, HUNGARY
sszabo7@hotmail.com

Modelling the available storage options in practical batch process scheduling is an important issue. It is customary to distinguish time and capacity type constraints on storage. In this work only time restrictions between consecutive tasks are considered. The problem typically modelled using integer linear and constraint programmings. It also handled in graph theoretical settings. For example in the disjunctive graph or S-graph frameworks. Scheduling can be treated as a maximum clique problem of a suitable constructed conflict (or agreement) graph. In this work we will show how to incorporate the time constraints into this graph based model. We carry out numerical experiments to test the proposed approach.

References

Visualizing Transfer Learning

Róbert Szabó¹, Dániel Katona², Márton Csillag¹, Adrián Csiszárik¹,³, Dániel Varga³

¹Eötvös Loránd University, Budapest, Hungary
²Budapest University of Technology and Economics, Budapest, Hungary
³Alfréd Rényi Institute of Mathematics, Budapest, Hungary

¹dsuciu@cs.ubbcluj.ro
²danut.ilisei@tum.de

Abstract

We provide visualizations of individual neurons of a deep image recognition network during the temporal process of transfer learning. These visualizations qualitatively demonstrate various novel properties of the transfer learning process regarding the speed and characteristics of adaptation, neuron reuse, spatial scale of the represented image features, and behavior of transfer learning on small datasets. We publish the large-scale dataset that we have created for the purposes of this analysis, browsable at https://bit.ly/visualizing-transfer-learning.

Deep neural networks are still commonly conceptualized as black boxes, despite all the recent progress made in interpretability and feature visualization [2, 11, 4, 12, 1, 13].

The current work is following in the footsteps of the Clarity research programme [9, 10, 3, 11], both in the techniques employed, and in the qualitative flavour of the research: creating images of neurons, and trying to identify interesting patterns.

Our main focus is using feature visualization to get a better understanding of what happens during transfer learning, both by comparing neurons before-and-after transfer learning, and by observing what happens during the transfer learning process. We also present a channel visualization technique employing a learned prior [8] utilizing the StyleGAN2 generator [5].

Another output of the current work is a large-scale visualization of the transfer learning behavior of an InceptionV1 network. This mapping of the InceptionV1 network with its 57 convolutional layers and 7280 channels on four datasets resulted in approximately 30 000 visualization images. The produced dataset is presented in a browsable form at https://bit.ly/visualizing-transfer-learning.

Feature visualization via activation maximization

Gradient-based methods of feature visualization strive to maximize the aggregated activation of a network layer, channel, or single neuron by computing the activation’s gradient with respect to the input image, and doing gradient ascent [15, 6, 14, 7, 9]. To achieve good results, some image parametrizations or priors must be added that guide the optimization process to an output interpretable to human observers.

Usually the goal is to add priors that bring the least amount of their own biases, but a particularly interesting exception is the use of generative models: feeding the output of an independently trained generator to the inspected model, and doing regularized gradient ascent in the latent space of the generator [8]. We employ this technique on a CelebA classifier’s top convolutional layer, using the StyleGAN2 generator [5]. In effect, we can solve the highly nonlinear activation maximization of face recognition neurons, constrained to the manifold of face images.
Figure 7: Visualizations for layer $\text{Mixed}_5c\_\text{Branch}_3\_b\_1\times1$ channels. Columns correspond to channels, top row shows Lucid visual, bottom row shows StyleGAN2-based visual. Note the strong correspondence of facial features between the two kinds of visualizations.
Figure 8: Visualization of transfer learning (from ImageNet to CelebA) at different training iterations. Columns show distinct channels, rows show the following iterations, respectively: 0, 10, 20, 30, 60, 150, 1000, 3000. Note that for this experiment the batch size is 10 to visualize a finer grained detail of the transfer learning process.
References


Figure 9: Visualization of transfer learning from ImageNet to CelebA at different layer depths. Each row corresponds to a single layer, odd columns correspond to layer’s first 4 channels pre-transfer, even columns are the same neurons post-transfer. The selected layers are every 7th layer ending with the deepest layer, namely Conv2d_2b_1x1, Mixed_3b_Branch_3_b_1x1, Mixed_4b_Branch_1_a_1x1, Mixed_4c_Branch_2_a_1x1, Mixed_4d_Branch_0_a_1x1, Mixed_4e_Branch_1_b_3x3, Mixed_4f_Branch_2_b_3x3, Mixed_5b_Branch_3_b_1x1. (The periodicity of the InceptionV1 layers is 6.) Layers are deeper from top to down.
Applying Modules for Modern C++ Libraries

Richárd Szalay, Zoltán Porkoláb

Department of Programming Languages and Compilers,
Faculty of Informatics,
Eötvös Loránd University,
Budapest, Hungary

szalayrichard@inf.elte.hu, gsd@elte.hu

Modules [4] is an upcoming feature of the newest version of the C++ standard, C++20. With this new language element, visibility control of symbols from libraries – similarly to the packages in Java, for example – can be achieved. However, there are a sizeable corpus of C++ software projects that were developed against older standards, without using this feature. While Modules is standardised in a way that compatibility of importing a conventionally #included header is given, questions arise when existing projects are to be migrated to this feature.

The first question is from the way generative programming (templates) is done in C++ [5]. Templates are parsed by each compiler invocation and are instantiated based on the usage in the client code. A common practice before Modules was to put implementation detail into the detail namespace, and as a rule of thumb forbid users from directly interfering with such. With the private symbols feature of Modules, the natural approach would be putting the detail namespaces into a private part, and make the symbols not nameable from client code.

When it comes to upgrading existing libraries to such a new approach, in the world of machine-code compiled software such as those written in C++, Application Binary Interface (ABI) compatibility [1] must be ensured. In this paper, we will investigate the applicability of Modules to the designing of a “greenfield” program library, and also to existing large libraries, such as the famously template-heavy Boost library collection. For an already existing library of a history of multiple decades, it is imperative that backwards compatibility is kept towards clients using older versions of the C++ standard. Previous works detail re-engineering existing corporate code bases to Modules, but the amount of kept backwards-compatibility is questionable [2]. It is also a challenge to investigate how much restructuring will the moving to Modules require. Previously, “unity build” was suggested as a workaround for compilation times [3], which can not always be done without changes to the project at hand that break client compatibility, either on the Application Programming Interface (API) or the ABI.

References


In situ Enhancement of Type Safety using Fictive Types

Richárd Szalay, Ábel Sinkovics, Zoltán Porkoláb

Department of Programming Languages and Compilers,
Faculty of Informatics,
Eötvös Loránd University,
Budapest, Hungary

szalayrichard@inf.elte.hu, {abel,gsd}@elte.hu

Type systems are a crucial tool in the hands of software developers for ensuring an increased level of soundness to their programs, and to guard against bugs. However, in many cases, the type systems are not used to their full capabilities, and trade-offs are made. One infamous accident that resulted from the lack of type safety is the case of the exploded Mars probe [6], where developers using both metric and imperial measurements mixed these values without annotating the unit, leading to excessive throttling which caused the accident. Various widely used programming languages, such as C++, Java, Python, Rust are yet to implement constrained types as a language feature. In most cases, constrained types, which are the easiest way to ensure a unit-aware type system might only exist as libraries, such as <chrono> for C++. Extensive type systems are readily available in functional programming languages, however, with the convergence of paradigms and the availability of functional elements in traditionally object-oriented or imperative languages [1], it is a reasonable expectation to raise the level of safety expected from software. C++ has supported compile-time functional programming elements through template metaprogramming [4], but ensuring both compile-time validation of operations and run-time validation of data is not possible automatically. While user-defined types are common in modern languages that support such notion, developers resort to using the most conceptually fundamental types, such as int or string, as they are, and not associating any “strong type” [3] to range-constrained integers.

The goal of refactoring efforts is to produce equivalent software that is “better” by some metric or idea. Simply defining a new type and changing a program element’s type to this new type is not sufficient, as the code will break due to not defined operations and conversions. A non-compiling state of the project is detrimental, as the refactoring tools often rely on the code being in an invariant-compliant state, and thus can not be used anymore. As such, incremental refactoring is needed [7]. This can be achieved primarily by investigating the landscape and finding offending types and creating automated rules that refactor the code based on previously explored knowledge. Such an effort works for unit-based systems, such as timekeeping [8] or physics [5].

In this paper, we investigate an incremental approach that uses code annotations. The upside of the approach is that information encoded in annotations can be easily understood by tools without changing the semantics of the code at hand. Previously, type migration has been done in a well-scaling manner between already existing types [2]. We propose a method by which the developer can paint a particular program element (of any type in the type system, e.g. int) with an annotation, which will be propagated by tools across the software project. Until this propagation is done, the existing code is not altered and can still be compiled and developed. Once the propagation has finished, the insight gained from the operations required on this fledgeling type (e.g. velocity) can be reviewed, and the new type can be generated. Afterwards, usage points are rewritten to this new type.

References


Numerical methods for space-dependent SIR mod2 with constant delay

Bálint Takács1, Róbert Horváth2, István Faragó1,2

1Eötvös Loránd University, Budapest, Hungary
2Budapest University of Technology and Economics, Hungary

takacsbm@caesar.elte.hu, rhorvath@math.bme.hu, faragois@cs.elte.hu

The SIR model, first introduced by Kermack and McKendrick [5] in 1927 can be used to describe any process in which some property is passed among a group of individuals. During the process, we distinguish three classes: the first one (labeled by S) contains the ones which has not acquired the property yet, the second (denoted by I) has those which have the property and have the ability to pass it on to others, and the last one (class R) contains the ones which had the property, but they cannot transmit it any more. Such processes include epidemics (the process the model first was introduced for) or other biological phenomena like a fire in a forest.

The original system of ordinary differential equations were extended introducing a spatial dependence (like in [4]), resulting in a system of partial integro-differential equations. Then, a constant delay in time is introduced to model processes in which individuals do not have the ability to pass over the property modeled in the process right when they get the aforementioned property, but only after a given time. In the talk we are going to show several numerical mod2 arising from this continuous system. First we use different techniques to discretize our problem in space (namely, approximate the integral in the equations), and then solve the obtained system of ordinary differential equations using time integration methods, like the method of Elsgolts [2], the method of steps introduced by Bellman [1] and strongly positivity preserving Runge-Kutta methods [3].

First we are going to prove that the solutions of the original, continuous equations behave in a biologically reasonable way. Then we show that the numerical mod2 preserve the qualitative properties of the original continuous model for a sufficiently chosen time-step, and then show the connection between this bound and the ones acquired in the case of the equations without delay. The theoretical results are then demonstrated on numerical experiments.

References

Error detection and analysis of packet processing paths of P4 programs

Gabriella Tóth, Máté Tejfel

Faculty of Informatics, Eötvös Loránd University
kistoth@inf.elte.hu, matej@inf.elte.hu

In this paper, we introduce a solution for error detection and analysis of P4 programs. Our solution does not only contain error cases but suspicious ones, which can cause errors. These can be caused by the usage of invalid header or uninitialized fields, incorrect reading or writing of metadata, or improper usage of the drop flag. The analysis of the program is based only on the P4 source and separates the handling of the ingress and egress pipeline. A prototype of the error detector and analyzer is created with which we checked the possible error cases in many P4 programs, and monitor the different way for packet processing, which results will be shown in this paper.

P4 [3, 1] is a domain-specific programming language to develop the processing of network packets in network devices. In the code, we can define what kind of header information we would like to handle. The programs have three main parts: parser, modifier and deparser. The parser defines the way how the program gets the header information from the input packet, the modifier describes how it changes the header information, and after the modification, the deparser contains that how it creates the new output packet from the new header information.

P4 creates a more flexible way to develop network devices, although this makes it easier to develop inconsistent and incorrect programs. Therefore, different approaches have already been created to verify and analyze P4 programs. There are some tools – for example P4V [5] and Assert-P4 [4] – which expect annotated programs to check the given properties. With these solutions, the developers can get more specific analysis but need to do more work to learn the language of the annotations and how to create them. We would like to give a solution, which only uses the simple source of the program for the checking. There are similar approaches for this too - for example, Vera [6] - where they use symbolic execution. A new area in our result is to check the correctness of the programs based on the PSA, which is a new approach with static analysis.

The base of this research was written in our previous publication [7], where we showed the theoretical part of the idea to detect specific errors in P4 programs. In this paper, we supplement this idea with new case analysis, detection, and the practical part, as a prototype of the analyzer tool.

The checking is based only on the P4 source from which the specification of the program is created; it gets the precondition from the parser, the post-condition from the deparser, and the description of the functional part of the program from the modifier. The method checks if the execution of the program, starting from any initial state (where the precondition is true), will reach one of the final states (where the post-condition is true). During the checking, it stores the validity of the used header information and checks the correct usage of them in all possible execution paths.

Besides the validity checking, the solution was extended to handle the ingress and egress pipeline separately. In Figure 10 we can see the different paths of the packet processing. If we only see the most simple path, then first the ingress pipeline gets the input packet and processes it, and based on this result, the egress pipeline continues the processing.

Although, the path of packet processing can be more complicated. It depends on the definition of the target architecture. We work with the latest, official one, the Portable Switch Architecture (PSA). As it can be seen in Figure 10, there are three ways for the more complex paths: the resubmit, in which after the ingress deparser, the packet goes back to the ingress parser; the CE2E, when a clone of the packet goes through the egress again; and the recirculate,
in which after the execution of the egress deparser, it continues with the beginning, i.e. ingress parser.

![Figure 10: Packet processing paths [2]](image)

The way, how the packet goes, is defined by the specification of the PSA. We can check if the P4 program is correct for the PSA by simulating all of the possible packet paths to give further information for the developer about the execution. During this calculation, we can extend our solution to work with the metadata too, besides the header information.

In both of the ingress and egress pipelines, there are limitations of the usage of certain metadata — among others in standard metadata, there is a field egress spec, which can be written only in the ingress pipeline, and ingress port is only readable. The improper usage of these types of data can be easily checked and reported to the developer.

The drop of the packet can be controlled too. Possible drop paths and suspicious drop usages — for example drop the packet twice in a path or undo the drop of the packet — can be reported.

Our goal is to create a tool for P4 developers to make their work easier by giving a report about their P4 code. These reports contain error and warning detection while giving some useful results of the analysis, and all of it calculated only from the P4 code.

References


Real-life applications of pseudorandom generators

Viktória Tóth, Robin Kiss
Department of Computer Algebra, Eötvös Loránd University
viktoria@inf.elte.hu, gekko42@gmail.com

Recently a constructive theory of pseudorandomness of finite sequences has been developed and many constructions for binary sequences with strong pseudorandom properties have been given.

We used the theory and notions presented by Mauduit and Sárközy in [4]. They initiated several measures of pseudorandomness. The sequences can be considered as pseudorandom, if its measures behave as the measures of a real random sequence. They proved that it is true for some very important binary families. Furthermore, it is proved in [6] and in [7] that they have other strong pseudorandom properties as well: these families are collision free and possess the strict avalanche effect. Both mentioned pseudorandom properties have practical importance, e.g. in cryptographic applications.

Now our goal was testing the constructions in “real life”. We could verify that these good constructions have not only mathematically provable nice properties, but they can be used easily and in a fast way in applications as well.

We generated and tested these constructions with a computer program written in C++. We analyzed the families of binary sequences generated by the extended Legendre symbol (see in [2]) and the construction using additive characters (see in [3]) by computing the well-distribution measure and the correlation measure (the definitions of the measures see in [4]) of them. We used elliptic curves over finite fields in both constructions. We set together the measures of the families generated with variable initial parameters, then checked that all these measures suited the requirements presented in [1] and in [5]. Since the proved bounds are asymptotic, it was interesting to study the behavior of the measures in practice.

Thus our observations show that the practice correspond with the theory. It means that the constructions, which have mathematically provable strong pseudorandom properties, can be used in real-life applications as well.

Supported by the project "Integrált kutatói utánpótlás-képzési program az informatika és számítás-tudomány diszciplínáris területein (Integrated program for training new generation of researchers in the disciplinary fields of computer science) ", No. EFOP-3.6.3-VEKOP-16-2017-00002. The project has been supported by the European Union and co-funded by the European Social Fund.

References


Another generalization of Euler’s arithmetic function and of Menon’s identity

László Tóth
Department of Mathematics, University of Pécs
ltoth@gamma.ttk.pte.hu

We define the \( k \)-dimensional generalized Euler function \( \varphi_k(n) \) as the number of ordered \( k \)-tuples \((a_1, \ldots, a_k) \in \mathbb{N}^k\) such that \( 1 \leq a_1, \ldots, a_k \leq n \) and both the product \( a_1 \cdots a_k \) and the sum \( a_1 + \cdots + a_k \) are prime to \( n \). If \( k = 1 \), then \( \varphi_1(n) = \varphi(n) \) is the classical Euler function.

We prove the following new results. Let \( \tau(n) \) denote the number of divisors of \( n \), let \( \mu \) denote the Möbius function and \( * \) the Dirichlet convolution of arithmetic functions.

**Theorem 19** For every \( k, n \in \mathbb{N} \),

\[
\varphi_k(n) = \varphi(n)^k \prod_{p \mid n} \left(1 - \frac{1}{p-1} + \frac{1}{(p-1)^2} - \cdots + \frac{1}{(p-1)^{k-1}}\right)
\]

(23)

\[
= n^k \prod_{p \mid n} \left(1 - \frac{1}{p}\right) \left((1 - \frac{1}{p})^k - \frac{(-1)^k}{p^k}\right).
\]

(24)

**Theorem 20** Let \( k \geq 2 \) be fixed. Then

\[
\sum_{n \leq x} \varphi_k(n) = \frac{C_k}{k+1} x^{k+1} + O \left(x^k (\log x)^{k+1}\right),
\]

(25)

where

\[
C_k = \prod_p \left(1 + \frac{1}{p^{k+1}} \left((1 - \frac{1}{p}) \left((p-1)^k - \frac{(-1)^k}{p^k}\right)\right)\right).
\]

(26)

**Theorem 21** Let \( f \) be an arbitrary arithmetic function. Then for every \( k, n \in \mathbb{N} \),

\[
\sum_{a_1 \cdots a_k = 1 \atop \gcd(a_1, \ldots, a_k, n) = 1} f(\gcd(a_1 + \cdots + a_k - 1, n)) = \varphi_k(n) \sum_{d \mid n} \frac{(\mu * f)(d)}{\varphi(d)}.
\]

(27)

**Collorary 22** \((f(n) = n)\) For every \( k, n \in \mathbb{N} \),

\[
\sum_{a_1 \cdots a_k = 1 \atop \gcd(a_1, \ldots, a_k - 1, n) = 1} \gcd(a_1 + \cdots + a_k, n) = \varphi_k(n) \tau(n).
\]

(28)

Identity (6) is a new generalization of Menon’s identity. If \( k = 1 \), then it gives Menon’s classical identity. If \( k = 2 \), then it reduces to the identity obtained by Sita Ramaiah [2] and investigated recently by Ji and Wang [1, 3].

**References**


Local fixed point theorems and open mapping principles for generalized contractions

Radu Trușcă
Faculty of Mathematics and Computer Science, Babeș-Bolyai University
radu.t10ca[at]math.ubbcluj.ro

In this talk we will present some fixed point theorems for generalized contractions defined on an open ball in a complete metric space. We will consider Reich type contractions, Chatterjea type contractions and Berinde type contractions. As an application, open mapping principles are given for the above mentioned operators.

References

A family of barely expansive polynomials

M. J. Uray
Eötvös Loránd University Budapest
Faculty of Informatics
Department of Computer Algebra
uray.janos@inf.elte.hu

In this paper, \( f(x) \) denotes an integer polynomial of degree \( n \) and coefficients \( a_i \in \mathbb{Z} \), i.e.\[ f(x) = a_n x^n + \ldots + a_1 x + a_0 \text{ with } n \geq 1 \text{ and } a_n \neq 0. \]

We are interested in the following type of polynomials:

**Definition 23** The polynomial \( f \) is expansive if all of its roots lie outside the unit circle, i.e. for all roots \( x_i \) (either real or complex) we have: \( |x_i| > 1 \).

We estimate the following quantity:

**Definition 24** The expansivity gap of an expansive polynomial, whose roots are \( x_1, x_2, \ldots, x_n \), is:

\[ \varepsilon := \min_{i=1}^{n} |x_i| - 1. \]

Our goal is to find out how small the expansivity gap of an integer polynomial can be, i.e. how close the roots can be to the unit circle, or in other words, how close the polynomial is to being non-expansive. In doing so, we estimate the expansivity gap using the degree \( n \) of the polynomial and the following property:

**Definition 25** Denote by \( H(f) \) the height of \( f(x) \): \( H(f) := \max_{i=0}^{n} |a_i| \).

In an earlier paper [2], the author proved the following:

**Theorem 26** The expansivity gap of an expansive integer polynomial \( f(x) \) has the following lower bound:

\[ \varepsilon \geq \frac{1}{{n\choose 2}! H^{n-1} + {n\choose 2} + 1}, \]

where \( H := H(f) \) is the height of \( f(x) \), and \( n \geq 3 \).

This means that the roots cannot be arbitrarily close to the unit circle, at least for fixed parameters \( n \) and \( H \).

In this paper, we examine the sharpness of this lower bound, i.e. how close it is to the best possible such lower bound. The main result of this paper is the following:

**Theorem 27** For each \( n \geq 2 \) and for each sufficiently large \( H \), there exists an expansive integer polynomial \( f(x) \) of degree \( n \) and height \( H \) whose expansivity gap is:

\[ \varepsilon = \frac{1}{2H^{n-1}} + O \left( \frac{1}{H^n} \right). \]

This means that the lower bound in Theorem 26 is quite close, and in fact, its dependence on \( H \) is asymptotically sharp. The proof of this statement is constructive, i.e. we define the family of expansive integer polynomials which has so small expansivity gap.

For the construction of these barely expansive polynomials, we use a certain table of values, the so-called Motzkin triangle [1]. Let \( M_{n,k} \) \( (n,k \geq 0) \) be the number of possible paths from the origin \((0,0)\) to the point \((n,k)\) using the steps \((1,1),(1,0),(1,-1)\) and never going below the \(i\)-axis. The first few values are:

<table>
<thead>
<tr>
<th>( M_{n,k} )</th>
<th>( k = 0 )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n = 0 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>12</td>
<td>9</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
It is convenient to set $M_{-1,-1} := 1$ and $M_{-1,k} = M_{n,-1} := 0$ for $k, n \geq 0$. The $M_{n,k}$-values have a simple recurrence relation:

$$M_{n,k} = M_{n-1,k-1} + M_{n-1,k} + M_{n-1,k+1} \quad (n, k \geq 0). \tag{29}$$

The family of polynomials for which Theorem 27 will be proved, is defined below, using the Motzkin numbers. For each $n \geq 2$ and $H \geq 1$, define the coefficients of $f(x)$ as follows:

$$a_0 := H$$
$$a_1 := H - (M_{n-3,0} + 1)$$
$$a_2 := H - M_{n-2,0}$$
$$a_i := -M_{n-2,i-2} \quad (3 \leq i \leq n)$$

For the first few degrees, these polynomials are:

- $n = 2 : \quad (H - 1)x^2 + (H - 1)x + H$
- $n = 3 : \quad -x^3 + (H - 1)x^2 + (H - 2)x + H$
- $n = 4 : \quad -x^4 - 2x^3 + (H - 2)x^2 + (H - 2)x + H$
- $n = 5 : \quad -x^5 - 3x^4 - 5x^3 + (H - 4)x^2 + (H - 3)x + H$
- $n = 6 : \quad -x^6 - 4x^5 - 9x^4 - 12x^3 + (H - 9)x^2 + (H - 5)x + H$

The full paper contains the proof of Theorem 27 for these polynomials. We use certain linear algebra tools, including some determinant-based conditions on the expansivity of polynomials, which was developed by the author in his previous work [2].

References


Refactoring concurrent Erlang applications for distribution

Balázs Varga, István Bozó, Melinda Tóth
ELTE, Eötvös Loránd University, Faculty of Informatics,
Department of Programming Languages and Compilers
balazsvarga@student.elte.hu, bozoistvan@elte.hu, tothmelinda@elte.hu

Applications are getting more and more complex and require an increasing amount of resources. Distributed software are able to scale easily to take advantage of the available resources. They offer great reliability and fault tolerance. The development of a software project might start with distribution not in mind. At some point, as the complexity of the software increases, the developer might realize that the originally concurrently written program should be transformed to function in a distributed manner.

Making concurrent programs distributed is a wide topic, and as such can be approached in many ways. We have focused on Erlang programs, analyzing and transforming them before runtime. Our approach is based on static program analysis and syntax tree transformations.

Erlang [1] is a functional programming language built for distributed systems. It can be used to easily express concurrent and distributed concepts at the language level. It uses the actor model of concurrency, where lightweight processes running on nodes (Erlang Virtual Machines) communicate with each other using message passing. Concurrent and distributed Erlang programs use similar language constructs, making them ideal for transforming one into the other. The communication between processes on different nodes is handled transparently.

Our goal has been to transform concurrently written, single-node Erlang applications to work in a distributed environment across multiple nodes. Specifically, we have focused on the task of moving a set of processes to be spawned on a different node with a known name. This transformation involves many aspects, as processes generally interact with each other using message passing, as well as with various resources on the node. When moving a process to a different node, much care needs to be taken, so that the communication and the access to resources stays intact. Therefore, this transformation, when done manually, is tedious and error-prone.

Erlang programs can be analyzed by static analysis frameworks such as RefactorErl [2, 3]. By examining the code statically, without running it, RefactorErl builds up a Semantic Program Graph that represents the syntactic and semantic relationships within the program. It supports various simple and complex queries, which we can use to understand and reason about concurrent Erlang code. RefactorErl is also a refactoring tool, so it is able to perform syntax tree transformations, allowing us to programmatically define modifications of the code.

Such tools have motivated us to explore ways to use static analysis to automate the process of discovering and transforming portions of the code that must be changed to move a process to another node. This is not a trivial task, as processes, as well as connections between sent and received messages, are syntactically implicit in Erlang [4]. We have made use of queries and transformations that are available in RefactorErl, but have defined them in a general, tool- and framework-agnostic way.

The main contribution of our research is the definition of transformation schemes [5] focused on changing the spawning of easily movable processes, as well as processes that are bound by local registration. We have devised transformations that cause the registration to happen on the new node, so that the moved process can still be referenced by its registered name. We have also considered the passing of messages between processes on the original node and the moved process. We have defined methods to find the relevant send expressions in the code, and

---

12 The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002), and by the ÚNKP-19-1 New National Excellence Program of Ministry for Innovation and Technology. The research is part of the "Application Domain Specific Highly Reliable IT Solutions" project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme.
transform them to work across nodes. For the messages from the moved process to the original
node, we have defined a series of transformations to dynamically determine and pass the name
of the original node to the appropriate send expressions at runtime.

For all transformation schemes, we have explored the preconditions along with the necessary
compensations and considerations. We have defined methods to find and select the relevant
portions of the syntax tree using static analysis, and the changes that need to be made in the code
to achieve the desired effect. We have implemented the basic cases of the transformations using
the open-source RefactorErl framework. To evaluate our results, we have used the RefactorErl
implementation to transform various open-source examples, and compared the functionality of
the original and transformed programs.

References


Central European Functional Programming Summer School – Fourth Summer School, CEFP
2011, Revisited Selected Lectures, Lecture Notes in Computer Science (LNCS), Vol. 7241,

- source code analysis and refactoring in Erlang. In Proceedings of the 12th Symposium
Tallin, Estonia, October 2011.

[4] Huiqing Li, Simon Thompson, György Orosz, and Melinda Tóth. Refactoring with Wrangler,
updated: Data and process refactorings, and integration with eclipse. In Proceedings of the
7th ACM SIGPLAN Workshop on ERLANG, ERLANG '08, page 61–72, New York, NY,
USA, 2008. Association for Computing Machinery.

Student Association Conference, Faculty of Informatics, Eötvös Loránd University, May
2020, Received 1st prize.
Label Propagation with Graph Neural Networks in Interactive Video Segmentation Annotation

Viktor Varga
Department of Artificial Intelligence, Eötvös Loránd University, Budapest
vv@inf.elte.hu

Supervised methods in deep learning usually require large amounts of labelled training data in order to avoid overfitting. While different approaches aim to automatically label data, achieving high quality annotation still involves manual checking and correction of labels. A number of present day machine learning applications are trained with densely labelled image sequences [5, 2]. Semantic segmentation of videos is required to indicate the exact position and outline of objects in camera images. However, precise labeling of this type even in short clips could take hours of human labour. Interactive annotation of image sequences aim to utilize automatic tools for labeling and only depend upon human supervision in as few cases as possible.

Numerous research questions arise in the field of interactive annotation. Asking the user the fewest times possible [13, 9] and propagating sparse user labeling across the whole video [11] are probably the most important of them. In this paper, we intend to examine the second question. The foundation of our approach is a superpixel segmentation of the images. Superpixel segmentation is often used in computer vision applications to reduce the dimensionality of the problem and extract information about the spatial coherence of pixels [12, 8]. We build a graph with the nodes being the superpixels and connect them with edges based on spatial adjacency of segments within images and temporal causality of segments in consecutive images of the video. We approximate temporal causality by optical flow estimation [6]. Several papers utilize a similar, graph based setup in interactive video segmentation [7, 4].

However, the semantics of labels used in video segmentation may vary between different videos: specific object categories might not even appear during the training of our model. On the other hand, neural network based models require a fixed label set by default. As a result of that, classic approaches (e.g. Markov Random Fields) [1] were the state-of-the-art until recently in graphical label propagation problems. On the contrary, our approach utilizes a graph neural network [3]. We eliminate the problem of varying label sets by randomly partitioning the label set into foreground and background, and we train our network to solve the binary classification of segment nodes in the presence of sparse ground truth annotation. We show that our approach significantly outperforms classic techniques on the DAVIS [10] video segmentation dataset.

Acknowledgements

Viktor Varga was supported by the Hungarian Government and co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002: EFOP-3.6.3-VEKOP-16-2017-00002, Integrated Program for Training New Generation of Scientists in the Fields of Computer Science).

References


[8] László Kopácsi et al. “Common Fate Based Episodic Segmentation by Combining Supervoxels with Deep Neural Networks”. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019.


Scale-free Loss for Absolute Pose Estimation
Márton Véges
Eötvös Loránd University, Department of Artificial Intelligence
vegesm@inf.elte.hu

The task of human 3D pose estimation is to predict the coordinates of the joints of a person based on an input video or image. There are numerous potential applications, including entertainment, sport analytics or automated physiotherapy. The output of such an algorithm is a skeleton representing the pose of the human in the image.

While a large part of the literature focuses on relative pose estimation [1, 4], where the coordinates of the joints are estimated relative to the hip, recently absolute pose estimation also received increased attention [3, 6]. In the latter, not only the pose of the person is predicted, but their location as well. This might be needed in certain applications, for instance analyzing interactions between multiple people.

It is important to note that the prediction of the pose and location of a person is an under-defined problem. Unless multiple cameras are available, a single image does not contain enough information about the scale of the real-world scene. In other words, we can only predict the coordinates up to a constant multiplier. We also note that often this is not a problem, since we are interested in the relative position of the actors on a video, for instance.

However, most papers do not account for this and reports errors using the Mean per joint position error (MPJPE) metric which is simply the average euclidean error over all joints and all people in the dataset [6, 3]. This error is often calculated in millimeters. Another recently introduced metric is the Normalized MPJPE (N-MPJPE) [5], that, unlike the original MPJPE, does not take into account the scale of the prediction. Specifically, it is defined as:

$$\min_{s \in \mathbb{R}} \| s \hat{P} - P \|_2,$$

where $P$ is the ground-truth pose and $\hat{P}$ is the predicted value. This error effectively searches for the optimal scaling factor $s$ and calculates the error using this factor.

While this metric reports a more realistic error, the authors of [5] still used an $L_2$ error during training, thus the network still had to learn the scaling of the image. We propose an alternative loss function, based on the N-MPJPE metric, that is scale-free thus the network do not have to learn the scaling of the image. The $s$ in the definition of the N-MPJPE metric can be analytically expressed, and this form is then plugged in a regular $L_1$ loss, resulting in a scale-free loss.

We test our new loss on the MuPoTS-3D dataset [2] that contains videos of multiple people performing various activities. The database is diverse, having both outdoor and indoor shots, recorded on different cameras with different resolution and focal lengths. We show an improvement on the N-MPJPE metric with our new loss. Interestingly, it also results in better MPJPE errors.

Acknowledgements
Márton Véges was supported by the Hungarian Government and co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002: EFOP-3.6.3-VEKOP-16-2017-00002, Integrated Program for Training New Generation of Scientists in the Fields of Computer Science).

References


The dynamics of the Hopfield model for homogeneous weight matrix

Anita Windisch, Péter L. Simon
Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University

The dynamics of the Hopfield model for homogeneous weight matrix

One of the simplest networks that implements autoassociative memory is the Hopfield network. This can be represented as a fully connected graph where the nodes are the neurons and there are directed and weighted edges between each of them which represent the strengths of connections. The Hopfield network is a recurrent neural network as signals coming out from a neuron can flow back to itself via other nodes and that is why the spread of the information is not obvious. The Hopfield model\cite{1} is a system of ordinary differential equations which describes the change that happens in the state of the neurons. Its general form is \cite{2}

\[ \dot{x} = Dx + Wy, \quad y_i = f(x_i), \quad (30) \]

where \( i \) denotes the membrane potential and \( y \) represents the firing rate of the neurons, the matrix \( W \) contains the strengths of connections and \( f \) is the activation function. The main goal is to determine the dynamics of this model when the weight matrix \( W \) has a special structure, using both analytical and numerical tools.

We suppose that the weights coming from a given neuron are the same, i.e. \( w_{ij} = w_j \) for all \( i \neq j \), and none of the neurons are connected to themselves directly, that is every diagonal element of \( W \) is zero. We apply the sigmoid function \( f(x) = (1 + \exp(a - bx))^{-1} \) as activation function and assume that neurons do not receive any external input. We show that if the network contains such neurons that have the same positive weight then to describe the asymptotic behaviour of the model it is enough to investigate a lower dimensional system of differential equations.

First, we assume that every neuron has the same positive weight \( w \) and we choose \( a \) and \( w \) as bifurcation parameters. In this case the saddle-node bifurcation curve can be expressed explicitly. This curve consists of those parameter values where the number of equilibria changes. The saddle-node bifurcation curve divides the parameter plane into some domains where we investigate the number and the stability of the fixed points.

If there are some neurons in the network which have different weights than others have then the behaviour of the Hopfield model is much more complicated. Several articles have already been written about different special cases such as \cite{3}. We assume that there is only one neuron in the network which has an arbitrary weight \( w_1 \) and all of the others have the same positive weight \( w \). In this special case we determine the saddle-node bifurcation curve depending on the weights \( w \) and \( w_1 \). We also specify the Andronov-Hopf bifurcation curve where the stability of an equilibrium changes and a periodic orbit appears. Both of these bifurcations are local bifurcations where the phase portrait changes in a neighbourhood of an equilibrium and they can be investigated analytically.

In contrast, global bifurcations can not be expressed explicitly so we need numerical tools to detect them. We use MatCont which is a Matlab toolbox for the study of parametrized dynamical systems. A global bifurcation that we detect numerically is the homoclinic bifurcation where a periodic orbit touches a saddle and becomes homoclinic, then it disappears. To describe the dynamics of the Hopfield model we put all of the detected local and global bifurcation curves together in the parameter plane and determine the behaviour of the system in each domain. We show some examples for the more interesting phase portraits including periodic and homoclinic orbits.
References


Anonymization Techniques in Social Networks

Yuping Yan, Peter Ligeti

Department of Computer algebra, Eotvos Lorand University
yupingyan@inf.elte.hu, turul@cs.elte.hu

Due to the needs of social analysis, behavior prediction and personal customization, social network data is collected and released in large quantities. In this process, the abuse of private information is a widespread problem. The contradiction between the public service of social networks and the protection of individual private information has become one of the focus issues of current privacy protection. There are mainly three quizzes in cloud computing environment: large data privacy protection, data credibility and access control. K-anonymization [1] as a main data anonymity protection technology, shows its advantages in neighbour attacks and privacy-preserving data publishing (PPDP). However, most of the researches focus on static, one-time release. Thus, these models can not prevent data analysis by collision attacks, and data anonymization in cloud computing environment. In overview, it is a survey paper, and we compare and analyze the differences of various approaches of social network anonymization.

Comparing with the traditional security problems, the challenges of social networks and cloud computing environment lie in the following:

- **Users’ Privacy.** The threats that people are facing is not only the personal privacy leakage, but also the behavior prediction based on social networks. The social network analysis research also shows that the attributes of users can be found through the group characteristics. For example, by analyzing the user’s Instagram information, the user’s consumption habits and preferences can be found.

- **Data Credibility.** A common view is that data shows the fact. However, one of the threats to the credibility of social networks is forge data, thus to create illusions, and induce users. The other is the gradual distortion of the data in the propagation. The introduction of errors in the data collection process will ultimately affects the accuracy of the data analysis results.

- **Big data access control.** The difficulties of access control is to present roles, predict the actual permissions of each role, and meet the requirements of diverse access control requirements.

Based on these problems, researches abroad and overseas conduct a lot of researches. Among these work, the data anonymization is one of the most effective approach. Currently, most of the anonymization researches focus on the relational data. There are some typical privacy persevering models, such as $k$-anonymity[1], $l$-diversity[2], $t$-closeness and $m$-invariance. The characteristic of relational data set is the independence of the records. Thus, the approach for privacy preserving is to break the relationship between individual identifiers and sensitive information. However, in cloud environment, we usually use graphs to present the social network, which contains more information, such as the nodes information, degree information, lines information and so on. Due to the various types of information in social network, there are some typical attacks, such as linking attack, homogeneous attack, skewness attack, similarity attack, value equivalence attack, neighbour attack and background knowledge attack.

Fig.11 shows the current existing anonymization models of social networks. Each model adopts different strategies, they are mainly the following three types: 1) based on generation and limitation; 2) based on cluster; 3) based on randomize modification. We will briefly introduce the concepts of the techniques.

- **Anonymization based on generation and limitation.** This method was first applied to relational data. In relational data, generalization refers to replacing the original data table with a more general and abstract data area or attribute value. Restriction means not to publish or directly delete some tuple records in the data table. It can be divided into

---

13This research has been partially supported by project no. ED_18-1-2019-0030 (Application-specific highly reliable IT solutions) has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme.
Table 1: Anonymization models of social networks

<table>
<thead>
<tr>
<th>Object</th>
<th>Model</th>
<th>Privacy Info</th>
<th>Anti-attack</th>
<th>Anonymity Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undirected graph</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$k$-degree</td>
<td>Identities of nodes or edges</td>
<td>Degree attack</td>
<td>Graph reconstruction, greedy algorithm</td>
</tr>
<tr>
<td></td>
<td>$d$-neighborhood</td>
<td>Neighbor attack</td>
<td></td>
<td>Generation, cluster</td>
</tr>
<tr>
<td></td>
<td>$k$-automorphism</td>
<td>Structure attacks</td>
<td></td>
<td>Block division</td>
</tr>
<tr>
<td></td>
<td>$k$-isomorphism</td>
<td>Isomorphism division</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$k$-symmetry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Directed graph</td>
<td>$k$-anonymity</td>
<td>Sensitive information of nodes and edges</td>
<td>Re-identity attack</td>
<td>Greedy cluster, generation</td>
</tr>
<tr>
<td></td>
<td>$t$-confidence</td>
<td></td>
<td></td>
<td>Randomize edge modification</td>
</tr>
<tr>
<td></td>
<td>$l$-diversity</td>
<td>Homogeneous attack</td>
<td></td>
<td>Greedy algorithm</td>
</tr>
<tr>
<td></td>
<td>Edge weight</td>
<td>Weighted edges attack</td>
<td></td>
<td>Greedy algorithm</td>
</tr>
<tr>
<td>Dynamic network graph</td>
<td>$k$-automorphism</td>
<td>Identities of nodes or edges</td>
<td>Edges attack</td>
<td>Greedy algorithm</td>
</tr>
<tr>
<td></td>
<td>$k$-isomorphism</td>
<td>Structure attacks</td>
<td></td>
<td>Nodes generation</td>
</tr>
<tr>
<td></td>
<td>Link-prediction</td>
<td></td>
<td></td>
<td>Edge prediction</td>
</tr>
</tbody>
</table>

Figure 11: Anonymization models of social networks

global generalization and local generalization. Global generations conduct a generation on all attribute sets. There are classical algorithms like Datafly algorithm, Incognito algorithm, MinGen algorithm and so on. However, these algorithms will lead to a information lose. In order to solve this problem, local generalization was proposed.

- **Anonymization based on aggregation.** The idea of aggregation anonymization is to divide nodes or edges into clusters, and conduct an aggregation in each cluster to achieve nodes or edges anonymization.

- **Anonymization based on randomize modification.** It includes the addition, deletions of nodes, edges, the copy of sub-graph. In order to keep the efficiency of data, it always combines with the greedy algorithm, optimization algorithm. Among these algorithms, reconstruction based on optimization graph is one of the most popular approach.

References

