TURING
CENTENARY CONFERENCE

CiE 2012: How the World Computes

University of Cambridge
18-23 June, 2012

ABSTRACTS
OF INFORMAL PRESENTATIONS

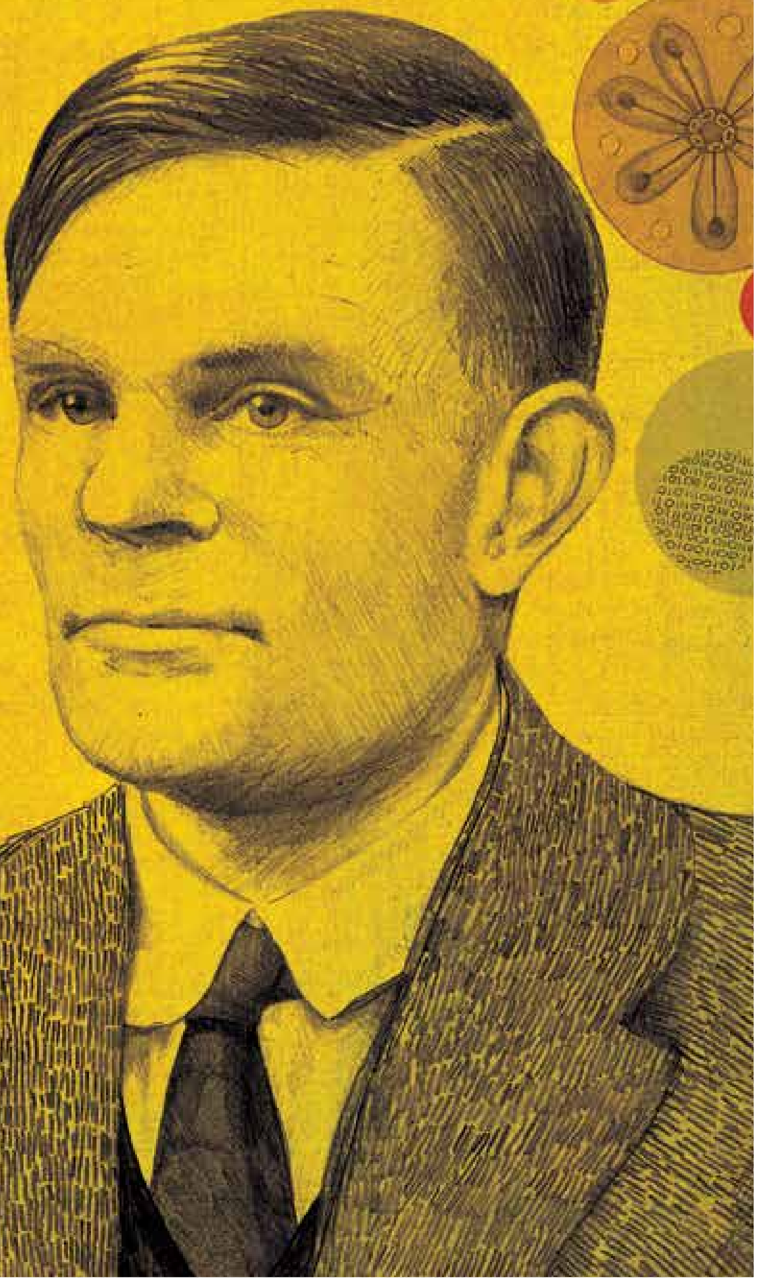


Table of Contents

Author Index

Collective Reasoning under Uncertainty and Inconsistency

Martin Adamčík

School of Mathematics, The University of Manchester, Oxford Road, M13 9PL, Manchester, martin.adamcik@manchester.ac.uk

In practice probabilistic evidence is incomplete, derived from different sources, and as a consequence often contradictory. To built an artificial expert system such as one recognizing diseases from list of symptoms, one classical approach is to define an inference process which picks the "most rational" probabilistic belief function which an agent should have, based solely on the given evidence. For a single incomplete but consistent probabilistic knowledge base satisfying certain reasonable topological criteria, the Maximum Entropy (ME) inference process championed by Jaynes was uniquely characterized by an elegant list of axioms developed by Paris and Vencovská [1]. ME enables a single rational agent to choose an optimal probabilistic belief function.

If however probabilistic evidence is derived from more than one agent, where the evidence from each individual agent is consistent, but the evidence from all agents together is inconsistent, then the question as to how to merge the evidence in such a manner as to be able to choose a single "most rational" probabilistic belief function on the basis of the merged evidence from all agents, has been much less studied from a general theoretical viewpoint.

In this informal presentation we will briefly describe a "social" inference process extending ME to the multi-agent context, called the Social Entropy Process (SEP), based on Kullback-Leibler information distance, and first formulated by Wilmers in [2],[3]. SEP also turns out to be a generalisation of the well-known logarithmic pooling operator for pooling the known probabilistic belief functions of several agents. A new result obtained by us shows that SEP satisfies a natural variant of the important principle of Irrelevant Information which is known to be satisfied by ME. We also indicate how the merging process described by SEP satisfies a suitable interpretation of the set of merging axioms for knowledge bases formulated by Konieczny and Pino Pérez in [4].

Keywords: Uncertain reasoning, discrete probability function, social inference process, maximum entropy, merging operators, Kullback-Leibler.

References:

[1] J. B. Paris and A. Vencovská, (1990) *A Note on the Inevitability of Maximum Entropy*, International Journal of Approximate Reasoning, 4, pp. 183-224

[2] G. Wilmers (2010), *The Social Entropy Process: Axiomatising the Aggregation of Probabilistic Beliefs*, Probability, Uncertainty and Rationality, pp 87-104

[3] G. Wilmers (2011), *Generalising the Maximum Entropy Inference Process to the Aggregation of Probabilistic Beliefs*, http://manchester.academia.edu/GeorgeWilmers/Papers

[4] S. Konieczny and R. Pino Pérez:, (1998) *On the Logic of Merging*, KR 1998: 488-498

Facticity as the amount of self-descriptive information in a data set

Pieter Adriaans

ILLC, IvI, Department of Computer Science
University of Amsterdam,
Science Park 107
1098 XG Amsterdam,
The Netherlands.
pietera@science.uva.nl

Abstract. Ever since the seminal papers by Koppel on *sophistication* some thirty years ago the idea that the "interestingness" of a data set could be measured in terms of its model information has been proposed by a variety of authors. A Problem with all these proposals was that it was not clear how the amount of model information in a data set should be measured exactly. Using the theory of Kolmogorov complexity the notion of *facticity* $\varphi(x)$ of a string is defined as the amount of self-descriptive information it contains. It is proved that (under reasonable assumptions: the existence of an empty machine and the availability of a faithful index) facticity is definite, i.e. random strings have facticity 0 and for compressible strings $0 < \varphi(x) < 1/2|x| + O(1)$. Consequently facticity measures the tension in a data set between structural and ad-hoc information objectively. For binary strings there is a so-called facticity threshold that is dependent on their entropy. Strings with facticty above this threshold have no optimal stochastic model and are essentially computational. The shape of the facticty versus entropy plot coincides with the well-known sawtooth curves observed in complex systems. The notion of factic processes is discussed. This approach overcomes problems with earlier proposals to use two-part code to define the meaningfulness or usefulness of a data set. Based on these results I develop a theory of factic processes. These processes are edge-of-chaos phenomena: they are non-random (maximizing entropy) and non-deterministic (fixed model). Instead they maximize facticity over time and thus never have a fixed predictive model. These findings are consistent with empirical data. Factic processes are abundant in nature (evolution, games, stock markets).

keywords: facticity, factic processes, useful information, sophistication, Kolmogorov complexity, two-part code optimization, nickname problem, sawtooth curves, edge of chaos.

Towards a Type Theory of Predictable Assembly

Assel Altayeva and Iman Poernomo

Department of Informatics, King's College London,
Strand, London WC2R 2LS, UK
iman.poernomo@kcl.ac.uk, a.altayeva.11@ucl.ac.uk

Predictable assembly is an approach to the development of component-based systems in which both functional (correctness) properties and nonfunctional (security, reliability and performance) properties can be predicted, analysed and evaluated by virtue of the compositional nature of the component architecture. The past decade has seen a growth in the design-level treatment of nonfunctional, stochastic properties of components to facilitate the specication, analysis and eventual implementation of code that meets a desirable quality of service. In parallel, the formal methods community has developed techniques for architectural specication of component composition, with a focus on treating functional correctness via compositional analysis. In this work, we bring both these approaches together, developing a type system for component-based architectures. The system is based in the Impredicative Calculus of Constructions. Because this formalism allows dependent product types, by virtue of the Curry-Howard isomorphism, we can dene interface signatures that involve logical propositions, and a semantics of component composition that is correct-by-construction: components can only be composed if their interfaces are logically equivalent.

Worst case analysis of non-local games ⋆

Andris Ambainis, Artūrs Bačkurs, Kaspars Balodis, Agnis Škuškovniks, Juris Smotrovs, and Madars Virza

Faculty of Computing, University of Latvia, Raina bulv. 19, Riga, LV-1586, Latvia, e-mail: andris.ambainis@lu.lv

Non-local games are studied in quantum information because they provide a simple way for proving the difference between the classical world and the quantum world. A non-local game is a cooperative game played by 2 or more players against a referee. The players cannot communicate but may share common random bits or a common quantum state. A referee sends an input x_i to the i^{th} player who then responds by sending an answer a_i to the referee. The players win if the answers a_i satisfy a condition that may depend on the inputs x_i.

Typically, non-local games are studied in a framework where the referee picks the inputs from a known probability distribution. We initiate the study of non-local games in a worst-case scenario when the referee's probability distribution is unknown and study several non-local games in this scenario.

For several commonly studied non-local games, the worst case and the average case game values are the same. We show that this happens for two well-known non-local games: classical CHSH game and Mermin-Ardehali game (an n-player XOR game with the biggest advantage for quantum strategies). We also present several games for which this is not the case.

EQUAL-EQUAL (EE_m) game defined as a two-player XOR game with input data set $\{1,\ldots,m\}$ and the winning condition $(x_1 = x_2) \Leftrightarrow (a_1 = a_2)$. The worst case winning probability for even m is $p_{\text{win}}^{classical} = p_{\text{win}}^{quantum} = \frac{2(m-1)}{3m-4}$ and for odd m is $p_{\text{win}}^{c} = \frac{2m}{3m-1}$ and $\frac{2m}{3m-1} \le p_{\text{win}}^{q} \le \frac{2m(m-1)+1}{(3m-1)(m-1)}$ while the average case winning probability for $m \ge 4$ is $p_{\text{win}}^{c} = p_{\text{win}}^{q} = \frac{m-1}{m}$.

n-party AND game ($nAND$), a symmetric XOR game with binary inputs and the winning condition $(\bigoplus_{i=1}^{n} a_i = \bigwedge_{i=1}^{n} x_i)$. In the average case the winning probability is close to 1: all players output $a_i = 0$. In the worst case scenario $lim_{n\to\infty} p_{win}^{c} = lim_{n\to\infty} p_{win}^{q} = \frac{2}{3}$

We also consider the question: what can the players do if they are not allowed to share common randomness (nor common quantum state)? If the probability distribution on the inputs is fixed, it is equivalent to players sharing common randomness. In the worst-case setting, we get different results. For many games, not allowing shared randomness results in players being unable to win with $p > 1/2$. We show an example where players can still win with a non-trivial probability, even if they are not allowed to share randomness.

⋆ Supported by ESF project 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044, FP7 Marie Curie International Reintegration Grant PIRG02-GA-2007-224886 and FP7 FET-Open project QCS.

Numerical evaluation of the average number of successive guesses

Kerstin Andersson

Karlstad University, Department of Computer Science, SE–65187 Karlstad, Sweden,
Kerstin.Andersson@kau.se

Abstract. This work has been inspired by problems addressed in the field of computer security, where the attacking of, e.g., password systems is an important issue. In [2] Lundin *et al.* discuss measures related to the number of guesses or attemts a supposed attacker needs for revealing information. Similar problems are considered in [1], [3] and [4]. In this presentation numerical approaches are discussed for evaluating the average number of successive guesses required for correctly guessing the value of a string of independent and identically-distributed random variables. The guessing strategy used is guessing strings in decreasing order of probability [1].
The main conclusion is that it is possible to calculate the average number of successive guesses with moderate requirements concerning both memory and CPU time. The exact evaluation demands high storage and CPU time requirements. If n is the size of the alphabet and m is the size of the word the requirements are of $O(n^m)$ and $O(n^{m+1})$, respectively, for storage and CPU time. In a first approximation (using quantification) the high storage demand was removed (to $O(m)$), but the high CPU time demands remained (of $O(mn^m)$). In a second approximation (using random selection) and a third approximation (using a normal distribution) also the high CPU time demands were removed and reduced to $O(m^2)$ for both approximations. However, for all probability distributions the normal distribution is not an accurate approximation.
Considering realistic sizes of alphabets (50) and word lengths (50) both approximations are able to give an estimate of the average number of successive guesses within minutes.

Keywords: guess, randomness, algorithm, complexity

References

1. Arikan, E.: An Inequality on Guessing and its Application to Sequential Decoding. IEEE Transactions on Information Theory, vol. 42, no. 1, 99–105 (1996)
2. Lundin, R., Lindskog, S., Brunström, A., Fischer-Hübner, S.: Using guesswork as a measure for confidentiality of selectivly encrypted messages. Advances in Information Security, Volume 23, Part 4, 173–184 (2006)
3. Malone, D., Sullivan, W. G.: Guesswork and entropy. IEEE Transactions on Information Theory, vol. 50, no. 3, 525–526 (2004)
4. Massey, J. L.: Guessing and entropy. Proc. IEEE Int. Symp. Information Theory, p. 204 (1994)

Turing-degree of first-order logic FOL and reducing FOL to a propositional logic

Hajnal Andréka and István Németi

Alfréd Rényi Institute of Mathematics,
Reáltanoda u. 13-15, Budapest, H-1053 Hungary

Abstract. The set of valid formulas of first-order logic is Turing-equivalent to the equational theory of Boolean algebras with three commuting complemented closure operators. This is a strong improvement on Tarski's result saying that set theory can be formalized in the equational theory of relation algebras and seems to be best possible in the sense that the equational theories of weaker classes of algebras get decidable.

Keywords: Turing degree, Boolean algebras with operators, diagonal-free cylindric algebras, translation functions, first-order logic

By FOL we mean first-order logic with equality and with a decidable set of relation and function symbols, countably many for each finite rank. $\mathsf{BAO3c}$ denotes the class of Boolean algebras (BA) with 3 commuting complemented closure operators c_i ($i < 3$), i.e., c_i are unary functions on the BA satisfying the following equations for all $i, j < 3$: $\mathsf{c}_i\mathsf{c}_j x = \mathsf{c}_j\mathsf{c}_i x$ (commuting), $\mathsf{c}_i - \mathsf{c}_i x = -\mathsf{c}_i x$ (complemented), $x \leq \mathsf{c}_i x = \mathsf{c}_i\mathsf{c}_i x$ (closure), $\mathsf{c}_i(x + y) = \mathsf{c}_i x + \mathsf{c}_i y$ (operators).

Theorem 1. *It is just as hard to decide validity of a FOL-formula as to decide validity of an equation of* $\mathsf{BAO3c}$. *I.e., the set of valid FOL-formulas is Turing-equivalent to the set of valid* $\mathsf{BAO3c}$*-equations.* □

The equational theories of BA's with 2 commuting complemented closure operators, as well as that of BA's with 3 (not necessarily commuting) complemented closure operators are decidable. Hence the number 3 and the adjective "complemented" are important in Thm.1 above. We do not know whether the equational theory of BA's with 3 commuting (not necessarily complemented) closure operators is undecidable or not.

The equational theory of $\mathsf{BAO3c}$ is equivalent to a propositional modal logic with 3 commuting S5 modalities, denoted as [S5,S5,S5] in the literature. Thm.1 is a corollary of Thm.2 below. Let $\vdash$ denote the provability relation of the propositional multi-modal logic [S5,S5,S5].

Theorem 2. *FOL can be reduced to propositional modal logic [S5,S5,S5], i.e., there is a computable Boolean-preserving translation function* tr *from FOL to [S5,S5,S5], which means that (i)-(ii) below hold for all sets* $Th \cup \{\varphi, \psi\}$ *of FOL sentences.*

(i) $Th \models \varphi$ *if and only if* $\mathsf{tr}(Th) \vdash \mathsf{tr}(\varphi)$
(ii) $\vdash \mathsf{tr}(\varphi \vee \psi) \leftrightarrow (\mathsf{tr}(\varphi) \vee \mathsf{tr}(\psi))$ *and* $\vdash \mathsf{tr}(\varphi \to \psi) \to (\mathsf{tr}(\varphi) \to \mathsf{tr}(\psi))$. □

Computing without a computer: the analytical calculus by means of formalizing classical computer operations

Vladimir Aristov[1] and Andrey Stroganov[2]

[1] Dorodnicyn Computing Centre of Russian Academy of Sciences,
[2] Moscow State Technical University of Radioengineering, Electronics and Automation

Keywords: theoretical model of computer; method of computer analogy; digit shifting; nonlinear differential equations

Abstract. The Turing machine is an example of a theoretical model of a classical computer, and it was the logical basis of constructing electronic computers. The recent general tendency shows that most researches aim on developing new computing devices (e.g. quantum computers) and analytical procedures for improving computer software [1]. We propose an alternative theoretical model of a computer in the framework of which we introduce a new calculation technique based on the properties of the digital computer. We refer to this approach as the method of computer analogy. This model utilizes the following aspects of classical computer: 1) numbers represented as segments of a power series; 2) a procedure of digit shifting. The method of computer analogy can be used for obtaining the explicit form of the solutions for problems which can be solved numerically using iterative or finite difference schemes, in particular for solving nonlinear differential equations [2]. The value of the unknown function is represented as a segment of the power series in powers of the step τ of the independent variable. The less significant digits exhibit stochastic behaviour. This allows us to use probabilistic methods to predict changes in the internal state of the model of a computer under consideration. In terms of numerical solution, this leads to excluding intermediate computations in the reccurrent formula. This method does not only reduce the number of the arithmetical operations in calculations, but it can also provide a solution in the explicit form (as a classical computer provides a solution in the numerical form, after executing many intermediate and "hidden" operations). The final analytical solution is treated as the limit when τ tends to zero.

References

1. Davenport, J.H., Siret, Y., Tournier, E.: Computer algebra: systems and algorithms for algebraic computation. Academic Press, San Diego, 1993.
2. Aristov, V.V., Stroganov, A.V.: A method of formalizing computer operations for solving nonlinear differential equations. Applied Mathematics and Computation. Vol. 218 (2012) 8083-8089, doi:10.1016/j.amc.2011.09.029.

On Models of the Modal Logic GL

İlayda Ateş[1] and Çiğdem Gencer[2]

Department of Mathematics and Computer Science,
İstanbul Kültür University, Ataköy Campus,
Bakirköy, 34156, İstanbul/Turkey,
i.ates@iku.edu.tr[1], *c.gencer@iku.edu.tr*[2]

Keywords: Canonical Model, finite Henkin method, filtration method

In this survey talk we show that the models of **GL** that is obtained by two different methods are isomorphic. For that purpose we use modal completeness and the canonical model of **GL**.

Definition 1. (i)The modal logic **GL** is axiomatized by adding the scheme L to the logic **K** $L : \Box(\Box\varphi \to \varphi) \to \Box\varphi$.
(ii) A Kripke frame for **GL** is a pair $\langle W, \mathcal{R}\rangle$ with $\mathcal{R}$ a transitive relation such that the converse of $\mathcal{R}$ is well-founded.
(iii) A Kripke model for **GL** is a triple $\langle W, \mathcal{R}, V\rangle$ with $\langle W, \mathcal{R}\rangle$ a Kripke frame for **GL** together with a valuation $\models$ between worlds and propositional variables.

It is known that **GL** is decidable and complete with respect to the finite, irreflexive (and therefore conversely well-founded), transitive frames. Completeness theorems are model existence theorems. The finite Henkin method and the filtration method are two model building techniques. In both the points of the underlying frames are related to maximal consistent sets, and the relations and valuation are defined in terms of membership in such sets. An *adequate* set is closed under subformulas and contains the negation of each formula which is not a negation.

Definition 2. The model obtained by finite Henkin method for the modal logic **GL** $\mathcal{M}^{\Phi}_{\mathbf{GL}}$ w.r.t. the finite adequate set Φ is the model $\langle W^{\Phi}_{\mathbf{GL}}, \mathcal{R}^{\Phi}_{\mathbf{GL}}, V^{\Phi}_{\mathbf{GL}}\rangle$ with
(i) $W^{\Phi}_{\mathbf{GL}} = \{\Gamma \mid \Gamma$ is maximal **GL**-consistent in $\Phi\}$,
(ii) $\mathcal{R}^{\Phi}_{\mathbf{GL}} = \{\langle \Gamma, \Gamma'\rangle \mid$ for all φ, if $\Box\varphi \in \Gamma$, then $\varphi \in \Gamma'$ and $\Box\varphi \in \Gamma'$, and for at least one $\Box\varphi \in \Gamma'$, not $\Box\varphi \in \Gamma\}$.
(iii) $V^{\Phi}_{\mathbf{GL}}(p) = \{\Gamma \mid p \in \Gamma\}$.

Now, we give the main result.

Theorem 1. The models obtained for the modal logic **GL** by the finite Henkin method are isomorphic to the ones obtained by a filtration of the canonical model for **GL** if both are defined w.r.t. the same finite adequate set Φ.

Sketch of the Proof. We construct an isomorphism between the models obtained for the modal logic **GL** by the two different methods. Let $\mathcal{M}_{\mathbf{GL}}$ be the canonical model of **GL**. We define a filtration with regard to Φ as follows. First we take a unique final element $F(|\Gamma|)$ in each equivalence class $|\Gamma|$ w.r.t. Φ where Γ' is final in $|\Gamma|$ if for no $\Gamma'' \in |\Gamma|$, $\Gamma'\mathcal{R}\Gamma''$. Then we take $|\Gamma|\mathcal{R}^f|\Gamma'|$ if for all $\Box\varphi \in \Phi$, if $\mathcal{M}_{\mathbf{GL}}, F(|\Gamma|) \models \Box\varphi$, then $\mathcal{M}_{\mathbf{GL}}, F(|\Gamma'|) \models \Box\varphi \wedge \varphi$ and for some $\Box\psi \in \Phi$, $\mathcal{M}_{\mathbf{GL}}, F(|\Gamma'|) \models \Box\psi$ and $\mathcal{M}_{\mathbf{GL}}, F(|\Gamma|) \models \neg\Box\psi\}$. The function $h(|\Gamma|) = F(|\Gamma|) \cap \Phi$ is an isomorphism.

1-Genericity and the Π^0_2 Enumeration Degrees.

Liliana Badillo and Charles M. Harris

Department of Mathematics,
Leeds, UK.

lilianabsmath@gmail.com
harris.charles@gmail.com

Abstract. Using results from the local structure of the enumeration degrees we show the existence of prime ideals of Π^0_2 enumeration degrees. We begin by showing that there exists a 1-generic enumeration degree $\mathbf{0}_e < \boldsymbol{a} < \mathbf{0}'_e$ which is noncuppable—and so properly downwards Σ^0_2—and low$_2$. The notion of *enumeration* 1-*genericity* appropriate to positive reducibilities is introduced and a set A is defined to be *symmetric enumeration* 1-*generic* if both A and $\overline{A}$ are enumeration 1-generic. We show that, if a set is 1-generic then it is symmetric enumeration 1-generic, and we prove that for any Π^0_2 enumeration 1-generic set B the class $\{\, X \mid \leq_e B \,\}$ is uniform Π^0_2. Thus, picking 1-generic $A \in \boldsymbol{a}$ (from above) and defining $\boldsymbol{b} = deg(\overline{A})$ it follows that every $\boldsymbol{x} \leq \boldsymbol{b}$ only contains Π^0_2 sets. Since $\boldsymbol{a}$ is properly Σ^0_2 we deduce that $\boldsymbol{b}$ contains no Δ^0_2 sets and so is itself properly Π^0_2.

Complexity of complexity and maximal plain versus prefix-free Kolmogorov complexity

Bruno Bauwens ⋆

Instituto de Telecomunicações Faculdade de Ciência da Universidade do Porto.
Brbauwens *at* gmail

Peter Gacs showed [2] that for every n there exists a bit string x of length n whose plain complexity $C(x)$ has almost maximal conditional complexity relative to x, i.e., $C(C(x)|x) \geq \log n - \log\log n - O(1)$. Following Elena Kalinina [3], we provide a game-theoretic proof of this result; modifying her argument, we get a better (and tight) bound $\log n - O(1)$. We also show the same bound for prefix-free complexity.

As an intermezzo we state symmetry of information for plain complexity [1] as:

$$C(a,b) = K(a|C(a,b)) + C(b|a, C(a,b)),$$

which has two interesting known corollaries: Levin's formula $C(a) = K(a|\,C(a))$ (taking $b = C(a)$), and every infinitely often C-random real is 2-random.

Finally, we provide a short proof for Solovay's result [4] (a bit improved) stating that for some strings plain complexity can be maximal but prefix-free complexity not. More precise: infinitely many strings x have $C(x) = |x| - O(1)$ and $K(x) = |x| + K(|x|) - \log\log|x| + O(1)$. The proof only uses symmetry of information of prefix-free complexity, and Levin's and Gacs' results (see above).

References

1. B.Bauwens, A.Shen. An additivity theorem for plain complexity. Theory of Comput. Sys., accepted for publication.
2. P. Gács. On the symmetry of algorithmic information. Soviet Math. Dokl., 15(5):1477–1480 (1974).
3. E. Kalinina. Some applications of the method of games in Kolmogorov complexity. Master thesis. Moscow State University, 2011.
4. R.Solovay, Draft of a paper (or series of papers) on Chaitin's work. Unpublished notes, 215 pages, (1975).

⋆ Supported by the Portuguese science foundation FCT (SFRH/BPD/75129/2010), and is also partially supported by the project CSI^2 (PTDC/EIAC/099951/2008). The author is grateful to Elena Kalinina and (Nikolay) Kolia Vereshchagin for giving the text [3]. The author is also grateful to (Alexander) Sasha Shen for his very generous help, especially to reformulate the original game and strategy in a much cleaner way.

When is the denial inequality an apartness relation?

Josef Berger

Ernst Moritz Arndt Universität Greifswald
Institut für Mathematik und Informatik
Walther-Rathenau-Straße 47
17487 Greifswald, Germany
bergerj@uni-greifswald.de

A binary relation $\bowtie$ on a set X is called *apartness relation* if for all $x, y, z \in X$ the following holds:

a) $\neg(x \bowtie x)$
b) $x \bowtie y \Rightarrow y \bowtie x$
c) $x \bowtie y \Rightarrow x \bowtie z \vee y \bowtie z$

We work within Bishop's constructive mathematics. A real number is a sequence (x_n) of rationals such that

$$\forall m, n \left(|x_m - x_n| \leq m^{-1} + n^{-1}\right).$$

The real number 0 is represented by the sequence (z_n) with $z_n = 0$ for all n. For two reals x, y we define *equality* by

$$x = y \overset{def}{\Leftrightarrow} \forall n \left(|x_n - y_n| \leq 2n^{-1}\right),$$

which is equivalent to

$$\forall k \exists n_0 \forall n \geq n_0 \left(|x_n - y_n| \leq k^{-1}\right).$$

The negation of equality, the so-called *denial inequality* is given by

$$x \neq y \overset{def}{\Leftrightarrow} \neg (x = y)$$

and clearly fulfills a) and b). We present a logical axiom which is equivalent to the statement: 'The denial inequality is an apartness relation.'

A formula Φ is called a Π^0_1*-formula* if there exists a binary sequence α such that

$$\Phi \Leftrightarrow \forall n \, (\alpha n = 0).$$

Consider the following axioms:

DA The denial inequality is an apartness relation.

Π^0_1**-**DML For all Π^0_1-formulas Φ and Ψ, $\neg(\Phi \wedge \Psi) \Rightarrow \neg\Phi \vee \neg\Psi$.

The latter is an instance of the the *De Morgan law*.

Lemma The axioms Π^0_1-DML and DA are equivalent.

Logical Agency: consulting instead of collecting

Carolina Blasio[1] and João Marcos[2*]

[1] IFCH / UNICAMP, Campinas-SP, Brazil
[2] LoLITA and DIMAp, UFRN, Natal-RN, Brazil

A traditional and much explored model of information processing envisions computers as machines collecting reports presented by a variety of (trustworthy) sources that feed the machines with data on which the latter are to base their inferences. While it is very natural for this model to give support to inconsistencies and undeterminedness phenomena that often underlie multi-agent systems, it is not obvious how the model should accommodate consultations with full-blown agents, who come equipped with their own reasoning apparatus. The present contribution claims —and illustrates the claim from a logical viewpoint— that a more natural model for explicating the behavior of societies of agents, and at the same time for taking their inferential capabilities into account, is one that treats agents not as sources of unanalysed pieces of information being either asserted or denied, but as judgmental beings who are consulted upon their inclinations either to accept or to reject given pieces of information. Some setups of the new model are then explored at the level of combinations of agents.

Keywords. Reasoning about uncertainty, information sources, combinations of agents

* Corresponding author.

Virtual Worlds as Portals for Information Discovery

Shannon Bohle, MLIS, FRAS

University of Cambridge

Virtual worlds provide immersive environments ideal for education, training and simulation, artificial intelligence research, engineering and robotic modeling. Through implementation of increasingly realistic graphics, artificial intelligence, improved human-computer interfaces, and the mantra of “gamification”, virtual worlds have become explorative and interactive settings to research information seeking patterns and behaviors. This presentation will discuss what I have learned over the past three years about how organisations, such as NASA, the US Army, and the IEEE use virtual worlds for serious games, looking specifically at points of intersection I have had with these groups. Firstly, I will discuss the information seeking behavior of avatar-based library patrons visiting the Neil A. Armstrong Library and Archives (2008-2011), where I volunteered as the founder and director of the first virtual world digital library or archive recognized by the Library of Congress. A machinima video featuring a patron’s astrophysics question in the library was shown at the Nobel Museum in Sweden. Secondly, virtual worlds are computerized environments where people and intelligent agents conceptually have a level playing field in which to interact. Intelligent agents hold the potential to automate multiple actor scenarios otherwise conducted by human controllers. To attract new ideas in virtual worlds and artificial intelligence, the White House advertised an international competition called the Federal Virtual Worlds Challenge. In 2011, 2nd place in the category “AI Concept Exploration” was awarded to “Curiosity AI”, my virtual 3D simulation of Mars, its rovers, robots, and satellites. LIS literature played a role in my design of AI functions. The question-answer manner in which librarians interact with patrons (a “reference interview”) is similar to the structure of a Turing test, relying upon discourse satisfaction as the primary success measure. Curiosity AI took Turing tests farther by employing a humanoid embodied agent that communicated with facial expressions, gestures, and spatial movement, as well as controlling other agents using text-based chat. Additional AI areas covered included: autonomous movement and obstacle avoidance, expert systems, rovers, swarms, and in-situ data analysis. Also mentioned will be Project MOSES (US Army), CPMI (DISA affiliated), as well as a virtual world study group for the “Introduction to AI” course taught by Peter Norvig and Sebastian Thrun. In short, the key to improving information discovery is changing how we interact with digital information and the information seeking tools that mine information for us, and cutting edge virtual worlds researchers are at the forefront of the future of AI and HCI. *Disclaimer*: All opinions expressed are my own and are not representative of any organization named.

On Turing Scores

R.J. Booth /clamjet@hotmail.co.uk/

GCHQ (UK)

1 Abstract

This work comes from a declassified version of a 1994 paper internal to GCHQ. It considers the problem of how best to combine "weights of evidence" when some of this weight – let us call it X – is not in the form of a log Bayes factor (LBF), this being the standard yardstick shown by the Neyman-Pearson lemma to be optimal for simple null and alternative hypotheses.

In order to use the information in X, we propose a transformation $\widehat{S}(X)$, subject to one free parameter, which approximately satisfies the so-called "Turing Relations" (see Good: Biometrika 66, 393-396, Section 7), though they are not so named there). The Turing relations record the fact that if Y is an LBF for an alternative H_1 against a null H_0, which is normally distributed under both hypotheses with means $\mu_i = \mathbf{E}[Y|H_i]$ and variances $\sigma_i^2 = \mathrm{Var}[Y|H_i]$, then

$$\mu_0 = -\mu_1$$
$$\sigma_0^2 = \sigma_1^2 = 2\mu_1$$

The Turing relations are important because they hold approximately even in the non-normal case *if* H_1 is "close" to H_0. For example, with binomial distributions $H_1 : K \sim B(1, \frac{1}{2}(1-b))$ versus $H_0 : K \sim B(1, \frac{1}{2})$ and small b, the LBF $\log(1 + (-1)^K b)$ satisfies the Turing relations with relative error $O(b^2)$.

Therefore, making a statistic approximately satisfy the Turing relations is likely to make it approximate the true LBF. We coin the verb "to ture" to describe this process or its result, and hence we may say "$\widehat{S}$ was derived by turing X".

The particular method of turing which we propose comes from study of LBFs which arise from normal distributions, since the Turing relations can be exact for these. In the general normal case the LBF is quadratic, but the quadratic term disappears when the "equal variance" relation holds, so we drop it. This gives us the equations

$$\hat{\mu} = (1-\lambda)\mu_0 + \lambda\mu_1$$
$$\widehat{S}(X) = \left(\frac{\hat{\mu}-\mu_0}{\sigma_0^2} + \frac{\mu_1-\hat{\mu}}{\sigma_1^2}\right)X - \frac{\hat{\mu}^2-\mu_0^2}{2\sigma_0^2} - \frac{\mu_1^2-\hat{\mu}^2}{2\sigma_1^2}$$

for a family of tured scores with free parameter λ. In the case where X is drawn from a binomial distribution $B(n, \frac{1}{2}(1-b))$ with small b, we show that $\lambda = \frac{2}{3}$ is optimal for matching $\widehat{S}(X)$ to the true LBF $S(X)$.

We then study the turing of $H_1 : X \sim B(100, 0.55)^k$ v $H_0 : X \sim B(100, 0.5)^k$ for each $k \in \{0.5, 1, 2\}$. Further, following subsequent work by Bender & Kochman and by Ostapenko, who proposed alternative methods of turing, we show that for this small set of examples our method generally outperforms the others – provided we set $\lambda = \frac{1}{2}$ instead of the previously suggested $\frac{2}{3}$.

The author was privileged to start his GCHQ career working alongside Joan Murray, who was at one time Alan Turing's fiancee, and during the presentation he will reminisce about Joan and life at GCHQ at that time.

NAÏVE UNREVEALABLY SECRET COMMUNICATION

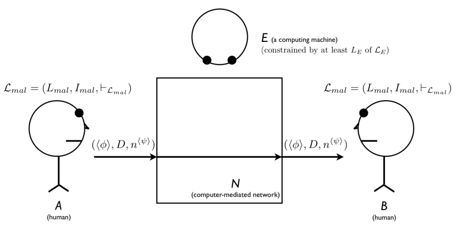
Selmer Bringsjord

Rensselaer Polytechnic Institute (RPI)

A wishes to communicate to B the proposition ϕ by sending information through a computer-mediated network. The communication is naïve because A and B forego use of cryptographic algorithms. A third agent E eavesdrops on what A sends to B. A communicates ϕ to B in **unrevealably secret** fashion iff E, despite seeing what A sends B, cannot understand that A communicates ϕ to B. E is itself understood to be constrained by elements of a given logic $\mathcal{L}_E$.

We distinguish between **cognitively immature** (c.i.) vs. **cognitively mature** (c.m.) logics by (among other things) taking the latter to be those needed to model cognitive performance shown by psychologists to mark a progression to abstract reasoning that calls for intensional operators. It has been found for example that in false-belief tasks, the drosophila of logico-computational theories of mind, introduced by Wimmer & Perner (1983), subjects who pass are able to reason about the beliefs, desires, and intentions of other people.

We give a concrete example (the G$\#^{\geq 4}$ Protocol; pictured below) of secret communication between A' and B' that is provably unrevealably secret relative to an eavesdropper E'. Cognitively immature logics are inadequate against the G$\#^{\geq 4}$ Protocol because such logics are all exclusively linguistic in nature. The formal framework for diagrammatic reasoning known as 'Vivid' (Arkoudas & Bringsjord 2009) offers some hope, but we need machine eavesdroppers based on logics that enable them to not only represent visual content, but to: represent baseline phenomena for which full first-order logic was invented; capture a full range of intensional operators, including those central to modeling minds of malicious agents (e.g., **believes**, **knows**, **intends**); and provide concomitant automated reasoning technology.



References

Arkoudas, K. & Bringsjord, S. (2009), 'Vivid: An AI Framework for Heterogeneous Problem Solving', *Artificial Intelligence* **173**(15), 1367–1405.

Wimmer, H. & Perner, J. (1983), 'Beliefs about Beliefs: Representation and Constraining Function of Wrong Beliefs in Young Children's Understanding of Deception', *Cognition* **13**, 103–128.

Average Case Complexity of ε-NFA's*

Sabine Broda, António Machiavelo, Nelma Moreira, Rogério Reis**
sbb@dcc.fc.up.pt,ajmachia@fc.up.pt,{nam,rvr}@dcc.fc.up.pt

Centro de Matemática da Universidade do Porto
Rua do Campo Alegre 687, 4169-007 Porto, Portugal

The worst-case complexity of the conversions between different representations of regular languages is well studied. However, for practical purposes, the average-case complexity of such conversions is much more relevant than its worst-case complexity, which is often due to some particular and rarely occuring cases. Still, the average-case analysis is, in general, a difficult task. One approach is to consider uniform random generators and to perform statistically significant experiments. Another approach is the use of asymptotic methods.

In this presentation, we discuss asymptotic average-case results on the size of non-deterministic finite automata obtained from regular expressions, using the symbolic method and the framework of analytic combinatorics [1]. The symbolic method allows the construction of a combinatorial class C in terms of simpler ones, $\mathsf{B}_1,\ldots,\mathsf{B}_n$, by means of specific operations, and such that the generating function $C(z)$ of C is a function of the generating functions $B_i(z)$ of B_i, for $1 \leq i \leq n$. The methodology of analytic combinatorics used can be summarized in the following steps:

a) consider an unambiguous context-free grammar for regular expressions;
b) for each measure of a given NFA construction (number of states, number of transitions, etc.) obtain a generating function;
c) see generating functions as analytic complex functions, and study their behaviour around their dominant singularities to obtain an asymptotic approximation of their coefficients.

Among other results, we show, for instance, that the size (number of states plus transitions) of the Thompson automaton, is on average 3.25 times the size of the original regular expression. We illustrate how to use a computer algebra system, in our case the Maple system, to carry out most of the symbolic computations involved in such a study.

References

1. Sedgewick, R., Flajolet, P.: Analysis of Algorithms. Addison Wesley (1996)

* This work was partially funded by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT under projects PEst-C/MAT/UI0144/2011 and CANTE-PTDC/EIA-CCO/101904/2008.

** Corresponding author.

Cavity motion affects entanglement

David Edward Bruschi

School of Mathematical Sciences, University of Nottingham

We propose a scheme to analyze how relativistic motion of cavities affects entanglement between modes of bosonic or fermionic quantum fields contained within. A cavity is modeled by suitable boundary conditions depending on the quantum field being bosonic or fermionic. We can address motion of a cavity when the walls undergo (different) uniform acceleration or coast at constant speed. When there has been at least one period of acceleration, all field modes inside of a cavity become mixed and the transformation between the initial region modes and final region modes is called Bogoliubov transformation. We work in a perturbative regime where the parameter $h \ll 1$ is the product of the proper acceleration of the center of the cavity and the length of the box.

We consider scenarios with two cavities, one of which undergoes some "general" trajectory; in this case we analyze the effects of motion on the entanglement initially present between modes in the two boxes and find that in general entanglement is degraded [1,2]. We find that the effective mass, which accounts for transverse momenta and the mass of the field, increases the degradation effect in a dramatic way. All these effects occur to second order in h. In this case, entanglement degradation between modes in two different $\sim 10m$ cavities can become observable for massless tranverse photons of optical wavelength at accelerations of $\sim 1g$. Our results indicate that gravity might affect quantum information tasks.

We also consider scenarios where one cavity follows some general trajectory and in this case we analyze the entanglement between different modes of the field contained inside: we find that entanglement is created [3]. Surprisingly, we find that there is entanglement created at order h and we also find that, given special trajectories, entanglement generation can be enhanced by repeating any travel scenario [4].
Motion of one cavity can produce linear increase of entanglement under suitable conditions. In particular, if any travel scenario is repeated, total proper times of travel can be chosen to have linear increase with the number of repetitions. The final state is a two mode squeezed state. We aim at understanding which types of quantum gates can be performed by just "shaking" the cavity. Last, we notice that initial single mode squeezing can enhance the final results.

References

1. D. E. Bruschi, J. Louko and I. Fuentes: **Voyage to Alpha Centauri: Entanglement degradation of cavity modes due to motion**, Phys. Rev. D 85 (R):061701, (2012)
2. N. Friis, A. Lee, D. E. Bruschi, J. Louko: **Kinematic entanglement degradation of fermionic cavity modes**, 2011, Phys. Rev. D 85:025012, (2012)
3. N. Friis, D. E. Bruschi, J. Louko and I. Fuentes: **Motion generates entanglement**, Phys. Rev. D. 85 (R):081701 (2012)
4. D. E. Bruschi, A. Dragan, A. R. Lee, I. Fuentes and J. Louko: **Motion-generated entanglement resonance**, arXiv:1201.0663 (2012)

Heuristics and Intelligence: Turing's Vision

Cameron Buckner

University of Houston, USA and Ruhr-University Bochum, Germany
cjbuckner@uh.edu
http://cameronbuckner.net/professional/

Towards the end of his life, Alan Turing made several comments that reflected a new attitude towards the relationship between computation and intelligence. Turing had been considering the limits of computation and the ability of the human mind to solve apparently incomputable problems through intuition and "bursts of insight". In earlier works, he often appealed to a conceptual tool called an 'oracle,' which could compute an incomputable step "as if by magic." As one version of the story goes, Turing became increasingly dissatisfied with oracles as too mysterious to be useful in the study of creativity and insight. However instead of capitulating to these challenges, Turing renewed his renewed his conviction that the processes performed by the human brain must be computable and began to focus on methods of reducing incomputable problems to more manageable ones by abandoning what he called the "pretense of infallibility." Tragically, Turing died before he was able to pursue this research program to fruition.

Many may suppose that these insights have adequately been pursued by Turing's intellectual descendants in the field of heuristic search. Heuristic search algorithms speed up search by ordering search space exploration using heuristic functions. Heuristic functions estimate the cost or distance between any node and a solution (e.g. the Manhattan Distance function), often by exploiting domain-specific heuristic knowledge derived from a simplified domain model (sometimes alternatively called heuristic information). This approach to heuristics reached a zenith in the 1984 work by Judea Pearl, *Heuristics*[1]. Pearl notes that search heuristics order the search space by appealing to simplified domain models. Heuristic search orderings, however, are typically considered admissible only if they are still guaranteed to find a solution, thus not exploring Turing's idea that "if a machine is expected to be infallible, it cannot also be intelligent"[2]. A persistant challenge for heuristic search has been to automate the generation of simplified domain models. In this talk, I question whether Turing's insights have been adequately explored in the field of heuristic search, arguing that attention to the ways that humans and animals form simplified domain models can provide new directions for research in the area of heuristic search.

References

1. Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley (1984).
2. Turing, A.: Lecture to the London Mathematical Society. Typescript in the Turing Archive, King's College, Cambridge (1947).

Notes on Spiking Neural P systems and Petri nets

Francis George C. Cabarle, Henry N. Adorna

Algorithms & Complexity Lab
Department of Computer Science
University of the Philippines Diliman
Diliman 1101 Quezon City, Philippines
E-mail: fccabarle@up.edu.ph, ha@dcs.upd.edu.ph

Abstract. Spiking Neural P systems (in short, SNP systems) are biologically inspired computing devices whereas Peri nets are graphical and mathematical modeling and analysis tools. It has been pointed out by Păun in 2007, about a year after SNP systems were introduced, that the two share a common 'look and feel'. Several recent work have investigated mostly the translation of SNP systems to Petri nets. In this work we further investigate Petri nets and their properties, translating them into SNP systems, while maintaining Petri net semantics. In particular we observe how behaviors and structures fundamental to Petri net theory such as (among others) parallel or decision routing, deadlock, free-choice, and well-handledness translate into SNP systems. The insights from our investigations provide precise additional details and ideas on how closely associated SNP systems and Petri nets are, allowing for further future results for both.

Keywords: Natural Computing, Membrane Computing, Spiking Neural P systems, Petri nets

Interweavings of Alan Turing's Mathematics and Sociology of Knowledge*

Isabel Cafezeiro[1] and Ivan da Costa Marques[2]

[1] Universidade Federal Fluminense, Rua Passo da Pátria 156, Niterói, RJ
isabel@dcc.ic.uff.br,
[2] HCTE / Universidade Federal do Rio de Janeiro,
Av Atlântica 822 apt 402, Leme, RJ
imarques@ufrj.br

We start from the analysis of how Alan Turing proceeded to build the notion of computability in his famous 1936 text 'On computable numbers, with an application to the Entscheidungsproblem'. Turing's stepwise construction starts from the materialities of a human computer, that is, the human body, pencil and inscriptions on paper. He then proceeds by justifying each step of abstraction based on meticulous observations about that materiality. We identify a strong conformity of his way of doing math to methodologies developed by anthropologists and propose a 'translation' of an anthropological concept, that is, an ethnography of mathematical knowledge. Here 'translation' indicates not only a resemblance but also an inevitable difference that results from transposition and use of a concept from an area onto another (*traduttori traditori*). Turing did not configure collective entities, as is now done in the anthropology of knowledge (Bruno Latour). We claim, however, that in aiming to characterize the computable, Turing acted as an ethnographer. Borrowing the words of one of the founders of ethnography, Franz Boas, he acted 'considering every phenomenon as worthy of being studied for its own sake' (Boas, The Study of Geography,Science, feb, 1887,p.210), working on the elicitation of the links between the abstract thinking and the world where one lives. We demonstrate how Turing's way of doing mathematics was one that constructs mathematical knowledge by evading a definite separation between matter and form; in this way, making the world and language come together. Following the same line of reasoning, the abstract and the concrete, the deduction and the induction, the technical and the social as well as the objective and the subjective are unthinkable as pure entities. By considering the controversies and discussions from the mid-nineteenth century until now, we can indicate local (social) elements that necessarily participate in what is usually considered 'technical content' or 'objectivity'. While Alan Turing was a precursor of what today might be said to be an 'anthropological approach to mathematical culture', unveiling and reviving approaches that enable the axis of authority for mathematics, logic and computing to be shifted, he also opened different paths for the construction of a variety of mathematical knowledge as well.

* We thank Fundação Euclides da Cunha for the partial funding of this work and Kay Shipton for her kind assistance in reviewing the English.

The Legacy of Turing Reducibility

William C. Calhoun

Department of Mathematics, Computer Science and Statistics
Bloomsburg University, Bloomsburg, PA 17815, USA
wcalhoun@bloomu.edu

Abstract. Soare [4] has written an extensive paper on oracle Turing machines, tracing their origins from a brief passage in Turing's Ph.D. thesis through their development by Post and Kleene to the central role they play in computability theory. Oracle Turing machines provide a natural way to define relative computability, the key concept in computability theory. Here we consider the continuing importance of oracles in areas of research that have grown out of computability theory such as computational complexity and algorithmic randomness. The use of oracles in computational complexity is quite natural in analogy to computability theory. Just as oracle Turing machines can be used to define Turing reducibility, polynomial-time oracle machines can be used to define polynomial-time Turing reducibility. A striking use of oracles in computational complexity is the Baker, Gill, Solovay [1] result that the truth of $P^A = NP^A$ depends on the oracle A. The usefulness of oracles in algorithmic randomness is not so obvious. It seems particularly surprising that the definition A *is low-for-random* coincides with the definition A *is K-trivial*. The first indicates that A is weak as an oracle, the second is defined without reference to oracles. Neverless, the remarkable equivalence of these definitions was shown by Nies [3]. We will also discuss some new results concerning the Turing degrees of K_m-trivial sets [2] and *almost-K-trivial* sets (a set A such that $K(A \upharpoonright n) \leq^+ aK(n)$ for some real number a).

Keywords: Turing reducibility; oracles; computational complexity; algorithmic randomness

References

1. Baker, T., Gill, J., Solovay, R.: Relativations of the P=? NP question, SIAM Journal on Computing 4, 431-442 (1975).
2. Calhoun, W.C.: Triviality and minimality in the degrees of monotone complexity, Journal of Logic and Computation 22, 197-206 (2012).
3. Nies, André: Lowness properties and randomness, Advances in Mathematics 197, 274-305 (2005).
4. Soare, Robert I.: Turing oracle machines, online computing, and three displacements in computability theory, Annals of Pure and Applied Logic 160, 369-399 (2009).

On the distribution of recognizable reals

Merlin Carl

Fachbereich für Mathematik und Statistik der Universität Konstanz

Abstract. Infinite Time Register Machines ($ITRM$s) are a machine model for infinitary computations generalizing classical register machines by allowing infinite ordinals as running times, established e.g. in [1], [2], [5], [6]. Let us call a real $x \subset \omega$ $ITRM$-computable iff there is an $ITRM$-program P that stops with output 1 on input $i \in x$ and with output 0 otherwise. Furthermore, let us call $r \subset \omega$ recognizable iff there is an $ITRM$-program P^x using a real oracle x that stops on the empty input with output 1 iff $x = r$, and otherwise with output 0. In [2] it is shown that, similar to Infinite Time Turing Machines (see [3]), ITRMs satisfy a Lost Melody Theorem: Namely there are $ITRM$-incomputable, recognizable reals. The $ITRM$-computable reals are proved in [5] to coincide with the reals in L_μ, where μ is the limit of the first ω many admissible ordinals. In particular, the $ITRM$-computable ordinals form an initial segment of the constructible ordinals in the constructible well-ordering $<_L$ of the constructible hierarchy. Considering the class $RECOG$ of recognizable reals, we show that $RECOG$ has gaps in $<_L$, i.e. there are $r_1, r_2, r_3 \in L \cap^\omega 2$ such that $r_1 <_L r_2 <_L r_3$ with $r_1, r_3 \in RECOG$ and $r_2 \notin RECOG$. Furthermore, we show that the ordertype of such a gap (with respect to $<_L$) can be larger than any ordinal below the limit of the first ω many admissibles.

References

1. M. Carl. Alternative Finestructural and Computational Approaches to Constructibility. Phd Thesis. (2010)
2. M. Carl, T. Fischbach, P. Koepke, R. Miller, M. Nasfi, G. Weckbecker. The basic theory of infinite time register machines. Archive for Mathematical Logic 49 (2010) 2, $249 - 273$
3. J.D. Hamkins, A. Lewis. Infinite Time Turing Machines. J. Symbolic Logic 65(2), $567 - 604$ (2000)
4. P. Koepke. Infinite time register machines. In A. Beckmann et al. (eds.) Logical approaches to computational barriers, LNCS, vol. 3988, pp $257 - 266$. Springer, Heidelberg (2006)
5. P. Koepke, R. Miller. An enhanced theory of infinite time register machines. In Arnold Beckmann et al. (eds) Logic and Theory of Algorithms, LNCS, vol. 5028, pp. $306 - 315$ (2008)
6. P. Koepke, P. Welch. A Generalized Dynamical System, Infinite Time Register Machines, and $\Pi^1_1 - CA_0$ In CiE 2011. B. Löwe et al. (eds.), LNCS 6735, $152 - 159$ (2011)

Eigenforms, Natural Computing and Morphogenesis

Arturo Carsetti

University of Rome
{a.carsetti@gmail.com}

Decades later, Scott's basic intuition, that computability could be interpreted as continuity, continues to exert a decisive influence. In accordance with this seminal intuition, it is possible to identify the features characterizing a reflexive domain, a development language, that is, able to express and reflect within itself the structure of the "ideas" and the performances which constitute its texture, as well as to express, still within itself, its own truth predicate. In a reflexive domain every entity has an eigenform, i.e. fixed points of transformations are present for all transformations of the reflexive domain. With respect to a reflexive model and according to von Foerster and L. Kauffman, the objects of our experience appear as the fixed points of specific operators, these operators, in turn, constitute the structures of our perception. The classical reflexive models cannot lead, however, to true creativity and real metamorphosis if they do not loosen the knot of the intricate relationships between invariance and morphogenesis as it arises with respect to the actual realization of a specific embodiment. Hence the necessity of making reference to theoretical tools more complex and variegated (as, for instance, the tools offered by non-standard mathematics and epistemic complexity theory) in order to provide an adequate basis for a meaningful theoretical extension of these very models. Let us resort to an exemplification: the von Koch curve is an eigenform, but it is also a fractal. However, it can also be designed and explored utilizing the sophisticated mechanisms of non-standard analysis. In this last case, we have the possibility (but at the level of a coupled system and in the presence of specific co-evolutionary processes) to enter a universe of replication, which may also open to the reasons of real emergence. At this level, the growth of the linguistic domain and the correlated introduction of ever-new individuals appear strictly linked to the opening up of meaning and to a continuous unfolding of hidden potentialities with respect to this very opening. Hence the very necessity to bring back the inner articulation of the eigenforms not only to the structures of "simple" perception but also to the motifs of intentionality. This line of analysis permits to revisit some original ideas by Turing about morphogenesis, it permits, in particular, to understand that to go beyond Turing (but in accordance with his seminal ideas) we have to simulate, explore and prime (within the fuzzy boundaries of a coupled universe) the possible paths of morphogenesis by means of models that should be objectively identified with reference to a continuous (and self-organizing) remodeling of our neural system.

Bounded Primitive Recursive Randomness

Douglas Cenzer
University of Florida, Gainesville, FL 32611-8105, *cenzer@math.ufl.edu*

Keywords: algorithmic randomness, computability, effectively closed sets

There are three interesting versions of algorithmic randomness for real numbers which have been intensely studied in recent years. Let $X = (X(0), X(1), \dots)$ be an element of $\{0,1\}^{\mathbb{N}}$. First, we may say that X is random if it is *incompressible*, that is, the initial segments $(X(0), X(1), \dots, X(n))$ have high Kolmogorov or Levin-Chaitin complexity. Second, we say that X is random if it is *typical*, that is, X belongs to all effective sets of measure zero, in the sense of Martin-Löf. Third, we say that X is random if it is unpredictable, that is, there is no effective martingale which one can use to successfully bet on the values of X. The usual notion of algorithmic randomness is that of 1- randomness, where definitions have been given in all three versions and been shown by Schnorr to be equivalent. Many other notions of algorithmic randomness have been studied and it is often difficult to find equivalent formulations of all three sorts.

There are some important properties of Martin-Löf randomness which are need for applications. First, there is van Lambalgen's theorem, which states that the join $A \oplus B$ of two random sets is random iff A is random relative to B and B is random. Second, there is Ville's theorem, which states that any effective subsequence of a random sequence is also random.

In this paper, we present three equivalent formulations of the notion of bounded primitive recursive algorithmic randomness, as well as relativized versions, and prove Ville's theorem and van Lambalgen's theorem for this notion.

Here are the three equivalent notions of bounded randomness.

Version I: Let $C_M(\tau)$ be the length $|\sigma|$ of the shortest string σ such that $M(\sigma) = \tau$. An infinite sequence X is *bounded primitively recursively (b.p.) random* if there do not exist primitive recursive M and f such that, for every c, $C_M(X \upharpoonright f(c)) \leq f(c) - c$, that is, X cannot be primitive recursively compressed.

Version II: A *bounded primitive recursive test* is a primitive recursive sequence $\{U_n : n \in \mathbb{N}\}$ of clopen sets such that, for each n, $\mu(U_n) < 2^{-n}$. X *passes* this test if there is some n such that $X \notin U_n$. X is b.p. random if X *passes* every bounded primitive recursive test.

Version III: A *martingale* is a function $d : \{0,1\}^* \to \mathbb{Q} \cap [0, \infty]$ such that, for all σ, $d(\sigma) = \frac{1}{2}(d(\sigma^\frown 0) + d(\sigma^\frown 1))$. The martingale d *succeeds primitive recursively* on X if there is a primitive recursive function f such that, for all n, $d(X \upharpoonright f(n)) \geq 2^n$. Then X is b.p. random if there is no primitive recursive martingale which succeeds primitive recursively on X.

Resolution Systems with Substitution Rules*

Anahit Chubaryan and Armine Chubaryan

Department of Informatics and Applied Mathematics, Yerevan State University, Armenia
{achubaryan,chubarm}@ysu.am

For Classical, Intuitionistic and Minimal (Johansson's) propositional logics (CPL, IPL and MPL) we introduce the family of resolution systems with full substitution rule and with restricted substitution rule.

For CPL we use the well-known notions of literal, clause, resolution rule and definition of resolution system RC. Let $\mathfrak{C}$ be a set of clauses. We introduce the substitution rule allows one to derive from the set of clauses $\mathfrak{C}$ the results of substitution of some formula instead of a variable everywhere in the clauses of the set $\mathfrak{C}$, and *generalized* resolution rule, resolving on either some literal or in any steps substituted formula. By SRC we denote the system RC with substitution rule and generalized resolution rule. If the number of connectives of substituted formulas is bounded by ℓ, then the corresponding system is denoted by S_ℓRC. The analogous systems SRI, S_lRI, SRM, SR_lM for IPL and MPL are constructed also.

We use the known definitions of Frege systems $\mathcal{F}$, ($\mathcal{F}$I, $\mathcal{F}$M) for CPL (IPL, MPL), cut-free sequent system LK^- for CPL and cut-free multi-succedent sequent systems LI^-_{mc} for IPL and LM^-_{mc} for MPL. By LK^l (LI^l_{mc}, LM^l_{mc}) we denote the system LK^- (LI^-_{mc}, LM^-_{mc}) augmented with cut-rule, where the number of connectives of every cut formula is bounded by l, and by LK (LI_{mc}, LM_{mc}) - the corresponding system with unrestricted cut rule.

We use also the notions of p-equivalence and exponential speed-up.

The main results are:

Theorem 1.
1) $\forall \ell \geq 0$ $S_{\ell+1}RC$ has exponential speed-up over the $S_\ell RC$ (in tree form).
2) SRC, $\mathcal{F}$, LK are p-equivalent.

Theorem 2.
1. For every $\ell \geq 0$ $LI^{\ell+1}_{mc}$ ($LM^{\ell+1}_{mc}$) has exponential speed-up over LI^{ℓ}_{mc} (LM^{ℓ}_{mc}) in tree form.
2. For every $\ell \geq 0$ $S_{\ell+1}RI$ ($S_{\ell+1}RM$) has exponential speed-up over $S_\ell RI$ ($S_\ell RM$) in tree form.
3. The system SRI (SRM), $\mathcal{F}I$ ($\mathcal{F}M$) and LI^-_{mc} (LM^-_{mc}) are p-equivalent.

For proving this theorems we use the results of N.Arai about speed-up between LK^{l+1} and LK^l and method of transformation of resolution refutation into cut-free sequent proof, described by first author of this abstract.

* Supported by grant 11-1b023 of Government of RA

Linearization of Proofs in Propositional Hilbert Systems

Karel Chvalovský*

Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod Vodárenskou věží 271/2, 182 07 Prague 8, Czech Republic

chvalovsky@cs.cas.cz

Keywords: Hilbert-style proof systems, propositional logic, linear proofs

Let us have a propositional Hilbert-style proof system containing axioms (strictly speaking schemata of axioms) B (prefixing) and B′ (suffixing)

(B) $(\varphi \to \psi) \to ((\chi \to \varphi) \to (\chi \to \psi))$
(B′) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$

with implicit substitution and modus ponens as the only rule. We prove that any proof in such a proof system can be transformed into a linear proof. A proof is linear if it uses only a modified version of modus ponens: from φ and $\varphi \to \psi$ derive ψ, where φ can only be an instance of an axiom or assumption.

As prefixing and suffixing are provable in many propositional logics we can obtain similar property for many sets of axioms by adding B and B′. However, a new linear proof can be significantly longer than the original proof. It means that this result is unlikely to be used for the actual proof search, but it can be used for some metamathematical considerations.

* The author was supported by grant P202/10/1826 of the Czech Science Foundation.

Imperfect Information in Logic and Concurrent Games

Pierre Clairambault, Julian Gutierrez, Glynn Winskel

University of Cambridge Computer Laboratory, England

Concurrent games on event structures were introduced in [1] as a new basis for the formal semantics of concurrent systems and programming languages; these games allow for the explicit representation of causal dependencies between the events of a system. Recently, the concurrent games model was extended in [2] by winning conditions, in order to specify objectives for the players of the game—a useful tool for expressing and solving problems in logic and verification.

The games studied in [2] are of perfect information and determined (a property of games that ensures the existence of winning strategies) whenever restricted to games that are well-founded and satisfy a structural property, 'race-freedom', which prevents one player from interfering with the available moves to the other. This games model was used in [2] to provide a concurrent-game semantics for the predicate calculus, where *nondeterministic* winning strategies can be effectively built and deconstructed in a *compositional* manner.

We have now extended further the work in [2] by (i) defining a *bicategory of concurrent games and nondeterministic strategies* which allows imperfect information within the games and (ii) providing an *imperfect-information concurrent-game semantics* for a variant of Hintikka and Sandu's Independence-Friendly (IF) logic; we call Λ-IF this new logic since it introduces an explicit preorder Λ on variables. Such a preorder can be used to represent independence of formulae as well as "access levels" of information which restrict the allowable strategies of the semantic evaluation games for Λ-IF. The concurrent-game semantics in this work generalises the denotational model for the predicate calculus in [2].

Although strongly related to IF, the logic Λ-IF has a different evaluation game: a formula $\psi \vee \neg\psi$ is always a tautology within Λ-IF (since the co-called 'copy-cat strategy' is winning there), whereas it is not in IF when ψ is undetermined. Nevertheless, IF can be encoded into Λ-IF via its representation using Henkin quantifiers—a partial order generalization of those in classical logic.

The introduction of imperfect information, in particular, allows for a more accurate representation of the independent behaviour of the processes within a concurrent system. Although these games are no longer determined, nondeterministic winning strategies can still be constructed following a compositional approach. This feature of our games model may be the basis of a general framework for defining game-based reasoning tools and techniques for logics of independence. In fact, our results allow for the possibility of giving a proof theory for Λ-IF since the 'axiom rule' holds, which was not the case for IF.

1. Rideau, S., Winskel, G.: Concurrent strategies. In LICS, pp. 409–418, 2011.
2. Clairambault, P., Gutierrez, J., Winskel, G.: The winning ways of concurrent games. In LICS, 2012.

Infinity, Interaction and Turing Machines

Paul Cockshott(1), Greg Michaelson(2)

(1) University of Glasgow, william.cockshott@glasgow.ac.uk
(2) Heriot-Watt University, G.Michaelson@hw.ac.uk

The paper examines how as time passes the results of a famous scientists work are given a stylised standard interpretation that is frequently somewhat less complex than what they actually said. Far more computer scientists have read accounts of the Turing Machine in textbooks than have read his original description, and as a result a number of misconceptions have taken hold. In this paper, we explore two misconceptions about properties of Turing machine tapes, that is that they are infinite and that they may not be subject to external change during computations.

There is a widespread notion that Turing machines have infinite tapes. However, in Turing's original paper[5] there is no mention of the tape having to be infinite; since he is concerned with finite computation, it is clear that any terminating programme will have only modified a finite portion of the tape, so what he was assuming was effectively a finite but unbounded tape. We review the differing accounts given in the literature of the finity or infinity of the tape.

A more recent stylisation is the idea that, according to Turing, universal computers are not allowed to perform I/O. We examine and refute such claims by Wegner[7,1], arguing that the tape in Turing's model is both a store and an input output device. We argue that the key computational innovation Turing was concerned with in [5] was the Universal Computer, the computer that could emulate any other computer, and thus perform any computation. Special purpose computing machines were well known prior to Turing[2,3]. What was new, or at least a recovery of Babbage[4], in [5] was the idea of the Universal Computer. We further argue that in his mature conception of the universal computer[6], Turing allows for more convienient I/O devices like interactive terminals.

References

1. P. Cockshott, L. Mackenzie, and G. Michaelson, *Computation and its limits*, OUP, 2012.
2. T. Dawson, *Range Keeper For Guns*, November 30 1909, US Patent 941,626.
3. M. d'Ocagne, *Le calcul simpifié par les procédés mécaniques et graphiques: Esquisse générale compernant: calcul mécanique, calcul graphique, cacul graphomécanique, calcul nomographique, calcul nomomécanique*, Gauthier-Villars, 1928.
4. D. Lardner, *Babbage's calculating engines*, Edinburgh Review **59** (1834), 263–327.
5. A. Turing, *On Computable Numbers, With an Application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society **42** (1937), 230–65.
6. ———, *Lecture on the Automatic Computing Engine, 1947* , The Essential Turing (B. J. Copeland, ed.), OUP, 2004.
7. P. Wegner, *Why interaction is more powerful than algorithms*, Communications of the ACM **40** (1997), no. 5, 80–91.

Mental Experience and the Turing Test: This Double Face is the Face of Mathematics

Dedicated to Alan Turing's 100th Birthday

Leon N Cooper
Departments of Physics and Neurosciences, Institute for Brain and Neural Systems
Brown University, Providence, RI, 02912

Sorin Istrail
Department of Computer Science, Center for Computational Molecular Biology
Brown University, Providence, RI, 02912

We seem to be on the threshold of understanding the physiological basis for learning and for memory storage. But how would knowledge of events on the molecular and cellular level relate to human thought? Is complex mental behavior a large system property of the enormous numbers of units that are the brain? How is it that consciousness arises as a property of a very complex physical system? Undoubtedly, these questions are fundamental for a theory of the mind. On the other hand, there are questions of basic importance, pioneered by Turing, for his theory of the "human computer," that is, discrete state machines that "imitate" perfectly the mental processes achievable by his "human computer"; we will refer to it, although the name is only partially true to his vision, as his theory of "computational intelligence." Finding even a level of commonality to discuss both a theory of the mind with a theory of computational intelligence has been one of the grand challenges for mathematical, computational and physical sciences. The large volume of literature, following Turing's seminal work, about the computer and the brain and involving some of the greatest scientists of all time is a testimony to his genius.

In this paper we discuss, in the context of the Turing test, recent developments in physics, computer science, and molecular biology at the confluence of the above two theories, inspired by two seminal questions asked by Turing. First, about the physical not reducible to computation: "Are there components of the brain mechanism not reducible to computation?" or more specifically, "Is the physical space-time of quantum mechanical process, with its so called Heisenberg uncertainty principle, compatible with a [Turing] machine model?" Second, about computing time: "[in the Turing test] To my mind this time factor is the one question which will involve all the real technical difficulty." We relate the above questions to our work, respectively, on superconductivity and quantum mechanics, and the Ising model and the proof of its computational intractability (NP-completeness) in every 3D model, and share lessons learned to discourage, under high and long-term frustration of failure, the retreat under the cover of the positivist philosophy or other evasions.

Inspired by von Neumann, we relate the Turing's questions and his test difficulties to von Neumann's thesis about the "peculiar duplicity" of mathematics with respect to the empirical sciences. As von Neumann put it: "This double face is the face of mathematics." With a non-*a priori* concept of truth "the very concept of 'absolute' mathematical rigor is not immutable. The variability of the concept of rigor shows that something else besides mathematical abstraction must enter into the makeup of mathematics... something nonmathematical, somehow connected with the empirical sciences or with philosophy or both, does enter essentially…"

Are Bacteria Unconventional Computing Architectures?

Jole Costanza[1], Claudio Angione[2], Pietro Lió[2], and Giuseppe Nicosia[1]

[1]Department of Mathematics & Computer Science, University of Catania, Italy
[2]Computer Laboratory, University of Cambridge, UK
{costanza,nicosia}@dmi.unict.it
{claudio.angione,pietro.lio}@cl.cam.ac.uk

The logic of the metabolism could be framed as a Turing Machine; this providing an analogy between the von Neumann architecture and a bacterium. We introduce an effective formalism that enables to describe the behaviour of the bacterial cells in terms of the von Neumann architecture. One can think of the genome sequence as an executable code specified by a set of commands in a sort of ad-hoc low-level programming language. In our work, each combination of genes is coded as a string of bits $y \in \{0, 1\}^L$, each of which represents a gene set. By turning off a gene set, we turn off the chemical reaction associated with it. The bacterium takes as input chemicals (substrates) necessary for his growth and duplication, and thanks to its biochemical network (coded by the genes of its genome), produces small metabolites as output The string y is a program stored in the memory unit. The control unit is a function g_Φ that defines a partition of the string, and is uniquely determined by the pathway-based subdivision of the chemical reaction network. The processing unit of the bacterium could be modelled as the collection of all its chemical reactions. In this regard, a Turing Machine can be associated with the chemical reaction network of bacteria. By investigating the whole metabolism of bacteria considering pathways of many proteins, we extend the Bray's idea, i.e. thinking of a protein as a computational element. The cell receives, processes, and responds to inputs from the environment. However, since the question *'what does this cell do?'* has often more than one correct answer, we program molecular machines using a novel algorithm called Genetic Design through Multi-Objective (GDMO) optimisation. GDMO acts on the genetic level of the organism to find which are the genetic strategies in order to obey control signals; then, it executes the optimisation of multiple biological functions. GDMO explores effectively the whole space of gene knockouts. We optimise acetate and succinate, as well as other multiple biological functions in *E. coli, iAF1260*, and we compare our method with state-of-the-art ones, e.g. GDLS, OptFlux, OptGene, OptKnock. We use the Pareto optimal solutions with the aim of producing useful metabolites and effective drugs. Each point of the Pareto Front is a molecular machine able to accomplish a particular task. Pareto optimality is important to obtain not only a wide range of Pareto optimal solutions, but also the *best trade-off design*. Finally, we propose a solution for the problem of making the sensitivity analysis pathway-dedicated, and we call it Pathway oriented Sensitivity Analysis (PoSA). This method interrogates the functional components of the molecular machine, and reveals the most sensitive ones. To sum up, are bacteria unconventional computing architectures? Our work suggests we may answer in the affirmative.

Algorithmic complexity, entropy and the discovery of causal correlations

Duilio D'Alfonso

Università della Calabria
duilio.dalfonso@tin.it

Keywords: Kolmogorov complexity, entropy, induction, causality

In the framework of algorithmic information theory [1], complexity of an object (a bit-string codifying a set of data, a function, etc.) is expressed in terms of the length of the shortest program computing it. Shortening implies searching for regularities: if you find a recurrent pattern in a string of symbols, you can provide a receipt generating it which is significantly shorter than the string itself. Now, regular patterns, in a sequence of phenomena are, in the Humean-empiricist tradition, the only evidence of the existence of causal correlations. The Stuart Mill's famous theory of induction [2], expressly investigating the epistemological aspect of causality, is based on this assumption. When Solomonoff [3], independently from Kolmogorov, proposed the idea of (the length of) minimal programs as measure of complexity he was arguing for a theory of inductive methods. In his model, a theory is conceived as a program generating data to be explained. The shorter is the program, the more general is the theory. The transition from description to explanation is related to the real possibility of shortening. Hence, we can plausibly argue that minimization procedures can be generally considered as models of the inferential process from a set of (positive and negative) samples to a general law asserting cause-effect patterns. The plausibility of this hypothesis can be shown by means of a theoretical argument: Briefly, the rarity of non-random strings means that they are extremely unlikely. So, the presence of some regular pattern that is purely casual is highly unlikely. The detection of regularities is a sign that some kind of causal correlations must be active. This hypothesis is corroborated by some researches in machine learning such as decision tree induction. Tree induction algorithms [4] can be interpreted as a paradigmatic example of induction as minimization. I will show that in these algorithms an important role is played by entropy minimization. Entropy gain involves the discovery of parameters that induce order, by reducing entropy, in a set of samples structured as a matrix $N \times M$, where the N-th component is the target predicate and the other $N-1$ components are parameters supposed to be relevant. Inducing order means establishing a hierarchy in the causal relevance of parameters. To sum up, the theory of inductive reasoning known as decision tree induction could be a good candidate to formalize the reasoning inferring cause-effect patterns. There is a strong analogy in the characterization of inductive inference between tree induction methods and the Stuart Mill's canons for inferring causal relations. Making explicit this analogy could be a promising direction to argue in favour of this hypothesis.

References

1. Li, M., Vitànyi, P.: An introduction to Kolmogorov complexity and its applications. Springer-Verlag (1997)
2. Stuart Mill, J.: A System of logic, in: Robson, J. M. (ed.), Collected Works of John Stuart Mill, Toronto: University of Toronto Press (1963-1991)
3. Solomonoff, R. J.: A preliminary report on a general theory of inductive inference. Technical Report ZTB-138, Zator Company, Cambridge, Mass. (1960)
4. Quinlan, J., Rivest, R.: Inferring decision trees using the minimum description length principle. Inform. Comput. **80** (1989) 227–248

Computational Complexity of Interactive Behaviors

Ugo Dal Lago[1], Tobias Heindel[2], Damiano Mazza[3], and Daniele Varacca[4]

[1] Dip. di Scienze dell'Informazione – Univ. di Bologna, Italy – dallago@cs.unibo.it
[2] CEA – LIST, Gif-sur-Yvette, France – Tobias.Heindel@cea.fr
[3] LIPN – CNRS and Université Paris 13, France – mazza@lipn.univ-paris13.fr
[4] PPS – CNRS and Université Paris Diderot, France – varacca@pps.jussieu.fr

The standard theory of computational complexity is concerned with studying the intrinsic difficulty of problems (or functions). These may be seen as processes whose interactive behavior is trivial, *i.e.*, is limited to the question/answer pattern. We set out to provide grounds for a revised theory of computational complexity that is capable of gauging the difficulty of genuinely interactive behaviors.

Following standard lore of concurrency theory, we formally define behaviors as equivalence classes of *labelled transition systems* (LTS), which in turn are usually specified using process calculi, such as Milner's CCS [1]. Usual functions are recovered as a particularly simple form of behavior, called *functional*.

The next step is the development of an abstract cost model for interactive computation. In the functional case, a cost model is just a measure of the resources (time, space) required to compute the answer as a function of the size of the question. In the interactive case, we propose to measure costs using *weighted asynchronous* LTS (or WALTS): asynchrony is a standard feature that is added to transition systems to represent causal dependencies [2], which we need to generalize the trivial dependency between questions and answers; weights are used to specify additional quantitative information about space and time consumption.

Finally, we introduce a computational model, the *process machine*, which implements behaviors (just as Turing machines implement functions) by executing concurrent programs written in a CCS-based language. The process machine admits a natural semantics in terms of WALTSs and thus provides us with a non-trivial, paradigmatic instance of our abstract framework for measuring the complexity of behaviors.

Complexity classes are then defined as sets of behaviors that can be implemented by a process running within given time and space bounds on the process machine. We conclude by showing that if we restrict to functional behaviors we obtain several standard complexity classes; thus, at least in many paradigmatic cases, we have in fact a consistent extension of complexity theory into the realm of interactive computation.

References

1. Milner, R.: Communicating and Mobile Systems: The π-calculus. Cambridge University Press (1999)
2. Winskel, G., Nielsen, M.: Models for concurrency. In: Handbook of logic in Computer Science, vol. 4. Clarendon Press (1995)

Book Presentation:
The Dawn of Software Engineering: from Turing to Dijkstra

Edgar G. Daylight

Eindhoven University of Technology
Department of Technology Management
egdaylight@dijkstrascry.com

Abstract. Turing's involvement with computer building was popularized in the 1970s and later. Most notable are the books by Brian Randell (1973), Andrew Hodges (1983), and Martin Davis (2000). A central question is whether John von Neumann was influenced by Turing's 1936 paper when he helped build the EDVAC machine, even though he never cited Turing's work. This question remains unsettled up till this day. As remarked by Charles Petzold, one standard history barely mentions Turing, while the other, written by a logician, makes Turing a key player.

Contrast these observations then with the fact that Turing's 1936 paper *was* cited and heavily discussed in 1959 among computer programmers. An historical investigation of Turing's influence on computing shows that Turing's 1936 notion of universality became increasingly relevant among computer programmers during the 1950s. In 1966, the first Turing award was given to a computer programmer, not a computer builder, as were several subsequent Turing awards. In short, Turing had already acquired fame and recognition in the emerging field of software, well before his alleged role in the advent of the 'first universal computer' was publicized.

The aforementioned historical investigation is described in my book *The Dawn of Software Engineering: from Turing to Dijkstra*, published by Lonely Scholar in Spring 2012 (*www.lonelyscholar.com*). My central thesis is that **Turing's influence was felt more in programming after his death than in computer building during the 1940s.** Moreover, Turing's influence only emerged gradually and often without full comprehension. Many first-generation programmers did not read Turing's 1936 paper, let alone understand it. Those computer practitioners who did become acquainted with Turing's 1936 work during the 1950s–1960s received it in at least one of the following three ways:

1. The 'Turing machine' was viewed as a model for a computer. Some researchers tried to build 'Turing machines' during the 1950s, i.e. *after* the first all-purpose physical computers were already available.
2. Turing's notion of universality helped lay the foundations of programming.
3. The unsolvability of the Halting Problem helped researchers lay bare certain limitations in programming. The 1972 Turing award winner, Edsger W. Dijkstra, was definitely one of the first to do so.

Revamping the Turing Test by Computing Interaction Coupling

Gregorio de Miguel Casado[1*] and Manuel González Bedia[2**]

Aragon Institute for Research in Engineering (I3A)
University of Zaragoza, Spain
{gmiguel,mgbedia}@unizar.es

One of the most valuable and discussed contributions made by Alan Turing in the field of Artificial Intelligence was the famous test that takes his name. The standard interpretation presents the test as an effective criterion for determining if we are interacting with an intelligent entity or not. Turing was a man of his times, and his approach fell within the tradition of *behaviorism*: passing the test was supposed to be sufficient to justify ascriptions of mentality. In our contribution, we wonder about how Turing would have shaped his ideas if he had lived today and knew other approaches such as the embodied cognition and the enactive social cognition ones, which assume that recognizing other person is not a simulation process and that deducing the other's experience means is not always achieved through theoretical inference. Nowadays, in the light of scientific findings from disciplines such as cognitive and social sciences, a more sophisticated approach is proposed as to model how people act together and understand each other in interactive situations.

The main aim of the first part of this contribution is to show how the role of interactive elements in social cognition can be considered in order to motivate a re-evaluation of the foundations and limits of the Turing test. A review of Turing's use of the notion of *time* in the interaction and concepts such as *interactive coupling* will also be addressed. In the second part, these ideas are presented in mathematical terms under the scope of Computable Analysis. Rather than making an inference from what the other person does, we formalize the way to incorporate other people's actions in terms of own goals/values within contextualized situations. In this regard, the notion of *intelligent project* is introduced. It is obviously assumed that the human capacity to recognize other humans is rooted in strategies as the observation and reflection on others' actions. However, the objective of this article is to determine the way to include the *coupling role* involved in a social process in a formal framework based on Computable Analysis such as to provide feasible applications of it within Computer Science.

* Research supported by the Spanish Government MICINN Project TIN2011-27479-C04-01.

** Research supported by the Spanish Government MICINN Project TIN2011-24660.

A short history of small machines

Liesbeth De Mol[1⋆] and Maarten Bullynck[2]

[1] Center for Logic and Philosophy of Science, University of Ghent, Blandijnberg 2, 9000 Gent, Belgium
elizabeth.demol@ugent.be
[2] Université Paris VIII, Vincennes Saint-Denis
maarten.bullynck@kuttaka.org

"*This operation is so simple that it becomes laborious to apply*" (Lehmer, 1933)

One of the most famous results of Alan M. Turing is the so-called universal Turing machine (UTM). Its influence on (theoretical) computer science can hardly be overestimated. The operations of this machine are of a most elementary nature but nonetheless considered to capture all the (human) processes that can be carried out in computing a number. This kind of elementary machine fits into a tradition of '*logical minimalism*' that looks for simplest sets of operations or axioms. It is part of the more general research programme into the foundations of mathematics and logic that was carried out in the beginning of the 20th century. In the 1940s and 1950s, however, this tradition was redefined in the context of 'computer science' when computer engineers, logicians and mathematicians re-considered the problem of small(est) and/or simple(st) machines in the context of actual engineering practices. This paper looks into this early history of research on small symbolic and physical machines and tie it to this older tradition of logical minimalism. Focus will be on how the transition and translation of symbolic machines into real computers integrates minimalist philosophies as parts of more complex computer design strategies. This contextualizes Turing's machines at the turn from logic to machines.

⋆ Postdoctoral Fellow of the Fund for Scientific Research – Flanders (FWO)

Complexity of Fast Searching

Öznur Yaşar Diner[1], Dariusz Dereniowski[2], and Danny Dyer[3]

[1] Department of Information Technology, Kadir Has University, Turkey
oznur.yasar@khas.edu.tr
[2] Department of Algorithms and System Modeling, Gdańsk University of Technology, Poland deren@eti.pg.gda.pl
[3] Department of Mathematics and Statistics, Memorial University of Newfoundland, Canada dyer@math.mun.ca

Abstract. In the edge searching problem, searchers move from vertex to vertex in a graph to capture an invisible, fast intruder that may occupy either vertices or edges. Fast searching is a monotonic internal model in which, at every move, a new edge of the graph G must be guaranteed to be free of the intruder. That is, once all searchers are placed the graph G is cleared in exactly $|E(G)|$ moves. Such a restriction obviously necessitates a larger number of searchers. We examine this model, and characterize graphs for which 2 or 3 searchers are sufficient. We prove that the corresponding decision problem is NP-Hard. This is achieved by reducing the problem of node search to fast search (via a third game, called weak node-search). This result also shows a relation between the node search game and the fast search game. This is independent from the proof of B. Yang (2011).

Alan Turing's Legacy: Info-Computational Philosophy of Nature

Gordana Dodig-Crnković

School of Innovation, Design and Engineering, Mälardalen University
Västerås, Sweden
gordana.dodig-crnkovic@mdh.se
http://www.idt.mdh.se/~gdc01

Andrew Hodges describes Turing as a Natural philosopher: "He thought and lived a generation ahead of his time, and yet the features of his thought that burst the boundaries of the 1940s are better described by the antique words: natural philosophy." Turing's natural philosophy differs from Galileo's view that the book of nature is written in the language of mathematics (The Assayer, 1623). Computing differs from mathematics in that computers not only calculate numbers, but more importantly produce real time behaviors. Turing studied a variety of natural phenomena and proposed their computational modeling. He made a pioneering contribution in the elucidation of connections between computation and intelligence and his work on morphogenesis provides evidence for natural philosophers approach. Turing's 1952 paper on morphogenesis proposed a chemical model as the basis of the development of biological patterns such as the spots and stripes that appear on animal skin.

Turing did not originally claim that the physical system producing patterns actually performs computation through morphogenesis. Nevertheless, from the perspective of info-computationalism we can argue that morphogenesis is a process of morphological computing. Physical process, though not computational in the traditional sense, presents natural (unconventional), physical, morphological computation. An essential element in this process is the interplay between the informational structure and the computational process information self-structuring. The process of computation implements physical laws which act on informational structures. Through the process of computation, structures change their forms. All computation on some level of abstraction is morphological computation; a form-changing/form-generating process.

In this article, info-computationalism is identified as a contemporary philosophy of nature providing the basis for the unification of knowledge from currently disparate fields of natural sciences, philosophy, and computing. An on-going development in bioinformatics, computational biology, neuroscience, cognitive science and related fields shows that in practice biological systems are currently already studied as information processing and are modeled using computation-theoretical tools, according to Rozenberg and Kari. [1]

[1] The full article: http://www.mrtc.mdh.se/~gdc/work/cie-2012-dodig-8.pdf.

▶ SEBASTIAN EBERHARD, *Applicative theories for logarithmic complexity classes.*
Institut für Informatik und angewandte Mathematik, Universität Bern, Neubrückstr. 10, CH-3012 Bern, Switzerland.
E-mail: eberhard@iam.unibe.ch.

Applicative systems are a formalisation of the lambda calculus and form the base theory of Feferman's explicit mathematics. For many linear and polynomial complexity classes corresponding applicative systems have already been developed by authors as Kahle, Oitavem, and Strahm. In contrast to the setting of bounded arithmetic, this setting allows very explicit and straightforward lower bound proofs because no coding of graphs of functions is necessary.

We present natural applicative theories for the logarithmic hierarchy, alternating logarithmical time, and logarithmic space. For the first two algebras, we formalize function algebras having concatenation recursion as main principle. For logarithmical space, we formalize an algebra with safe and normal inputs and outputs. This algebra allows to shift small safe inputs to the normal side.

The mentioned theories all contain predicates for normal, respectively safe words. The set of safe words intuitively collects stored but not fully accessible words. The interplay between normal words, being fully accessible, and safe words allows an elegant formulation of induction principles justifying concatenation - and sharply bounded recursion.

A Simplified Characterisation of Provably Computable Functions of the System $\mathbf{ID}_1$

Naohi Eguchi[1*] and Andreas Weiermann[2]

[1] Mathematical Institute, Tohoku University,
6-3, Aramaki Aza-Aoba, Aoba-ku, Sendai 980-8578, Japan
eguchi@math.tohoku.ac.jp
[2] Department of Mathematics, Ghent University,
S22, Krijgslaan 281, B9000 Ghent, Belgium
weiermann@cage.ugent.be

Keywords: Provably Computable Functions; System of Inductive Definitions; Ordinal Notation Systems; Operator Controlled Derivations.

As stated by Gödel's second incompleteness theorem, any reasonable consistent formal system has an unprovable Π^0_2-sentence that is true in the standard model of arithmetic. This means that the total computable functions whose totality is provable in a consistent system, which are known as *provably (total) computable functions*, form a proper subclass of total computable functions. It is natural to ask how we can describe the provably computable functions of a given system. We present a simplified and streamlined characterisation of provably computable functions of the system $\mathbf{ID}_1$ of non-iterated inductive definitions. The idea is to employ the method of operator-controlled derivations that was originally introduced by Wilfried Buchholz ([1]) and afterwards applied by the second author to a characterisation of provably computable functions of Peano arithmetic PA([3]). This work aims to lift up the characterisation in [3] to $\mathbf{ID}_1$. We introduce an ordinal notation system $\mathcal{O}(\Omega)$ and define a computable function f^α for a starting number-theoretic function $f : \mathbb{N} \to \mathbb{N}$ by transfinite recursion on $\alpha \in \mathcal{O}(\Omega)$. The transfinite definition of f^α stems from [3]. We show that a function is provably computable in $\mathbf{ID}_1$ if and only if it is a Kalmar elementary function in $\{\mathsf{s}^\alpha \mid \alpha \in \mathcal{O}(\Omega) \text{ and } \alpha < \Omega\}$, where s denotes the successor function and Ω denotes the least non-recursive ordinal. The proof can be found in [2].

References

1. W. Buchholz. A Simplified Version of Local Predicativity. *Proof Theory*, pages 115–148. Cambridge University Press, Cambridge, 1992.
2. N. Eguchi and A. Weiermann. A Simplified Characterisation of Provably Computable Functions of the System $\mathbf{ID}_1$ of Inductive Definitions. Technical Report. *CoRR*, 1205.2879, 27 pages, 2012.
3. A. Weiermann. Classifying the Provably Total Functions of PA. *Bulletin of Symbolic Logic*, 12(2):177–190, 2006.

* The first author is supported by the research project *Philosophical Frontiers in Reverse Mathematics* sponsored by the John Templeton Foundation.

Automated Certification of a Logic-Based Verification Method for Imperative Loops

Mădălina Eraşcu* and Tudor Jebelean

Research Institute for Symbolic Computation, Johannes Kepler University
Altenbergerstraße 69, A-4040, Linz, Austria
merascu@risc.jku.at

In the *Theorema* system (www.theorema.org), we prove automatically the soundness of a verification method handling abruptly terminating `while` loops in imperative programs. The method is based on path-sensitive symbolic execution and functional semantics. Our main aim is the identification of the minimal logical apparatus necessary for formulating and proving (in a computer-assisted manner) a correct collection of methods for program verification. The distinctive features of our approach are:

– Loop correctness is expressed in predicate logic, without using any additional theoretical model for program semantics or for program execution, but only using the object theory (the theory of the objects handled by the program).
– The semantics of a loop is the implicit definition, at object level, of the function implemented by the loop.
– Termination is formulated as an induction principle corresponding to the structure of the `while` loop.

As usual, the verification conditions consist in invariant preservation and termination. We prove automatically that these imply the existence, the uniqueness, and the correctness of the function implemented by the loop. The knowledge necessary for our proofs contains basic axioms of natural numbers (including induction) and are performed using mainly first-order inferences (exception is Skolemization). One advantage of computer automation is that all axioms and all inference rules are explicit (no intuitive knowledge is used).

The main idea of the proof is inspired from the Scott fixpoint semantics, however our approach does not need set theory.

Our computer-aided formalization may open the possibility of reflection of the method on itself (treatment of the meta-functions as programs whose correctness can be studied by the same method). Because the formal specification and the verification of the method are performed in the same framework one could reason at object- and meta-level in the same system.

* Recipient of a DOC-fFORTE-fellowship of the Austrian Academy of Sciences

Toward the unification of logic and geometry as the basis of a new computational paradigm

Steven Ericsson-Zenith

Institute for Advanced Science & Engineering

In his “Aufbau” Rudolf Carnap (1891-1970) proposed that logical operations are differentiation from the entire manifold of sense and proposed “recollection of similarity” as the basic relation. This view fundamentally differs from the conventional mechanics of Alan Turing (1912-1954), that treats computable logic as the mechanical integration of logical atoms. This view, that logic be founded upon manifolds and not discrete logical atoms, was not uncommon before Alan Turing’s Universal Machine model of computation became pervasive, popularized during the critical period founding mathematical logic by the work of Ernst Schröder (1841-1902) and Charles Sanders Peirce (1839-1914).

A common argument against the manifold view is that it makes no difference to computed results. This is refuted when confronted by the challenges of general recognition and locality in large scale parallel computation.

We present work toward the development of realizable mechanisms for computable logic based upon a re-conception of logic as operations of differentiation upon closed manifolds. This approach requires a unification of conceptions in logic with natural geometric transformations of closed manifolds, combining symbol processing with response potential.

Confirmation of these mechanisms may exist in nature. Our investigation is founded upon the premise that it is the structure of closed manifolds in dynamic biophysical architectures that characterize sense and closely bind sense to directed response potential.

The presented exploration is experimental and purely mathematical. The approach argues that the effects we seek to characterize have a natural mathematical basis and that by the elimination of naive assumptions concerning apprehension from geometry a characterization of Carnap’s basic relation will suggest itself. We take this approach because it is the action of such apprehension that is the subject of our inquiry.

The resulting mechanics suggests the design and physical realization of a new model of computation; one in which structure and the concurrency of action are a first-order consideration. By this model symbolic processing is storage free and closely bound to response potentials, the capacity of symbol representation is combinatorial across these dynamic manifolds, suggesting general engineering principles that offer a symbolic processing capability in biophysical architectures greater than previously considered.

Intuitionistic asymmetry: how the classical CBN/CBV duality breaks

José Espírito Santo

Centro de Matemática**, Universidade do Minho, Portugal
jes@math.uminho.pt

In classical logic, cut-elimination has a critical pair that trivializes the identity of proofs. Through restriction of the cut-elimination rules involved, two dual sub-systems are generated that correspond to call-by-name (CBN) and call-by-value (CBV) computation. We re-evaluate this situation in the case of intuitionistic logic.

We need some proof-theoretical apparatus. Following the author's previous work[1], sequent calculus is considered along with an isomorphic natural deduction system. We define in the sequent calculus permutative conversions T and Q, whose normal forms constitute the well-known focused fragments for CBN and CBV, respectively; and by isomorphism we equip natural deduction with corresponding rules. In fact, in order to avoid using meta-theoretical devices like contexts, T (resp. Q) is best given in natural deduction (resp. sequent calculus)[2]:

$$
\begin{array}{lll}
(\mathsf{T}) & \mathsf{let}\, x = EN \,\mathsf{in}\, P \to [ap(EN)/x]P & (P \neq x) \\
(\mathsf{Q}) & (tk) :: k' \to (x)t(k@((y)x(y :: k'))) & (x, y \text{ fresh})
\end{array}
$$

The intuitionistic asymmetry between hypothesis and conclusion already generates a formal bias towards CBN that resolves the CBN/CBV critical pair. But the formal resolution can be naturally expressed in terms of permutative conversions: T solves the critical pair, while Q fails to do so. Therefore, for CBN we consider cut-elimination rules supplemented with T[3]; whereas for CBV we resort to constraining one of the cut-elimination rules.

It turns out that the obtained CBV system is in equational correspondence with Moggi's computational λ-calculus, and is therefore complete for CPS semantics, while being included in the CBN system (in the manner of Plotkin's CBV λ-calculus), since cut-elimination is not constrained in the CBN system.

Moreover, nothing forces us to consider the focused fragment of T-normal forms; but T-normalization contributes to an interesting syntactic collapse: the reflection of the whole syntax into a simple CBN fragment; in addition, if seen from the natural deduction side, the collapse explains why CBN can be defined by local reduction rules and without let-expressions.

** The author acknowledges financial support by Fundação para a Ciência e Tecnologia.

[1] *"The λ-calculus and the unity of structural proof theory"*, Theory of Computing Systems, 45:963-994, 2009

[2] Here the notation of *op.cit.* is followed, except that the primitive substitution operator of natural deduction is written as a let-expression.

[3] Curiously, supplementing cut-elimination with T *and* Q breaks termination.

A Semantic Framework for Real Computation with Polynomial Enclosures

Amin Farjudian

Division of Computer Science
University of Nottingham Ningbo, China

Amin.Farjudian@nottingham.edu.cn
www.cs.nott.ac.uk/~avf/

Abstract. Domain theory has been used with great success in providing a semantic framework for Turing computability, over both discrete and continuous spaces. On the other hand, classical approximation theory provides a rich set of tools for computations over real functions with (mainly) polynomial and rational function approximations.

We present a semantic framework for computations over real functions based on *polynomial enclosures*. As an important case study, we analyse the convergence and complexity of Picard's method of initial value problem solving in our framework. We obtain a geometric rate of convergence over Lipschitz fields and then, by using Chebyshev truncations, we modify Picard's algorithm into one which runs in *polynomial time* over polynomial space representable fields, a reduction in complexity which is impossible to achieve in the previous step-function based domain models.

Keywords: computable analysis, computational complexity, initial value problem, Picard's method, approximation theory, Chebyshev series

On the palindromic completion of words and languages

Szilárd Zsolt Fazekas[1]*, Robert Mercaş[2]**, and Kayoko Shikishima-Tsuji[3]

[1] Department of Mathematics, Kyoto Sangyo University, Motoyama, Kamigamo, Kita-Ku Kyoto 603-8555, Japan, szilard.fazekas@gmail.com
[2] Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, PSF 4120, D-39016 Magdeburg, Germany, robertmercas@gmail.com
[3] Tenri University, 1050 Somanouchi Tenri 632-8510, Japan, tsuji@sta.tenri-u.ac.jp

Palindromes are sequences which read the same starting from either end.

Several operations on words were introduced which are either directly motivated by the biological phenomenon called stem-loop completion, or are very similar in nature to it. The mathematical hairpin concept is a word in which some suffix is the mirrored complement of a middle factor of the word. The hairpin completion operation, which extends such a word into a pseudopalindrome with a non-matching part in the middle was thoroughly investigated and most basic algorithmic questions about hairpin completion have been answered with a noteworthy exception: given a word, can we decide whether the iterated application of the operation leads to a regular language?

The operation studied here, is palindromic completion, which requires the word to have a palindromic prefix or suffix in order to be completed by adding symbols to the other side of the word such that the new obtained word is a palindrome. The (iterated) palindromic closure considers all possible extensions which complete the starting word into a palindrome. This notion represents a particular type of hairpin completion, where the length of the hairpin is at most one and the binding occurs only among identical symbols.

An important observation in the case of palindromic completion is that right and the left completion of every palindrome coincide. We show that, just as in the case of the hairpin operation, even the iterated completion of a word can stand outside the regular language class. However, in this case, we are able to produce starting from a regular language a non-regular languages that is context-free. Furthermore, making use of an existing characterization of regular palindromic languages we give a characterization of all languages that remain regular after applying palindromic completion iteratively.

Due to this characterization, we are able to show that regularity of the iterated palindromic completion of regular languages is decidable. We also give techniques of efficiently answering algorithmic questions regarding the completion of single strings (membership problem, palindromic completion distance between two words).

* Work supported by *Japanese Society for the Promotion of Science* under no. *P10827*
** Work supported by *Alexander von Humboldt Foundation*

Cancellation-free circuits: An approach for proving superlinear lower bounds for linear Boolean operators*

Joan Boyar and Magnus Gausdal Find
{joan, magnusgf}@imada.sdu.dk

Department of Mathematics and Computer Science
University of Southern Denmark

Consider the following problem: Given a Boolean $n \times n$ matrix, A, what is the size of the smallest linear circuit computing A? Here linear circuit means a circuit consisting of XOR gates each having two inputs, and size denotes the number of gates. It is known that $O\left(\frac{n^2}{\log n}\right)$ XOR gates is sufficient, and that this is asymptotically tight. In fact, it holds for the vast majority of $n \times n$ matrices that $\Omega\left(\frac{n^2}{\log n}\right)$ gates are needed. Despite this fact, there exists no concrete family of matrices proven to require more than a linear number of gates. We continue the study of *cancellation-free linear circuits*. A circuit C is cancellation-free if for every pair of vertices v_1, v_2 in C, there do not exist two disjoint paths in C from v_1 to v_2.

We use a theorem by Lupanov to prove that almost all matrices can be computed by a cancellation-free circuit, whic is at most a constant factor larger than the optimal linear circuit that computes the matrix. We also provide an infinite family of matrices showing that optimal cancellation-free linear circuits can have twice the size of optimal linear circuits.

It appears to be easier to prove statements about the structure of cancellation-free linear circuits than about linear circuits in general. We prove two nontrivial superlinear lower bounds. First, we show that any cancellation-free linear circuit computing the $n \times n$ Sierpinski gasket matrix has size at least $\frac{1}{2}n \log n$. This supports a conjecture by Aaronson. The proof is based on gate elimination. This makes it the first, to the authors' knowledge, superlinear lower bound using gate elimination. Furthermore we show that a proof strategy due to Mehlhorn for proving lower bounds on monotone circuits can be almost directly converted to prove lower bounds on cancellation-free linear circuits. We use this together with a result from extremal graph theory due to Andreev to prove a lower bound of $\Omega(n^{2-\epsilon})$ for infinitely many $n \times n$ matrices for every $\epsilon > 0$. These lower bounds for concrete matrices are almost optimal since all matrices can be computed with $O\left(\frac{n^2}{\log n}\right)$ gates.

* Partially supported by the Danish Council for Independent Research, Natural Sciences.

Ambiguity of ω-Languages of Turing Machines

Olivier Finkel

Institut de Mathématiques de Jussieu
CNRS et Université Paris 7, France.
Olivier.Finkel@math.univ-paris-diderot.fr

Keywords: Automata and formal languages; infinite words; Turing machines; Büchi transition systems; ambiguity; degrees of ambiguity; logic in computer science; effective descriptive set theory; models of set theory.

An ω-language is a set of infinite words over a finite alphabet X. We consider the class of recursive ω-languages, i.e. the class of ω-languages accepted by Turing machines with a Büchi acceptance condition, which is also the class Σ_1^1 of (effective) analytic subsets of X^ω for some finite alphabet X. We investigate the notion of ambiguity for recursive ω-languages with regard to acceptance by Büchi Turing machines. We first show that the class of unambiguous recursive ω-languages is the class Δ_1^1 of hyperarithmetical sets. We obtain also that the Δ_1^1-subsets of X^ω are the subsets of X^ω which are accepted by strictly recursive unambiguous finitely branching Büchi transition systems; this provides an effective analogue to a theorem of Arnold on Büchi transition systems [Arn83]. Moreover, using some effective descriptive set theory, we prove that recursive ω-languages satisfy the following remarkable dichotomy property. A recursive ω-language $L \subseteq X^\omega$ is either unambiguous or has a great degree of ambiguity in the following sense: for every Büchi Turing machine $\mathcal{T}$ accepting L, there exist infinitely many ω-words which have $2^{\aleph_0}$ accepting runs by $\mathcal{T}$. We also show that if $L \subseteq X^\omega$ is accepted by a Büchi Turing machine $\mathcal{T}$ and L is an analytic but non Borel set, then the set of ω-words, which have $2^{\aleph_0}$ accepting runs by $\mathcal{T}$, has cardinality $2^{\aleph_0}$. Moreover, using some results of set theory, we prove that it is consistent with the axiomatic system ZFC that there exists a recursive ω-language in the Borel class $\mathbf{\Pi}_2^0$, hence of low Borel rank, which has also this maximum degree of ambiguity.

References

[Arn83] A. Arnold. Topological characterizations of infinite behaviours of transition systems. In *Proceedings of the International Conference on Automata, Languages and Programming, 10th Colloquium, Barcelona, Spain, July 18-22, 1983*, volume 154 of *Lecture Notes in Computer Science*, pages 28–38. Springer, 1983.

[CC89] J. Castro and F. Cucker. Nondeterministic ω-computations and the analytical hierarchy. *Journal Math. Logik und Grundlagen d. Math*, 35:333–342, 1989.

[CG78] R.S. Cohen and A.Y. Gold. ω-computations on Turing machines. *Theoretical Computer Science*, 6:1–23, 1978.

[PP04] D. Perrin and J.-E. Pin. *Infinite words, automata, semigroups, logic and games*, volume 141 of *Pure and Applied Mathematics*. Elsevier, 2004.

[Sta97] L. Staiger. ω-languages. In *Handbook of formal languages, Vol. 3*, pages 339–387. Springer, Berlin, 1997.

Turing Incomputable Computation

Michael Stephen Fiske

Aemea Institute, San Francisco, California
mf@aemea.org

Recent cyberattacks have demonstrated that current approaches to the malware problem (e.g., detection) are inadequate. This is not surprising as virus detection is Turing undecidable. [1] Further, some recent malware implementations use NP problems to encrypt and hide the malware. [2] Two goals guide an alternative approach: (a) Program execution should hide computational steps in order to hinder "reverse engineering" efforts by malware hackers; (b) New computational models should be explored that make it more difficult to hijack the purpose of program execution. The methods explained here pertain to (a) implemented with a new computational model, called the active element machine (AEM). [3]

An AEM is composed of computational primitives called *active elements* that simultaneously transmit and receive pulses to and from other active elements. Each pulse has an amplitude and a width, representing how long the pulse amplitude lasts as input to the active element receiving the pulse. If active element E_i simultaneously receives pulses with amplitudes summing to a value greater than E_i's threshold and E_i's refractory period has expired, then E_i fires. When E_i fires, it sends pulses to other active elements. If E_i fires at time t, a pulse reaches element E_k at time $t + \tau_{ik}$ where τ_{ik} is the transmission time from E_i to E_k.

The AEM language uses five commands – `Element`, `Connection`, `Fire`, `Program` and `Meta` – to write AEM programs, where time is explicitly specified and multiple commands may simultaneously execute. An `Element` command creates, at the time specified in the command, an active element with a threshold value, a refractory period and a most recent firing time. A `Connection` command sets, at the time specified in the command, a pulse amplitude, a pulse width and a transmission time from element E_i to element E_k. The `Fire` command fires an input element at a particular time. The `Program` command defines the execution of multiple commands with a single command. `Element` and `Connection` commands establish the AEM architecture and firing activity. The `Meta` command can change the AEM architecture during AEM program execution.

This model uses quantum random input to generate random firing patterns that deterministically execute a universal Turing machine (UTM) program η. Random firing interpretations are constructed with dynamic level sets on boolean functions that compute η. The quantum randomness is an essential component for building the random firing patterns that are Turing incomputable. [4] It is assumed that the following are all kept perfectly secret: the state and tape of the UTM, represented by the active elements and connections; the quantum random bits determining how η is computed for each computational step; and the dynamic connections in the AEM.

Formally, let $f_{1j}, f_{2j}, \dots f_{mj}$ represent the random firing pattern computing η during the jth computational step and assume an adversary can eavesdrop on $f_{1j}, f_{2j}, \dots f_{mj}$. Let q denote the current state of the UTM, a_k a UTM alphabet symbol and q_k a UTM state. Perfect secrecy means that probabilities $P(q = q_k | f_{1j} = b_1 \ \dots \ f_{mj} = b_m) = P(q = q_k)$ and $P(T_k = a_k | f_{1j} = b_1 \ \dots \ f_{mj} = b_m) = P(T_k = a_k)$ for each $b_i \in \{0, 1\}$ and each T_k which is the contents of the kth tape square. If these secrecy conditions hold, then there doesn't exist a "reverse engineer" Turing machine that can map the random firing patterns back to an unbounded sequence of UTM instructions. For an unbounded number of computational steps, define function $g : \mathbb{N} \to \{0, 1\}$ where $g((j-1)m + r) = f_{(r+1)j}$ and $0 \le r < m$. Then g is incomputable.

Proposed methods of hypercomputation currently have no physical realization. [5] The methods described here can be physically realized using an off-the-shelf quantum random generator device with a USB plug connected to a laptop computer executing a finite AEM program.

[1] Fred Cohen. Computer Viruses Theory and Experiments. Computers and Security. **6**(1), 22–35, February (1987).
[2] Eric Filiol. Malicious Cryptology and Mathematics. Cryptography and Security in Computing. Intech. 23–50 (2012).
[3] Michael Fiske. The Active Element Machine. Proceedings of Computational Intelligence. **391** Springer. 69–96 (2011). `http://www.aemea.org/AEM`
[4] Cristian Calude, Karl Svozil. Quantum randomness and value indefiniteness. Adv. Sci. Lett. **1**(2), 165–168 (2008).
[5] Martin Davis. Why there is no such discipline as hypercomputation. Applied Mathematics and Computation. **178**(1), 4–7, July (2006).

De Finetti's bets on partially evaluated frames

Tommaso Flaminio[1] and Hykel Hosni[2]

[1] IIIA - CSIC
Campus de la Univ. Autònoma de Barcelona s/n 08193 Bellaterra, Spain.
Email: tommaso@iiia.csic.es
[2] Scuola Normale Superiore,
Piazza dei Cavalieri 7 Pisa, Italy.
Email hykel.hosni@sns.it

Let $E_1, \ldots, E_n$ be of events of interest. De Finetti's *betting problem* is the choice that an idealised agent called *bookmaker* must make when publishing a *book*, i.e. when making the assignment $B = \{(E_i, \beta_i) : i = 1, \ldots, n\}$ such that each E_i is given value $\beta_i \in [0, 1]$. Once a book has been published, a *gambler* can place bets on event E_i by paying $\alpha_i \beta_i$ to the bookmaker. In return for this payment, the gambler will receive α_i, if E_i obtains and nothing otherwise. De Finetti constructs the betting problem in such a way as to force the bookmaker to publish *fair betting odds* for book B. To this end, two modelling assumptions are built into the problem, namely that (i) the bookmaker is forced to accept any number of bets on B and (ii) when betting on E_i, gamblers can choose the sign, as well as the magnitude of (monetary) stakes α_i. Conditions (i-ii) force the bookmaker to publish books with zero-expectation, for doing otherwise may offer the gambler the possibility of making a sure profit, possibily by choosing negative stakes thereby unilaterally imposing a payoff swap to the bookmaker. As the game is zero-sum, this is equivalent to forcing the bookmaker into sure loss. In this context, de Finetti proves that the axioms of probability are necessary and sufficient to secure the bookmaker against this possibility.

The crux of the Dutch book argument is the identification of the agent's degrees of belief with the *price* they are willing to pay for an uncertain reward which depends on the *future* truth value of some *presently unknown* propositions – the *events* on which the agents are betting. This clearly suggests that the semantics of events, which bears directly on the definition of probability, is implicitly endowed with an *epistemic structure*. The purpose of this paper is to give this structure an explicit formal characterisation and to show how the resulting framework helps us making de Finetti's elusive notion of *event* much clearer. In particular we shall be able to give a formal setting to the following remark:

> [T]he characteristic feature of what I refer to as an "event" is that the circumstances under which the event will turn out to be "verified" or "disproved" have been fixed in advance.

Reverse mathematics and a packed Ramsey's theorem

Stephen Flood

University of Notre Dame

Abstract. Ramsey's theorem states that each coloring has an infinite homogeneous set, but these sets can be arbitrarily spread out. Paul Erdős and Fred Galvin proved that for each coloring f, there is an infinite set that is "packed together" which is given "a small number" of colors by f. In this talk, we will give the precise statement of this packed Ramsey's theorem, and discuss its computational and reverse mathematical strength. In reverse mathematics, this theorem is equivalent to Ramsey's theorem for each exponent $n \neq 2$. For $n = 2$, it implies Ramsey's theorem and does not imply ACA_0. For each exponent, we will also discuss the arithmetical complexity of solutions to computable instances of this packed Ramsey's theorem.

Structures presentable by various types of automata

Ekaterina Fokina[1] and Pavel Semukhin[2]

[1] Kurt Gödel Research Center, University of Vienna, Austria
efokina@logic.univie.ac.at
[2] Department of Computer Science, University of Regina, Canada
semukhip@cs.uregina.ca

Keywords: computable structures, automatic structures, asynchronous presentation, pushdown automaton, decidable theory

We consider only countable structures in a finite relational language. Computable structures have the universe and the basic functions and relations recognizable by Turing machines. Considering structures with universes and basic relations recognizable by synchronous finite automata, we get an important and well studied subclass of computable structures — that of automatic structures [1]. One of the most important algorithmic properties of automatic structures is that they always have a decidable presentation, i.e., a presentation with the complete diagram being decidable.

One can also define intermediate subclasses of structures presentable by other types of automata. In particular, an asynchronously automatic structures has a regular set as its domain, and its basic relations are recognizable by asynchronous multi-tape automata [1]. Asynchronously automatic structures need not be decidable [2] but various interesting structures without an automatic presentation have an asynchronous automatic presentation.

An even wider class would be a class of structures presentable by pushdown automata. In the talk we will give possible approaches to define such structures.

We will discuss motivation, some examples, open problems that arise in the field. Some parts of this work are new to us and several open question are rather basic.

References

1. B. Khoussainov, A. Nerode, *Automatic presentations of structures*, LCC: International Workshop on Logic and Computational Complexity. Lecture notes in Computer Science, 960 (1995) 367–395.
2. W. Thomas, *A Short Introduction to Infinite Automata*, Proceedings of the 5th international conference Developments in Language Theory. Lecture Notes in Computer Science, 2295 (2002) 130–144.

Ultrametric versus Archimedean automata *

Rūsiņš Freivalds

Institute of Mathematics and Computer Science, University of Latvia,
Raiņa bulvāris 29, Riga, LV-1459, Latvia

Absolute value of rational number x is called *trivial* if it equals 0 for the number 0, and eqals 1 for all the other numbers.

For a prime number p, the p-adic absolute value on Q is defined as follows: any non-zero rational x, can be written uniquely as $x = p^n \frac{a}{b}$ with a, b and p pairwise coprime and $n \in \mathbb{Z}$ some integer; so we define

$$|x|_p := \begin{cases} 0, & \text{if } x = 0 \\ p^{-n}, & \text{if } x \neq 0. \end{cases}$$

In 1916 Alexander Ostrowski proved that any non-trivial absolute value on the rational numbers Q is equivalent to either the usual real absolute value or a p-adic absolute value.

Distances using the usual absolute value are called *Archimedean*, and the distances using p-adic absolute values are called *ultrametric*. P.Turakainen [2] proved that probabilistic automata can be generalized using arbitrary real numbers instead of probabilities and the languages recognized by these Archimedean automata are the same stochastic languages.

We generalize probabilistic automata in the same way, only we use arbitrary p-adic numbers numbers instead of probabilities. Complexity of *ultrametric* automata defined this way can differ from complexity of probabilistic automata essentially. For arbitrary prime p there is a language L_p such that there is a p-ultrametric automaton with $2p$ states recognizing L_p but every deterministic finite automaton for L_p needs at least $c^{p \log p}$ states. M.O.Rabin proved in [1] that error-bounded probabilistic automata with k can be simulated by deterministic automata with c^k states.

References

1. Michael O. Rabin. Probabilistic Automata. *Information and Control,* vol. 6, No. 3, pp. 230–245, 1963.
2. Paavo Turakainen. Generalized Automata and Stochastic Languages. *Proceedings of the American Mathematical Society,* vol. 21, No. 2, pp. 303–309, 1969.

* The research was supported by Project 2009/0216/1DP/1.1.1.2.0/09/IPIA/VIA/044 from the European Social Fund.

Digital Computation as Instructional Information Processing

Nir Fresco

School of humanities, UNSW, Sydney, Australia
Fresco.Nir@Gmail.com

Abstract. It is common in cognitive science to equate digital computation with information processing, for information processing is (supposedly) what digital computers do. This second statement is put to the test in this paper, which deals with digital computation as it is actualised in physical systems. I shall examine in what sense information processing is *equivalent* to digital computation and whether nontrivial digital computing systems can be distinguished from non-computing systems by virtue of information-processing features. The answers depend on what we take 'information' to be and what the *processing* of information is.

Information may be interpreted non-semantically (e.g., as either Shannon information or algorithmic information) or semantically (e.g., as either factual information or instructional information). Undoubtedly, other interpretations of information exist, but they are not discussed in this paper. To set the stage, the processing of information is characterised here as the production of new information, its modification and the removal thereof. I argue elsewhere that whilst algorithmic information fares much better than Shannon information as a candidate for a plausible information processing account of computation, the resulting account still faces problems. The focus of this paper is on semantic information and more specifically on instructional information.

Arguably, an instructional information processing account can adequately explain concrete digital computation. According to this account, concrete digital computation is the processing of digital data in accordance with finite instructional information for some purpose. The instructional information processing account has the suitable conceptual backbone to explain the essential features of digital computing systems in information-processing terms. It also does not have the problematic implication, which factual information carries, that digital computing systems can either evaluate or ensure the truthfulness of the information that they process. Further, it is evaluated favourably against the criteria for assessing the adequacy of accounts of concrete computation.

Keywords: Instructional Information, Factual Information, Concrete Digital Computation, Turing Machines, Cognitive Science, Digital Computers

Recursive Marriage Theorems and Reverse Mathematics

Makoto Fujiwara[1,2], Kojiro Higuchi[1], and Takayuki Kihara[1]

[1] Mathematical Institute, Tohoku University,
6-3, Aramaki Aoba, Aoba-ku, Sendai, Miyagi, Japan
[2] sb0m29@math.tohoku.ac.jp

Abstract. Countable marriage theorem (Hall, 1948) asserts that a countable bipartite graph $(B, G; R)$ satisfying B-locally finiteness (each vertex in B has a finite degree) and Hall condition (for all finite subset $X \subset B$, the number of vertices which are adjacent to X is more than or equal to $|X|$) has a solution (injection $M \subset R$ from B to G). However it does not hold recursively, namely, even if a bipartite graph $(B, G; R)$ satisfying B-locally finiteness and Hall condition is recursive, it does not have a recursive solution. Reverse mathematics reveals how far each theorem is from holding recursively. By a previous research, it has been known that countable marriage theorem is equivalent to ACA over RCA_0, and if the recursiveness is added to B-locally finiteness in its assumption, the assertion weakens to be equivalent to WKL over RCA_0 (Hirst, 1990).
On the other hand, Kierstead introduced expanding Hall condition which is a combinatorial expansion of Hall condition, and showed that a recursive bipartite graph $(B, G; R)$ satisfying recursive B, G-locally finiteness and recursive expanding Hall condition, has a recursive solution [1]. This suggests that a restricted marriage theorem with recursive B, G-locally finiteness and recursive expanding Hall condition as assumption, is provable in RCA_0. (Indeed this is the case.) Then we consider all possible marriage theorems with various levels of B-locally finiteness, G-locally finiteness and Hall condition as assumption, and classify them into ACA_0, WKL_0 and $\mathsf{RCA}_0(+\Sigma^0_3\text{-}\mathsf{IND})$ in the context of reverse mathematics. By this exhaustive investigation, we can see how the strength of marriage theorem varies by strengthening its assumption step by step, and consequently find that recursive G-locally finiteness and recursive expanding Hall condition are essential for a recursive bipartite graph to have a recursive solution. Moreover we also carry out the same investigation for the symmetric variation of countable marriage theorem, and classify restricted symmetric marriage theorems into ACA_0, WKL_0 and RCA_0 completely.

Keywords: Reverse mathematics, Recursive graph theory, Marriage theorem.

References

1. H. A. Kierstead, An effective version of Hall's theorem, *Proc. Amer. Math. Soc.* **88** (1983), pp. 124–128.

A simple and complete model theory for intensional and extensional untyped λ-equality

Michael Gabbay[1]

Department of Philosophy, Kings College London

Abstract. We discuss a (relatively) new method of providing models of λ-reduction by which we are able to interpret λ-terms compositionally on possible world structures with a ternary accessibility relation. The simplicity of the structures is striking, moreover, they provide us with a surprising richness of interpretations of function abstraction and application.

We show how the models can differentiate between 'extensional' λ-reduction, which supports β-contraction and η-expansion, and 'intensional' reduction which supports only β-contraction. We state semantic characterisation (i.e. completeness) theorems for both.

We then show how to extend the method to provide a sound and complete class of models for reduction relations that additionally support β-expansion and η-contraction (i.e. they are models of β-equality and η-equality). The class of models is complete in the sense that exactly the derivably equal λ-terms receive the same denotation in every model.

The models we present differ from the familiar models of the λ-calculus as they can distinguish, semantically, between intensional and extensional λ-equality.

The enumeration degrees of semi-recursive sets

Hristo Ganchev*

Sofia University, Faculty of Mathematics and Informatics
5 James Bourchier blvd., 1165 Sofia, Bulgaria
ganchev@fmi.uni-sofia.bg

The pairs of sets, Kallimulin [2] used for showing the definability of the jump operation in the structure of the enumeration degrees, are a generalization of Jockusch's notion of semi-recursive sets [1].

In this talk we shall investigate the properties of the enumeration degrees of the semi-recursive sets. We shall give an alternative first order defininition of the jump operation in the enumeration degrees. Further we shall generalize the result by Giorgi, Sorbi and Yang [3] by proving that every non-low Σ_2^0 enumeration bounds a downwards properly Σ_2^0 degree.

References

1. C. G. Jockusch, *Semirecursive sets and positive reducibility*, Trans. Amer. Math. Soc. **131** (1968), 420–436.
2. I. Sh. Kalimullin, *Definability of the jump operator in the enumeration degrees*, Journal of Mathematical Logic **3** (2003), 257–267.
3. M. Giorgi, A. Sorbi, and Y. Yang, *Properly Σ_2^0 enumeration degrees and the high/low hierracrhy*, J. Symbolic Logic **71** (2006), 1125–1144.

* The author was partially supported by BSNF through contract DMU03-7/12.12.2011.

Generating DNA Code Words Using Forbidding and Enforcing Systems

Daniela Genova[1] and Kalpana Mahalingam[2]

[1] Department of Mathematics
University of North Florida
Jacksonville, FL 32224, USA
d.genova@unf.edu

[2] Department of Mathematics
Indian Institute of Technology
Chennai, TN 600036, India
kmahalingam@iitm.ac.in

DNA code words arose in the attempt to avoid unwanted hybridizations of DNA strands for DNA based computations. Given a set of constraints, generating a large set of DNA strands that satisfy the constraints is a difficult and important problem in DNA computing. On the other hand, motivated by the non-determinism of molecular reactions, A. Ehrenfeucht and G. Rozenberg introduced forbidding and enforcing systems (fe-systems) as a model of computation that defines classes of languages based on two sets of constraints. We connect these two areas of research in natural computing by proposing a new way of generating DNA codes using the theory of forbidding and enforcing systems. Using fe-systems, we show how families of DNA codes that avoid unwanted cross hybridizations can be generated. Since fe-systems impose restrictions on the subwords or words of a language, they can be used to model the restrictions imposed by unwanted hybridizations and thus, provide a natural framework to study DNA codes.

One of the benefits of studying DNA codes with fe-systems defining families of languages is that one fe-system can define an entire class of DNA codes as opposed to just one DNA code that has been the norm in constructing/studying DNA code words using the classical formal language theoretic approach. We characterize θ-subword-k-m, θ-subword-k, θ-strict, θ-prefix, θ-suffix, θ-bifix, θ-infix, θ-intercode, θ-comma-free and θ-k codes for a morphic or an antinmorphic involution θ by fe-systems. Examples of generating specific DNA codes over the DNA alphabet by fe-systems are presented.

We confirm some properties of DNA codes using fe-systems, which exhibits the potential of the fe-systems approach to DNA codes for discovering new properties of DNA codes using forbidding-enforcing theory.

Finally, we show how some known methods for generating good DNA codes, which have been tested experimentally, can be generalized using fe-systems. This shows that fe-systems can be used to extend these methods and to generate large families of codes with a desired property, as opposed to just one code.

Complexity of the Continued Fractions of Some Subrecursive Real Numbers

Ivan Georgiev

Burgas Prof. Assen Zlatarov University,
Faculty of Natural Sciences,
Prof. Yakimov Str. 1, 8010, Burgas, Bulgaria,
ivandg@yahoo.com

Abstract. A theorem, proven in my master's thesis [1] states that a real number is $\mathcal{E}^2$-computable, whenever its continued fraction is in $\mathcal{E}^2$ (the third Grzegorczyk class). The aim of this presentation is to settle the matter with the converse of this theorem. It turns out that there exists a real number, which is $\mathcal{E}^2$-computable, but its continued fraction is not primitive recursive, let alone $\mathcal{E}^2$. A question arises, whether some other natural condition on the real number can be combined with $\mathcal{E}^2$-computability, so that its continued fraction has low complexity. Such a condition is, for example, $\mathcal{E}^2$-irrationality. A close scrutiny of a theorem of R. S. Lehman [6] shows that if a real number is $\mathcal{E}^2$-computable and $\mathcal{E}^2$-irrational, then its continued fraction is in $\mathcal{E}^3$ (elementary). Using some strong results on irrationality measures, we conclude that all real algebraic numbers and the number π have continued fractions in $\mathcal{E}^3$.

Keywords: Grzegorczyk's classes $\mathcal{E}^2$ and $\mathcal{E}^3$, $\mathcal{E}^2$-computable real number, $\mathcal{E}^2$-irrationality, continued fractions

References

1. Georgiev, I., Subrecursive computability in analysis, MSc Thesis, Sofia University, 2009. http://www.fmi.uni-sofia.bg/fmi/logic/theses/georgiev.htm
2. Grzegorczyk, A., Some classes of recursive functions, Dissertationes Math. (Rozprawy Mat.), 4, Warsaw, 1953.
3. Calude, C., Super-exponential non primitive recursive, but rudimentary, Information Processing Letters 25 (1987), 311-315.
4. Hata, M., Improvement in the irrationality measures of π and π^2, Proc. Japan Acad. Ser. A. Math. Sci. 68 (1992), 283-286.
5. Khinchin, A. Ya., Continued fractions, Dover Publications, 1997.
6. Lehman, R. S., On primitive recursive real numbers, Fundamenta mathematicae, vol. 49 (1961), 105-118.
7. Skordev, D., Computability of real numbers by using a given class of functions in the set of natural numbers, Math. Log. Quart., vol. 48 (2002), suppl. 1, 91-106.
8. Skordev, D., $\mathcal{E}^2$-computability of e, π and other famous constants, Electronic Notes in Theoretical Computer Science, vol. 14 (2008), 861-875.

On modernisation of historical texts

Stefan Gerdjikov * e-mail:st_gerdjikov@abv.bg

Sofia University "Sv.Kliment Ohridski",
5 James Bourchier blvd. 1164, Sofia, Bulgaria

Abstract. The digitisation of historical texts provides an access to a huge amount of the human knowledge and experience of our ancestors. However the spelling variations and the lack of norms make it difficult for practical usage, e.g. information retrieval and information extraction. The comprehension is also hindered for non-experts.

In this talk we address the problem of *normalisation* of historical texts. Specifically we focus on determining the modern equivalent m of a given historical word h. The approach we present is based on a modern dictionary $\mathcal{D}$, i.e. list of modern words, and a list of instances $\mathcal{I} = \{\langle h_i, m_i \rangle\}_{i=1}^N$ which specifies the modern equivalent m_i of the historical word h_i. Given this input data, our objectives are:

1. to determine a set of patterns $\mathcal{P} = \{\langle \chi, \mu \rangle\}$ and $p : \mathcal{P} \to (0; 1]$, so that $p(\chi, \mu)$ expresses how probable it is that an infix χ of a historical word is transformed into an infix μ of a modern word.
2. for an arbitrary historical word h provide a sorted list of candidates $\mathcal{C}(h)$ of its possible modern equivalents.

Once the first problem is solved, the second one is modelled via a Levenshtein edit-distance with edit-operations $\mathcal{P}$ and cost function p. So the challenging problem seems to be the first one. Our idea is to retrieve the desired information by embedding the *structure* of the historical words $\mathcal{H} = \{h_i\}_{i=1}^N$ into the structure of the modern words $\mathcal{D}$. We use the structure introduced by Blumer and Blumer et al. in 1984, 1987:

$$\mathcal{S}_{\mathcal{H}} = \{\chi \,|\, \chi \text{ is a prefix of } \mathcal{H} \text{ or there are characters } a \neq b (a\chi \text{ and } b\chi \text{ are infixes of } \mathcal{H})\}.$$

Similarly we define the structure $\mathcal{S}_{\mathcal{D}}$. Intuitively $\mathcal{S}_{\mathcal{H}}$ captures the interesting regularities in the set of historical words $\mathcal{H}$ and $\mathcal{S}_{\mathcal{D}}$ represents the regularities in the modern language and the task is to determine how the regularities from the first structure are transformed into the second. Thus we search $\mathcal{P}$ as a subset of $\mathcal{S}_{\mathcal{H}} \times \mathcal{S}_{\mathcal{D}}$. For each historical word $h_i \in \mathcal{H}$ we define a binary tree $\mathcal{T}_{\mathcal{H}}(h_i)$ which encodes the decomposition of h_i into smaller subwords with respect to the structure $\mathcal{S}_{\mathcal{H}}$. An *augmented binary tree* $\mathcal{AT}_{\mathcal{D}}(m_i)$ is assigned to each modern word. Then using a somewhat modified Levenshtein edit-distance allows us to find the best way to embed the structure $\mathcal{T}_{\mathcal{H}}(h_i)$ into the structure $\mathcal{AT}_{\mathcal{D}}(m_i)$ for each instance $\langle h_i, m_i \rangle \in \mathcal{I}$. Based on these embeddings we determine the set $\mathcal{P}$ and gather statistics in order to define the probability measure p.

We conclude by presenting some experimental results in favour of this approach and discussing further challenges in this area.

Key Words: Edit-distance, Approximate search, Alignments of structures.

* The work on this presentation was granted by FP7 CULTURA project, no. 269973.

Sequent calculus with algorithmic properties for logic of correlated knowledge

Haroldas Giedra and Jūratė Sakalauskaitė

Vilnius University, Institute of Mathematics and Informatics, Akademijos str. 4,
LT-08663 Vilnius, Lithuania,
haroldas.giedra@mii.vu.lt, jurate.sakalauskaite@mii.vu.lt

Abstract. Logic of correlated knowledge is epistemic logic enriched by observational capabilities of agents. This allows to formalize and reason about correlations between information carried by systems in different states. As an example of such information is quantum information carried by physical system composed of atomic particles. Sequent calculus is one type of deduction systems used to mechanically prove the truth of statements related to corresponding knowledge base. We presented sequent calculus for logic of correlated knowledge, proved soundness, completeness, admissibility of cut rule and analysed the termination of the system in this paper. As a method, we used internalization of semantics to rules and axioms of deduction system.

Keywords: Logic of correlated knowledge, sequent calculus, soundness, completeness, admisibility of cut rule, termination.

Approximating Asymmetric Metric TSP in Exponential Time

Alexander Golovnev

St. Petersburg University of the Russian Academy of Sciences, St. Petersburg, Russia
alex.golovnev@gmail.com

The fastest known exact algorithm for the Traveling Salesman Problem (TSP) uses $O^*(2^n)$ time and space [Held, Karp] (here, $O^*(\cdot)$ suppresses polynomial factors of the input length that is poly$(n, \log M)$, where n is the number of vertices in the input graph and M is the maximal edge weight). If we allow a polynomial space only, then the best known algorithm has running time $O^*(4^n n^{\log n})$ [Björklund, Husfeldt]. For TSP with bounded weights there is an algorithm with $O^*(1.657^n \cdot M)$ [Björklund] running time. It is a big challenge to develop an algorithm with $O^*(2^n)$ time and polynomial space.

In this short note we propose a very simple algorithm that, for any $\varepsilon > 0$, finds $(1+\varepsilon)$-approximation to directed metric TSP in $O^*(2^n\varepsilon^{-1})$ time and $O^*(\varepsilon^{-1}) = \varepsilon^{-1} \cdot \text{poly}(n, \log M)$ space. Thereby, for any fixed ε, the algorithm needs $O^*(2^n)$ steps and polynomial space to compute $(1+\varepsilon)$-approximation.

Our algorithm is inspired by FPTAS for the Knapsack problem due to Ibarra and Kim. They use the fact that Knapsack can be solved by a simple pseudopolynomial algorithm working in time $O(nW)$ where n is the number of items and W is the total weight. FPTAS for Knapsack first divides all input weights by some $\alpha(\varepsilon, n, W)$ and then invokes the pseudopolynomial algorithm. The resulting answer might not be optimal as the weights are not just divided by α, but also rounded after that. However a simple analysis shows that rounding does not affect the result too much.

We use the same idea. By OPT we denote the cost of an optimal solution. To get a polynomial-space approximation algorithm for directed Metric TSP we first divide all edge weights by a big enough number α and then use an algorithm based on inclusion-exclusion. Then the running time of inclusion-exclusion algorithm is $2^n \cdot \text{OPT}/\alpha$ and the length of the resulting cycle is at most $\text{OPT} + \alpha n$. So, we choose α s.t. $\alpha \geq \frac{\text{OPT}}{\text{poly(n)}}$ and $\alpha \leq \frac{\varepsilon \text{OPT}}{n}$. Metric TSP can be approximated in polynomial time, therefore we can find β s.t. $\text{OPT} \leq \beta \leq \text{OPT} \cdot \log n$ and take $\alpha = \frac{\beta \cdot \varepsilon}{n \log n}$. Then the algorithm has $O^*(2^n\varepsilon^{-1})$ time and $O^*(\varepsilon^{-1})$ space complexity, respectively.

Uncomputable Games: Toward Crowd-sourced Solving of Truly Difficult Problems

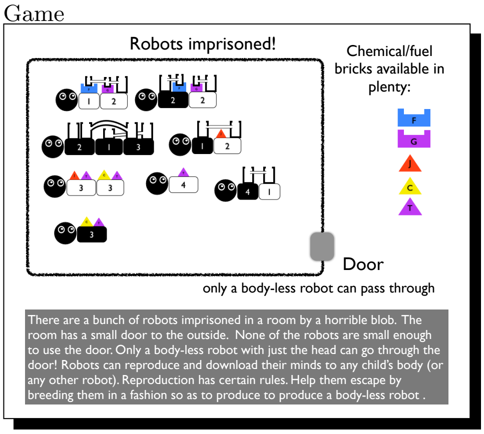
Naveen Sundar Govindarajulu

Computer Science, Rensselaer Polytechnic Institute, Troy NY 12180 USA
govinn@rpi.edu

Keywords: Uncomputable games, the Robot Devolution game, Resolution theorem proving

Given the huge rise in casual gaming in the last decade and the amount of human-hours put into gaming, quite a few researchers have sought to harness gaming labor for computationally hard tasks, for example, labelling images. So far all such tasks fall into the Turing-solvable realm. In this talk, a game which reduces to, and is reducible from, a Turing-unsolvable problem will be presented.

Fig. 1. An Instance of the Robot Devolution Game



Many important computational problems which arise in science and technology are computationally hard. For example, computational abstractions of the protein folding problem are known to be NP-complete, which renders the problem intractable by existing means if $\mathsf{NP} \neq \mathsf{P}$. Certain other problems are still harder, as they are Turing-unsolvable. First-order formalizations of problems such as formal verification of circuits, software correctness checking, theorem-proving, type inference etc. fall into this realm (in the general case). Some of these problems have immediate and wide-ranging economic consequences; for example, estimates of costs to the global economy due to detrimental effects of software bugs range in the tens to hundreds of billions of dollars annually. This problem does not have a straightforward solution, as formal verification of software is a very labor-intensive task. Exacerbating the situation is the lack of skilled labor: DARPA estimates that there are only about 1000 formal-verification experts in the US. Most instances of uncomputable problems, including formal verification of software, seem to require insight into the structure of the problem. This insight is possessed mostly by the experts of that domain. This leads to the question of whether such hard problems can be cast into a form, for example a natural game, which removes the need for domain knowledge by completely abstracting away from the original domain. Ideally, such a game should be able to exploit a human player's insight into the structure of the game. This insight should then be automatically transferrable into solving the problem. Some definitions are in order:

Game Define a game $\mathcal{G}$ to be a quadruple $\langle G, \gamma : G \mapsto 2^G, I, F\rangle$ composed of a set of game states G, an operation γ specifying valid moves in the game, a set of initial game states I, and a set of final game states F.[1] The player starts with an initial game state $i \in I$. Initial game states are also called *game instances*. The player produces a sequence of game states $i, g_1, g_2, \ldots$. If there is a finite sequence starting with $i \in I$ and ending with an $f \in F$, the game instance i is said to be *completable*; if there is no such sequence for i, then the game instance is said to be *uncompletable*.

Uncomputable Game A game $\mathcal{G} = \langle G, \gamma : G \mapsto 2^G, I, F\rangle$ is said to be an uncomputable game if there is no Turing machine m which decides for any arbitrary game instance $i \in G$ whether it is completable or uncompletable.

In CiE 2012, we will present the *Robot Devolution Game* (see Figure 1), an uncomputable game, which was designed to abstract first-order resolution theorem proving. The mapping from resolution, initial results, and videos from subjects playing the game will be presented during the talk.[2] A small instance of the game will be given to the audience to play live during the presentation.

---

[1] Note: G, I and F need not be finite.

[2] Thanks to Selmer Bringsjord for invaluable discussions, Konstantine Arkoudas and Marc Destafano for their comments, and Spencer Posson for helping with the implementation of the Robot Devolution game in the iOS platform. See http://www.cs.rpi.edu/~govinn/papers/RobotDevolution.pdf for the mapping.

Structure Diagrams in Type Theory

Henson Graves

Algos Associates
2829 West Cantey Street
Fort Worth, TX 76109 United States
henson.graves@hotmail.com

Structures consisting of components with part and connections between them occur frequently in science and engineering. The graphical syntax of SysML provides a well-developed language for specifying descriptions of structures with part and connection properties. An abstraction of structure diagrams are defined in which each property within a structure diagram has a domain and range. Axioms are given for part properties within a structure diagram which ensure that the part properties have a strict partial order. The axioms for part properties in a structure digram are syntactically checkable. An orthogonality axiom is used to ensure that in an implementation parts of the same type do not get reused. The decidability of the consistency of a structure diagram is established.

While Description Logics (DLs)have been used to model structure diagrams, there is no known DL corresponding to a structure diagram or a DL generated by a structure diagram. Further, the structure diagrams in engineering languages use operations, variables, behavior constructions, as well as parts. A structure diagram is embedded as an axiom set within a variant of type theory, called Algos. Algos is based directly on topos axioms. The embedding makes use of the fact that type theory contains a representation of Description Logic concept and role constructions. Type theory provides a direct inference semantics for the DL constructions. Type theory significantly extends the language constructions of DL and other type theory constructions, such as an equalizer type construction, are used to represent structure diagrams.

The decidability of the consistency is established using type theory properties to eliminate quantifiers to reduce the formula representation for a structure diagram to one which uses only a single universally quantified formula with monadic predicates. In type theory a functional property can be replaced with a map term which whose value for each argument satisfies the functional property. Since there are only a finite number of connection equations each connection equation can be replaced by a finite number of unary predicates. Conditions are given on structure diagrams which ensure that each structure satisfying the axioms has a unique part decomposition, i.e., all minimal models are isomorphic.

Keywords: Description Logic, SysML, OWL, Type Theory, Structure Diagrams

Quantum Secret Sharing with Graph States

Sylvain Gravier[1,2], Jérôme Javelle[3], Mehdi Mhalla[1,3], and Simon Perdrix[1,3]

[1] CNRS
[2] Institut Fourier, University of Grenoble, France
[3] LIG, University of Grenoble, France

Keywords: Quantum Information, Secret Sharing, Graph Theory, NP-completeness, Probabilistic Methods.

A threshold $((k, n))$ quantum secret sharing protocol [6, 2, 3] is a protocol by which a dealer distributes shares of a quantum secret to n players such that any subset of at least k players can access the secret, while any set of less than k players has no information about it. We investigate a particular family of quantum secret sharing protocols parametrized by a graph which represents the quantum state, called graph state, shared by the players [8, 7]. We show that the graph theoretical notion of weak odd domination can be used to characterize the sets of players that can access the secret.

Only few threshold quantum secret sharing schemes have been proved in the literature to be achievable using graph states: $((3, 5))$ can be done using a C_5 graph (cycle with 5 vertices) [8], and for any n, an $((n, n))$ protocol using the complete graph can be done, up to some constraints on the quantum secret [9]. Independently [1] introduced an $((n, n))$ protocol for any n. This protocol is based on the GHZ state which is locally equivalent to a complete graph state [5].

Using a construction based on lexicographic product of graphs, we introduce a family of graphs which can achieve any $((k, n))$ protocol when $k > n - n^{0.68}$. Moreover, we prove that for any graph the corresponding secret sharing protocol has a threshold greater than $\frac{79}{156}n$ where n is the number of players of the protocol. Using probabilistic methods we prove the existence of graphs which threshold is smaller than $0.811n$. More precisely, we show that a random graph of order n has a threshold smaller than $0.811n$ with high probability. However, one cannot easily double check that the threshold of the protocol associated with a particular (randomly generated) graph is actually smaller than $0.811n$ since we prove that the corresponding decision problem is NP-complete.

The presentation will be mainly based on the results presented in [7] and [4].

References

1. A. Broadbent, P. R. Chouha, and A. Tapp *The GHZ state in secret sharing and entanglement simulation. arXiv:0810.0259*, 2008.
2. R. Cleve, D. Gottesman, and H.-K. Lo, *How to Share a Quantum Secret Phys. Rev. Lett. 83, 648-651,* 1999.
3. D. Gottesman, *On the Theory of Quantum Secret Sharing. Phys. Rev. A* 61, 04231. Also quant-ph/9910067, 2000.
4. S. Gravier, J. Javelle, M. Mhalla and S. Perdrix. *On Weak Odd Domination and Graph-based Quantum Secret Sharing* arXiv:1112.2495 , 2011.
5. M. Hein, J. Eisert, and H. J. Briegel. *Multi-party entanglement in graph states. Physical Review A*, 69, 062311 quant-ph/0307130, 2004.
6. M. Hillery, V. Buzek and A. Berthiaume. *Quantum Secret Sharing Phys. Rev. A 59, p.1829; arXiv/9806063*, 1999.
7. J. Javelle, M. Mhalla, and S. Perdrix. New protocols and lower bound for quantum secret sharing with graph states. In *Theory of Quantum Computation, Communication and Cryptography. (TQC12)*, To appear in Lecture Notes in Computer Science, 2012. *eprint:arXiv:1109.1487*, 2012.
8. D. Markham and B. C. Sanders. *Graph states for quantum secret sharing. Physical Review A* 78, 042309, 2008.
9. D. Markham and B. C. Sanders. *Erratum: Graph states for quantum secret sharing [Phys. Rev. A 78, 042309 (2008)] Phys. Rev. A* 83, 019901(E), 2011.

Systems and Methods for Data and Algorithm Protection

Marc Groz[1]
[1]TransIDEA®, Stamford, CT, USA
marcgroz@gmail.com

Abstract. Systems and methods for the protection of sensitive data and/or algorithms are presented. One such system makes use of rotation of coordinates by a random vector, allowing for the protection of the input vectors associated with the calculation of a correlation coefficient. This system may be used to protect multiple such calculations, as for example, may be associated with a correlation matrix. Applications to financial systems, systemic risk, and cloud computing are explored.

Keywords: cryptography, cybersecurity, infosecurity, cloud, systemic, risk, financial, correlation, random, matrix

Self Developing Networks

Frederic Gruau

Laboratoire de Recherche en Informatique
Bât 650 Université Paris-Sud 11 91405 Orsay Cedex France
frederic.gruau@lri.fr http://blob.lri.fr

Résumé We present a work in two parts : [1] and [2]. The first part presents a formal system called "Self Developing Networks of agents" or SDN. It attempts to reproduce the quality of parallelism present in cellular biological development : development, communication, distribution and decentralized execution. Finite-state agents execute local graph rewriting rules that create other agents and links. An initial ancestor agent can thus progressively develop a network of agents. An obstacle to achieve a decentralized and distributed execution, happens when two neighbors try to simultaneously create new nodes connecting to each other. a first definition of SDNs called "basic SDNs" solve the problem by avoiding it : it imposes that when an agent can rewrite, its neighbors cannot, and can thus be used as stable gluing points. It is the programmer's task to prove mutual neighbor-exclusion.

We then extend the definition to a more general concept of higher order SDNs that alleviates this task : They are are node-replacement parallel graph grammars, endowed with the additional property that they can be simulated by basic SDNs. We illustrate those two definitions with graph grammar on undirected and directed graph, the simulation works like a software layer which incorporates the neighbor exclusion constraint at the syntactic level.

In the second part, Self Developing agents are programmed using a Finite State Automaton with output actions (Mealy machine). The FSA embodies the pure computational part, while the actions which are primitive node-replacement rule, encapsulate the pure development part and define the instructions of a Self Developing Machine (SDM). We present an example of SDM called the graph-machine using ten instructions, and prove that for any SDN, there exists a graph machine that simulates it. We introduce a natural model of complexity where an agent does a bounded processing in one unit of time. We prove an intrinsic universality result in linear time, while a simulation by a Turing Machine needs exponential time. This illustrates the parallel power of development.

Références

1. F. Gruau. Self developing networks, part 1 : the formal system. Technical Report 1549, LRI, 2012. http ://www.lri.fr/˜bibli/Rapports-internes/2012/RR1549.pdf.
2. F. Gruau. Self developing networks, part 2 : Universal machines. Technical Report 1550, LRI, 2012. submited to to UCNC 2012, http ://www.lri.fr/˜bibli/Rapports-internes/2012/RR1550.pdf.

Equilibriation of information in software systems

Les Hatton

CIS, Kingston University, UK
lesh@oakcomp.co.uk, http://leshatton.org/
School of Computing and Information Systems,
Penrhyn Road, Kingston on Thames, KT1 2EE

Abstract. This paper strengthens a recent result concerning the conservation of information in software systems. Using statistical mechanical arguments, a previous paper proved that subject only to the constraints of constant size and constant Shannon information, all software systems are overwhelmingly likely to evolve such that the probability of appearance of a component of n programming tokens obeys a power-law in the unique alphabet of tokens used, asymptoting to a constant for small components. Furthermore, this result is *independent* of application area, programming language or implementation details. In this paper, the result will be further strengthened and quantified using measurements on over 40 million lines of mixed open and commercial source code in seven programming languages. In addition, the rate at which equilibriation to the underlying distribution takes place will be demonstrated using animations on this corpus.

This result will then be generalised to consider the possible forms that defect distributions are most likely to take in software systems. This is of considerable importance because previous studies have only attempted to *fit* various assumed reasonable functions to existing data such as polynomial, logarithmic and hybrid schemes. However in order to understand the rate at which systems mature in defect terms, i.e. how and where defects appear with increasing usage, it is important to understand if there is any underlying natural functional form for this derivable from a mathematical model. Here a defect distribution derived directly from the power-law behaviour of component sizes is tested on real data.

Finally, these results will be used to explore the frequently observed phenomenon that defects in software systems *cluster*. This provides insights into the conditional probability of finding the (i+1)th defect in a component given that (i) defects have been found already, whatever kind of system is being built, whoever builds it and however it is built. It also throws a little light on the 'zero-defect' conundrum.

Keywords: Information content, conditional defect distribution, zero defect, Power-law, Equilibriation

Some constraints on the physical realizability of a mathematical construction

Francisco Hernández-Quiroz[1] and Pablo Padilla[2]

[1] School of Science, UNAM
Facultad de Ciencias, Ciudad Universitaria
D.F. 04510, MEXICO
fhq@ciencias.unam.mx
[2] Institute for Applied Mathematics, UNAM
IIMAS, Ciudad Universitaria
D.F. 04510, MEXICO
pablo@mym.iimas.unam.mx

Consider the question: given a mathematical construction and a particular physical system, is the latter adequate to "implement" the former? By implementation we mean an actual physical device that (a) has structural properties that correspond to components of the mathematical entity (some have talked about an isomorphism between physical and mathematical structures, but a weaker notion may also do); (b) a physical procedure that can produce experimental results which reflect accurately corresponding properties of the mathematical construction.

These are very intricate and hard questions to be answered definitely in a general case. Our aim is more modest, namely to explore a specific instance of this problem: we take the classical Cantor's diagonalization for the enumeration of the rational numbers and how it can be implemented by an Ising system. We provide a specific implementation and show its limitations deriving from properties of the physical system itself.

This leads us to think that some clearly defined mathematical questions cannot always be posed and answered within the context of a particular physical system. Of course, the more general question of the existence of a physical system realizing a particular mathematical construction is beyond the limits of this work but we hope our example helps to stimulate discussions on this line of thought. The standard interpretation of quantum mechanics regarding physically meaningful questions is that it should be possible to pose them in such a way that they can be answered experimentally.

The reciprocal question is also interesting: to what extent mathematical constructions should be considered valid? One possible approach, would imply that only those mathematical constructions that can actually be implemented by means of a physical system can in fact be used, at least in terms of computation.

keywords: Cantor's diagonalization, implementation, Ising system.

Slow and steady wins the race: relationship between accuracy and speed of Turing pattern formation

Ken-Ichi Hiraga[1], Kota Ikeda[2] and Takashi Miura[3]

[1] Kyoto University Faculty of Medicine
[2] Meiji University Graduate School of Advanced Mathematical Science
[3] Kyoto University Graduate School of Medicine

Abstract. Turing instability is extensively studied in the context of biological pattern formation [2]. However, several arguments against the application of Turing model has been raised. Here we examined one of these issues, the accuracy of pattern, and search for a condition to accurately generate periodic pattern in a linear regime. Firstly we showed that the system used by key reference by [1] is not within the diffusion-driven instability range. Next we showed the relationship between precision of pattern and shape of dispersion relation. The precision of pattern is partly determined by the range of unstable wavenumber K_{max}/K_{min}. We also showed the relationship between speed of pattern appearance λ_{max} and region of unstable mode. Possible strategy to generate accurate pattern and application to experimental results are discussed.

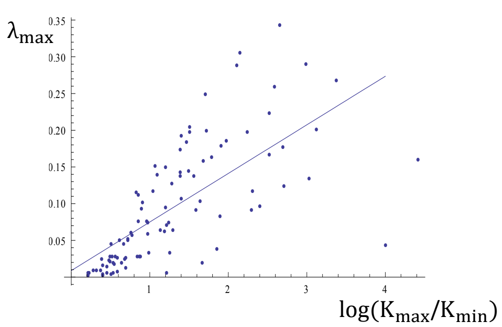


Fig. 1. The relationship between λ_{max} and region of unstable mode.

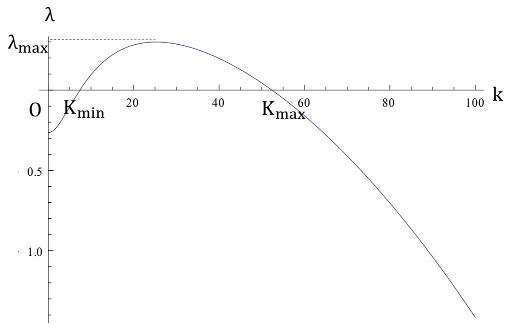


Fig. 2. Dispersion relation.

References

[1] Bard, J., Lauder, I.: How well does Turing's theory of morphogenesis work? Journal of Theoretical Biology 45(2), 501–531 (1974)
[2] Murray, J.: Mathematical Biology: Spatial models and biomedical applications (Jan 2003)

Computer Science Through Urn Games: An Unified Framework for a Hierarchy of Solvable and Unsolvable Problems

Dedicated to Alan Turing's 100th Birthday and to the memory of Professor Edsger W. Dijkstra

Sorin Istrail

Department of Computer Science

Brown University, Providence, RI 02912

In his last published paper, "Solvable and Unsolvable Problems," printed in 1954 in the popular journal Science News, Alan Turing presented several elegant puzzles aiming at explaining and popularizing problems for which there are algorithms for solutions -- "the solvable" -- as well as some for which no such algorithmic solution exists -- "the unsolvable." This paper could be seen as a continuation of his paper to popularize through puzzles a set of computational problems of various computational difficulties. Similar to his paper, where all his puzzles are unified as "substitution puzzles," our set of puzzles offers instances of a unified approach based on "urn games." Our (m, n1, n2) games have urns of two types: set urns and linear urns. E.g., the urns contain balls of m colors; in a move, a number n1 of balls are extracted and a number n2 of balls are returned to the urn; the solitary player performs moves, one after another, based on the rules of the game until no rule can be applied. Our urn games include the Turing substitutions puzzles, but defined with different goals. The class of computational problems obtained by varying the urn game parameters (m, n1, n2) turns out to form a hierarchy of complete problems, one for each of the complexity classes NL, P, NP, PSPACE, EXPSPACE, and "the unsolvable."

The urn games are generalizations of a "silly" game E.W. Dijkstra presented in his paper "Why correctness must be a mathematical concern" [Inaugural Lecture for the "Chaire Internationale d'Informatique" at the Universite de Liege, Belgium, 1979 (EWD720)]. His game is a (2,2,1) urn game in our framework. The generalizations are inspired by exciting correspondence I had with Professor Dijkstra in the wake of my somewhat critical comments of his game. [See my articles "Criticizing Professor Dijkstra Considered Harmless" and the followup, "When Professor Dijkstra Slapped Me in the Quest for Beautiful Code". http://www.cs.brown.edu/~sorin/non-tech-writing.htm. Dijkstra's silly game is shown to have a certain *incompleteness* property. This incompleteness relates to the apparent inseperability of two problems: (a) demonstrate how to predict the final outcome, and (b) demonstrate that the final outcome is completely predictable. It turns out that predictability is equivalent to associativity and to the existence of logical invariants for correctness proofs. We analyze the game and some natural variants inspired by the quest for understanding its incompleteness. It will turn out that a complementary problem, the \textit{Unpredictability of a given instance}, offers a pure combinatorial, (i.e., machine-independent) introduction of computational complexity classes. The game and its variants are disguises of decision problems of generic computational difficulty: Directed graph accessibility, Context-free grammar membership, Satisfiability of propositional Boolean formulas, Context-sensitive grammar membership, Uniform word problem for commutative semigroups, and Recursive-enumerable grammar membership. Indeed, the unpredictability problem for the game and its generalizations, turns out to provide us with a hierarchy of complete problem for the complexity classes NL, P, NP, PSPACE, EXPSPACE. The last mentioned disguise brings the complexity status of the problem to unsolvable.

Turing and von Neumann's Brains and their Computers

Dedicated to Alan Turing's 100th birthday and John von Neumann's 110th birthday

Sorin Istrail
Department of Computer Science, Center for Computational Molecular Biology
Brown University, Providence, RI 02912, USA
Solomon Marcus
"Simion Stoilow" Institute of Mathematics
Romanian Academy, Bucharest 010702, Romania

In this paper we discuss the lives and works of Alan Turing and John von Neumann that intertwined and inspired each other, focusing on their work on the brain. Our aim is to comment and to situate historically and conceptually an unfinished research program of John von Neumann, namely, towards the unification of discrete and continuous mathematics via a concept of thermodynamic error; he wanted a new information and computation theory for biological systems, especially the brain. Turing's work contains parallels to this program as well. We try to take into account the level of knowledge at the time these works were conceived while also taking into account developments after von Neumann's death. Our paper is a call for the continuation of von Neumann's research program, to metaphorically put meat, or more decisively, muscle, on the skeleton of biological systems theory of today.

In the historical context, an evolutionary trajectory of theories from Leibniz, Boole, Bohr and Turing to Shannon, McCullogh-Pitts, Wiener and von Neumann powered the emergence of the new Information Paradigm. As both Turing and von Neumann were interested in automata, and with their herculean zest for the hardest problems there are, they were mesmerized by one in particular: the brain. Von Neumann confessed: "In trying to understand the function of the automata and the general principles governing them, we selected for prompt action the most complicated object under the sun – literally."

Turing's research was done in the context of the important achievements in logic: formalism, logicism, intuitionism, constructivism, Hilbert's formal systems, S.C. Kleene's recursive functions and Kurt Gödel's incompleteness theorem. Turing's machine, exclusively built on the paper, as an abstract computing device, has been the preliminary theoretical step towards the programmable electronic computer. Turing's 1937 seminal paper, one of the most important papers in computer science, prepared the way for von Neumann's 1948 programmable computer.

Von Neumann's unfinished research program was outlined in his seminal articles "The general and logical theory of automata" (1951) and "Probabilistic logics and the synthesis of reliable organisms from unreliable components" (1956), his posthumous book *The Computer and the Brain* (1958) and the unfinished book *The Theory of Self-Reproducing Automata,* completed and published by A. Burks (1966). He proved in 1948, inspired by Turing's universal machine, part of his theory of self-reproduction of automata, five years before Watson and Crick, the structure of the DNA copying mechanism for biological self-reproduction. Biologist and Nobel Laureate Sydney Brenner, in his memoirs, acknowledges von Neumann's prophetic theorem: "You would certainly say that Watson and Crick depended on von Neumann, because von Neumann essentially tells you how it's done."

A Model of Computing as a Service for Cost Analysis

Kenneth Johnson[1] and John V. Tucker[2]

[1]Computer Science Research Group, Aston University,
Birmingham, B4 7ET
k.h.a.johnson@aston.ac.uk
[2]Department of Computer Science, Swansea University,
Swansea, Singleton Park, SA2 8PP
j.v.tucker@swansea.ac.uk

Abstract. Cloud computing can be succinctly described as *computing as a service* in which software, platforms and infrastructure are available from providers on a subscription basis such as pay-on-demand and pay-as-you-go. At the infrastructure level, *infrastructure services* are virtualised servers and storage devices purchased by the customer to host data and applications at the provider's data-centre.

The most attractive feature of cloud computing is its elastic nature, which enables customers to adapt service usage to suit variations in their computing requirements. A prominent factor of a customer's requirements is the *monetary costs* of the services, determined by the time duration the service is available (*service deployment*), and the volume of data handled by the service (*service usage*).

We present a model of computing as a service focused on quantifying monetary costs based on availability and usage and formulated as a function

$$Cost(\pi, t) = Cost_D(\pi, t) + Cost_U(\pi, t)$$

of the interaction π between provider and customer, where $Cost_D(\pi, t)$ is the cost of service deployment at the customer's request and $Cost_U(\pi, t)$ is the sum of costs of data handling across all services as scheduled by π over time t.

We develop an algebraic framework which models the virtual hardware infrastructure available for rental as abstract register machines, with memory and storage for any form of data and programs that define a non-terminating computation processing data in time. At the centre of our framework is a model of the customer's account enabling services to be chosen and made available over a time period. A model of the cost of these services is developed and we use computing service offers from Amazon Web Services to guide and illustrate the construction of the cost model.

The resulting framework is provider-independent and extendable to other forms of services. A special technical feature is the use of explicit clocks to model infrastructure offerings, index events and to measure time. The clocks are explicitly compared in different ways but the whole process of using a computing service is ultimately relative to physical *real-world time* as measured by a discrete clock T_W.

Languages Associated with Crystallographic Structures

N. Jonoska*, M. Krajcevski and G. McColm

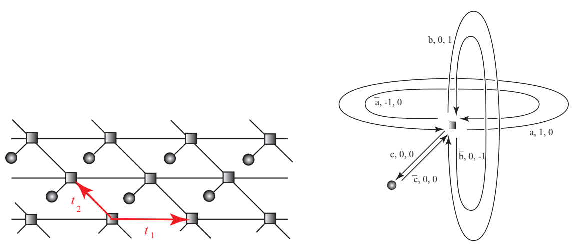
Department of Mathematics and Statistics,
University of South Florida, Tampa FL, 33620 USA
jonoska,mile@mail.usf.edu, mccolm@usf.edu

Crystals (periodic structures) can be associated with connected graphs embedded in Euclidean spaces. We model molecules by vertices in the graph, and bonds linking the molecules by non-oriented edges. The set of walks and the set of cycles uniquely determine the periodic graph up to isomorphism [3].

A graph is *periodic* if its automorphism group has a normal free abelian subgroup of finite index. Using this subgroup, we consider the quotient graph of the periodic graph by taking the orbits of the free abelian subgroup as vertices. The quotient graph is then used to form a finite state automaton that recognizes the language corresponding to the walks in the periodic graph. This regular language is known as strictly locally testable of order 2 [1].

If the finite state automaton is "decorated" with counters, then the set of all cycles in the periodic graph corresponds to an intersection of d context free languages, where d is the rank of the free abelian subgroup - we embed this graph in a d-dimensional Euclidean space so that the abelian group consists of translations. These machines are modeled after the PDA-like counter machines of [2], which employ counters rather than stacks. Following [4], we conclude that the set of words representing cyclic walks on a periodic graph in a d-dimensional Euclidean space is not the intersection of $d-1$ context free languages, so the languages of cyclic walks on periodic graphs provide examples of languages in this intersection hierarchy of context free languages. We believe that these counter machines more precisely capture the complexity of these languages.

Left: A 2-periodic (Euclidean) graph, with a translation subgroup generated by the translations t_1 and t_2. *Right:* Transition digraph of a DFA, with integers representing counter trnsitions, accepting the language of walks.



References

1. A. de Luca & A. Restivo, **Information & Control 44** (1980), 300 - 319.
2. O. Ibarra, **J. Assoc. Comput. Mach. 25** (1978), 116 - 133.
3. N. Jonoska & G. McColm, in: (LNCS 4135, 2006), 139 - 151.
4. L. Y. Liu & P. Weiner, **Math. Sys. Theory 7:2** (1973), 185 - 192.

* This work has been supported in part by the NSF grants CCF-1117254 and DMS-0900671.

Turing Machines, revisited

Joost J. Joosten

Department of Logic, History and Philosophy of Science
University of Barcelona, Carrer Montalegre, 6
08001 Barcelona, Spain
`jjoosten@ub.edu`

This informal presentation presents joint work with Hector Zenil and Fernando Soler Toscano. The content may best be described by the term experimental logic. We look at Turing machines (TMs) with a one-way infinite tape with just two tape-symbols and look at the fine-structure present is small Turing machines with for example, two, three or four states. There turns out to be surprisingly rich patterns in whatever aspect of what we call the universe of small Turing machines.

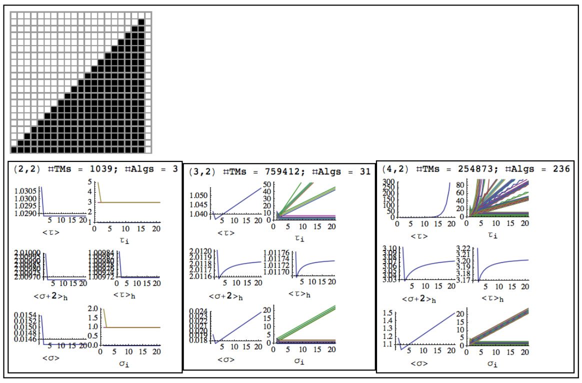


Fig. 1. A function computed by various Turing Machines with 2, 3 or 4 states.

In particular we will present and explain pictures like above. The upper square represents from top-row down, the tape-outputs by feeding a TM a tape with input just one black cell, two consecutive black cells, three, etc. So, effectively the TMs that compute this function just erase the right-most cell.

In the squares below we plot the runtimes (the τs) runtimes and space-usage (the σs) and some averages of TMs that compute this function. We do this for TMs with 2 colors and 2, 3 and 4 states respectively.

We will present our latest findings including a relation between the asymptotic Hausdorff dimension of the consecutive tape configurations on the one hand and asymptotic time complexity of the TM computation on the other hand.

Uniform polytime computable operators on analytic functions

A. Kawamura[1], N. Müller[2], C. Rösnick[3], M. Ziegler[3]

[1] University of Tokyo [2] Universität Trier [3] TU Darmstadt

Recursive analysis as initiated by Turing (1937) explores the in-/ computability of problems involving real numbers and functions by approximation up to prescribable absolute error 2^{-n}. Weihrauch's Type-2 Theory of Effectivity (TTE) extends this to mappings from/to the Cantor space of infinite binary strings encoding continuous universes in terms of so-called representations. Refining mere computability, classical complexity theory has successfully been transferred from the discrete to the real realm; cmp. e.g. Ker-I Ko (1991). However the common setting only covers real numbers x and (continuous) real functions f; operators $\mathcal{O}$ could be investigated merely in the non-uniform sense of mapping polytime computable functions to polytime ones — yielding strong lower bounds but weakly meaningful upper bounds for actual implementations of exact real number computation like `iRRAM`. As a major obstacle towards a uniform complexity theory for operators, the computable evaluation of a 'steep' function $f : x \mapsto f(x)$ requires more precision on x to approximate $y = f(x)$ than a 'shallow' one. More precisely as quantitative refinement of the sometimes so-called *Main Theorem* of recursive analysis, the (optimal) modulus of continuity of f constitutes a lower bound on its complexity — hence the evaluation operator cannot be computable on entire $C[0,1]$ in time bounded by the output precision n only.

In *Proc. 42nd Ann. ACM Symp. on Theory of Computing* (STOC 2010) one of the authors and his advisor have proposed and exemplified a structural complexity theory for operators $\mathcal{O}$ from/to continuous functions f on Cantor space — given by approximations as so-called regular string functions, that is, mappings $\varphi :\subseteq \{0,1\}^* \to \{0,1\}^*$ whose output length $|\varphi(\vec{\sigma})|$ depends only, and monotonically, on the input length $|\vec{\sigma}|$. They consider Turing machines converting such φ (given as a function oracle) and $\vec{\tau} \in \{0,1\}^*$ to $\mathcal{O}(\varphi)(\vec{\tau})$ in time uniformly bounded by a second-order polynomial in the sense of Kapron&Cook (1996) depending on both $|\vec{\tau}|$ and $|\varphi|$. For real operators, a reasonable (second-order) representation of functions $f \in C[0,1]$ as regular string functions φ amounts to $|\varphi|$ upper bounding f's modulus of continuity and renders evaluation (second-order) polynomial-time computable.

We further flesh out this theory by devising and comparing second-order representations and investigating the computational complexity of common (possibly multivalued) operators in analysis. Specifically, two rather different (multi-)representations are suggested for the space $\big\{(U, C, f|_C) : U \subseteq \mathbb{C} \text{ open}, C \subseteq U \text{ compact convex}, f : U \to \mathbb{C} \text{ analytic}\big\}$ and shown polytime equivalent. We then present second-order polytime algorithms computing some of the operators on this space that in (Ko 1991) had been shown non-uniformly polytime computable.

Second-Order Representations and Complexity of Compact Sets

Akitoshi Kawamura[1], Carsten Rösnick[2] and Martin Ziegler[2]

[1] University of Tokyo, Japan, kawamura@is.s.u-tokyo.ac.jp
[2] Darmstadt University of Technology, Germany,
{roesnick,ziegler}@mathematik.tu-darmstadt.de

In *Computable Analysis*, not only encodings of real numbers, functions and operators are considered, but also representations of subsets of, say, $\mathbb{R}^d$ for fixed dimension d. Although, in theory, the characteristic function χ_S uniquely represents a set S, it is a discontinuous function and therefore (using TTE [3]) non-computable for non-trivial S [3, Thm. 5.1.5]. However, using the topological properties of the metric space under consideration (X, for say), one *can* devise set-representations $f :\subseteq \{0,1\}^\omega \to \mathcal{A} := \{S \subseteq X \text{ closed}\}$:

- ω Asserts either $\overline{\mathrm{ball}}(\boldsymbol{a}, r) \cap S \neq \emptyset$, or $\overline{\mathrm{ball}}(\boldsymbol{a}, r) \not\subseteq S$ — [1, Def. 2.1.14];
- ψ^{dist} $[\rho^d_{\mathbb{R}} \to \rho_{\mathbb{R}\cup\{\infty\}}]$–representation of S' distance function d_S — [3, Def. 5.1.6];
- κ_G Enumerates for each $n \in \mathbb{N}$ vectors $(\boldsymbol{a}_{n,i})_i \subset \mathbb{D}^d_n$ s. t. the *Hausdorff-distance* between S and $\bigcup_i \overline{\mathrm{ball}}(\boldsymbol{a}_{n,i}, 2^{-n})$ is $\leq 2^{-n}$ — [3, Def. 7.4.1(3)], [4, Def. 2.2].

Those are essentially (but 'only' proven to be) *computationally* equivalent (cf. [3, Sec. 5], [5]), In [2], Kawamura and Cook introduced *second-order representations* $\tilde{f} :\subseteq (\{0,1\}^* \to \{0,1\}^*) \to \mathcal{A}$ (such $\tilde{f}$ can be constructed from a 'classical' representation f). In addition to $\tilde{\omega}$, $\tilde{\psi}^{\mathrm{dist}}$ and $\tilde{\kappa}_G$, consider:

- $\psi_{\circledcirc}$ Asserts either $\overline{\mathrm{ball}}(\boldsymbol{a}, 2r) \cap S \neq \emptyset$, or $\mathrm{ball}(\boldsymbol{a}, r) \cap S = \emptyset$ — [2, Sec. 4.2.1].

This notion of representation allows us to study mutual *second-order polytime translations* (i. e., (effective plus uniform) translations from one representation to another) of the above representations. For $X := [0,1]^d$ we get:

I. $\tilde{\kappa}_G \equiv_{\mathbf{poly}} \psi_{\circledcirc}$ w. r. t. any fixed polytime computable norm;
II. $\tilde{\psi}^{\mathrm{dist}}$ ($\equiv_{\mathbf{poly}} \delta_{\square}$ in this setting; [2, Sec. 4.3.1]) and $\psi_{\circledcirc}$ are (restricted to *convex* sets) polytime equivalent w. r. t. the *maximum-norm* (I. plus [4, Lem. 3.3]);
III. There is a polytime reduction from $\psi_{\circledcirc}$ to $\tilde{\omega}$ w. r. t. any fixed norm; but
IV. when restricting $\tilde{\omega}$ to *convex* sets and enriching $\tilde{\omega}$ with the center and radius of a $\overline{\mathrm{ball}}(\boldsymbol{a}, r) \subseteq S$, then there also is a polytime translation from $\tilde{\omega}$ to $\psi_{\circledcirc}$.

References

1. M. Grötschel, L. Lovász, A. Schrijver: Geometric Algorithms and Combinatorial Optimization, Springer, 1st ed. (1988)
2. A. Kawamura, S.A. Cook: Complexity Theory for Operators in Analysis, pp.495–502 in Proc. 42nd Ann. ACM Symp. on Theory of Computing (STOC 2010).
3. K. Weihrauch: Computable Analysis, Springer (2000).
4. X. Zhao, N.T. Müller: Complexity of Operators on Compact Sets, pp.101–119 in Electronic Notes in Theoretical Computer Science vol.202 (2008).
5. M. Ziegler: Computability on Regular Subsets of Euclidean Space, pp.157–181 in Mathematical Logic Quarterly vol.48 Suppl.1 (2002).

Σ-complete structures and universal functions

Asylkhan Khisamiev

Novosibirsk State University, 2 Pirogova Str., Novosibirsk, Russia
Sobolev Institute of Mathematics, 4 Acad. Koptyug avenue, Novosibirsk, Russia
hisamiev@math.nsc.ru

It is generally accepted presently that Σ-definability (generalized computability) in admissible sets is an important generalization of the concept of computability.

A crucial result of classical computability theory is the existence of an universal partially computable function. It is known (see [1]) that an universal Σ-predicate exists in every admissible set, but this is false for Σ-functions. It is proved in [1] that if $\mathfrak{M}$ is an algebraic system of a solvable model complete theory then an universal Σ-function exists in $\mathbb{HF}(\mathfrak{M})$. In [2] we constructed a torsion-free abelian group A such that no universal Σ-function exists in $\mathbb{HF}(A)$. In [3, 4] we introduced the concept of Σ-bounded algebraic system and obtained a necessary and sufficient condition for the existence of an universal Σ-function in the hereditarily finite admissible set over that system. We proved that every linear order, every Boolean algebra, and every abelian p-group are Σ-bounded systems, and that universal Σ-functions in the hereditarily finite admissible sets over them exist.

In this talk we introduce the notion of a Σ-complete structure. Using this notion we obtain the following results.

Theorem 1. *Let $\mathfrak{M}$ is a Σ-complete structure. Then in the hereditarily finite admissible set $\mathbb{HF}(\mathfrak{M})$ over $\mathfrak{M}$ there exists an universal Σ-function.*

Theorem 2. *If a structure $\mathfrak{M}$ is an abelian group of finite period or an almost bounded tree, then $\mathfrak{M}$ is Σ-complete structure.*

References

1. Ershov Yu. L.: Definability and Computability, Consultants Bureau. New York and London. (1996)
2. Khisamiev A. N.: On Σ-subsets of naturals over abelian groups. Siberian Math. J. **45** No. 3 (2006) 211–228
3. Khisamiev A. N.: Σ-Bounded algebraic systems and universal functions. I. Siberian Math. J. **51** No. 1 (2010) 217–235.
4. Khisamiev A. N.: Σ-Bounded algebraic systems and universal functions. I. Siberian Math. J. **51** No. 3 (2010) 676–693.
5. Khisamiev A. N.: On a universal Σ-function over a tree. Siberian Math. J. **53** No. 3 (2012) 312–320.

On the Computational Strength of Infinite Time Blum Shub Smale Machines

Peter Koepke[1] and Andrey Morozov[2]

[1] University of Bonn, Germany
koepke@math.uni-bonn.de
[2] Sobolev Institute of Mathematics, Novosibirsk, Russia
morozov@math.nsc.ru

Abstract. We show that a real number is computable by an Infinite Time Blum Shub Smale Machine iff it is an element of L_{ω^ω}.

In [1], Infinite Time Blum Shub Smale Machines (ITBM) have been introduced. These machines generalize the standard Blum Shub Smale Machines (BSM) [2] over the real numbers into transfinite time. At successor times, register contents are determined from previous register contents by rational functions. At limit times, registers are set to ordinary limits of previous register contents in the classical real continuum $\mathbb{R}$. In a legitimate computation those limits must always exist.

It was shown in [1] that the Turing halting problem and certain power series are ITBM–computable and that all ITBM–computable reals are elements of L_{ω^ω}, i.e., the ω^ω-th level of Gödel's constructive hierarchy.

We are able to show that this bound is exact.

Theorem *A real number x is ITBM–computable iff $x \in L_{\omega^\omega}$.*

The proof is based on the methods of transforming given ITBM programs into programs computing Turing jumps and iterations.

We are currently working on a natural description of ITBM–computable real functions and of ITBM–computable sets of reals.

References

1. Peter Koepke and Benjamin Seyfferth, *Towards a theory of Infinite Time Blum Shub Smale Machines*, submitted to CiE 2012.
2. Lenore Blum, Mike Shub, and Steve Smale. On a theory of computations and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1):1-46, 1989.

On Uniform Computability in Familiar Classes of Projective Planes

Nurlan Kogabaev

Sobolev Institute of Mathematics, Novosibirsk State University,
Koptyug Prospect 4, Novosibirsk 630090, Russia,
kogabaev@math.nsc.ru

Following [1] we treat a projective plane as partial algebraic system $\mathcal{A} = \langle A, (A^0, {}^0A), \cdot \rangle$ with a disjunction of A into two subsets $A^0 \cup {}^0A = A$, $A^0 \cap {}^0A = \emptyset$ and commutative partial operation "$\cdot$" which satisfy certain axioms.

In the present paper we investigate the existence problem of computable lists for familiar classes of projective planes. Let K be a class of projective planes, closed under isomorphism. A *computable list for K (up to computable isomorphism)* is a uniformly computable sequence $(\mathcal{A}_n)_{n\in\omega}$ of structures in K such that every computable $\mathcal{B} \in K$ is computably isomorphic to $\mathcal{A}_n$ for some n. We obtain the following results:

Theorem 1. *There is no computable list (up to computable isomorphism) for each of the following classes of projective planes:*
(1) freely generated projective planes;
(2) desarguesian projective planes;
(3) pappian projective planes;
(4) all projective planes.

This work was supported by RFBR (grant 11-01-00236) and by the Grants Council under RF President for State Aid of Leading Scientific Schools (grant NSh-276.2012.1).

References

1. Shirshov, A.I., Nikitin, A.A.: On the theory of projective planes. Algebra and Logic. 20, 330–356 (1981).

Polynomial-Time Modifications of Church-Turing Thesis for Three Programming Languages

Nikolay K. Kosovskiy

St.Petersburg State University, RUSSIA

A pascal-like function is defined as a function computable by a Pascal program using integers with unbounded number of digits as data.

Lists of elements of the traditional value type integer, or files, or constant terms of a special type (i.e. without variables) in a not interpreted signature are examples of data for the programming languages from the family of languages Prolog.

The **number of steps** for a pascal-like function is defined as the number of fulfilled statements, computed expressions, sub-expressions (boolean and arithmetic) and executed calls to functions.

The **run memory size** for a pascal-like function is defined as the used memory (including the length of stacks used for all actual parameters of a function or a procedure or a predicate call) during the run of a program (according to the rules of operational semantics).

Similar but simpler definitions are used for the other concided programming languages.

Let the definition of a pascal-like function not contain pointers, sets, records as well as procedures and functions as parameters.

Let any condition for a rule application in the definition of a Refal-5 function does not contain a recursive call of the defined function.

A pascal-like function or query of Turbo or Visual Prolog or Refal-5 function is called **double polynomial** if both the number of steps and the run memory size are less than a polynomial under the record length of input data.

Theorem 1. *The class of all double polynomial pascal-like functions and two classes of all double polynomial functions computable by queries situated in the goal section of a Turbo or Visual Prolog program and the class of all double polynomial Refal-5 functions equal to the class **FP**.*

Such a result essentially simplifies the proof of function belonging to the class **FP**.

So we have the following **modification of Church-Turing thesis**:
"A function computable in polynomial time in any reasonable computational model using a reasonable time complexity measure is computable by a double-polynomial pascal-like function without pointers, sets and records as well as procedures and functions as parameters and by a double-polynomial query from the goal section of a Turbo or Visual Prolog program and by a double-polynomial Refal-5 function without recursive calls in conditions of rule executions".

Permutation Pattern Matching and its Parameterized Complexity

Martin Lackner

Vienna University of Technology
lackner@dbai.tuwien.ac.at

Abstract. The concept of pattern avoidance (and, closely related, pattern matching) in permutations arose in the late 1960ies. It was in an exercise of his Fundamental algorithms that Knuth asked which permutations could be sorted using a single stack. The answer is simple: These are exactly the permutations that do not contain the pattern 231. A permutation T contains a permutation P as a pattern if there exists a subsequence of T that is order-isomorphic to P. Questions regarding permutations containing (or avoiding) certain patterns have been studied intensively within the field of enumerative combinatorics.
In this talk I will take the viewpoint of computational complexity. Computational aspects of pattern avoidance, in particular the analysis of the PERMUTATION PATTERN MATCHING (PPM) problem, have received far less attention than enumerative questions. This problem is defined as follows: Given permutations P and T, does T contain P as a pattern?
I will present first results obtained together with Marie-Louise Bruner towards a more fine-grained, parameterized complexity analysis of PPM[1].
We study the question how structural properties of permutations, so-called permutation statistics, influence the complexity of PPM. Permutation statistics have extensively been studied in combinatorics but their impact on computational questions regarding PPM is unexplored. In this talk I will focus on the permutation statistic "alternating runs". This statistic denotes the number of ups and downs in a permutation.
We were able to show that PPM can be solved in time $\mathcal{O}^*(1.79^{\mathsf{run}(T)})$, where $\mathsf{run}(T)$ denotes the number of alternating runs in T. Since $\mathsf{run}(T)$ is less than n, the length of T, this algorithm also solves PPM in time $\mathcal{O}^*(1.79^n)$. This is the fastest known algorithm for PPM.
I will conclude this talk by pointing out how such a parameterized complexity analysis can shed light on the relation between permutation statistics and computational questions concerning PPM.

Keywords: Algorithms, parameterized complexity analysis, permutation patterns, permutation statistics

[1] Bruner, M.L., Lackner, M.: A Fast Algorithm for Permutation Pattern Matching Based on Alternating Runs. ArXiv e-prints (2012). To appear in Proc. of SWAT'12

Trees for Subtoposes of the Effective Topos

Sori Lee

DPMMS, University of Cambridge
S.Lee@dpmms.cam.ac.uk

From the discovery of the effective topos around 1980, we know that the category of sets is its least (non-degenerate) subtopos, and that the Turing degrees embed (order-reversingly) into the lattice of subtoposes. Our knowledge of subtoposes of the effective topos has not much extended since, and there were only two particular subtoposes (the Lifschitz subtopos and an example by Andrew Pitts) apart from the Turing degrees that we were aware of.

What makes the study of subtoposes so difficult? We have a standard representation of subtoposes by functions $\mathbf{P}(\mathbf{N}) \to \mathbf{P}(\mathbf{N})$ with certain realizability-theoretic closure properties, and it is these functions which we work with when trying to make concrete calculations. Now, the crucial problem is that we lack general understanding of these representing objects:

> **Question.** Given a function $j : \mathbf{P}(\mathbf{N}) \to \mathbf{P}(\mathbf{N})$ representing a subtopos and a set $p \in \mathbf{P}(\mathbf{N})$, what does it mean for a number to belong the set $j(p)$?

The work [1] answers this question as follows, building upon Pitts' description of subtoposes that arise from making subobjects in the effective topos dense and Van Oosten's perspective that subtoposes are meets of sequences of a class of 'basic' subtoposes (in a sense internal to the effective topos).

> **Answer (*the method of sights*).** A number in $j(p)$ is a computable function that acts on some kind of well-founded tree (on some 'sight') and takes values inside p.

This observation proves effective, facilitating the establishment of an infinite family of new (basic) subtoposes as well as comparisons (inequalities and non-inequalities) in between these new examples and known ones such as the Turing degrees. The most of our new examples are "finite" in nature, and the calculations made with them (therefore) have combinatorial flavour, also interlinked with the involvement of trees as above.

The talk will outline this method of sights, demonstrate example calculations, comment on results obtained so far and discuss remaining questions.

References

1. Sori Lee, *Subtoposes of the Effective Topos*, Master's Thesis, Utrecht University, arXiv:1112.5325, 2011.
2. Sori Lee, Jaap van Oosten, *Basic Subtoposes of the Effective Topos*, arXiv:1201.2571, 2012.

On Existence of Strongly Computable Copies in Class of Boolean Algebras

Margarita Leontyeva

[1] Mechanics and Mathematics Department, Novosibirsk State University, 2 Pirogova St., Novosibirsk 630090 Russia, margarita.leontyeva@gmail.com

[2] Sobolev Institute of Mathematics, 4 Ak. Koptyuga av., Novosibirsk 630090 Russia

Boolean algebras are classical object that appears in different parts of mathematics and performs to be popular subject of investigation for many mathematicians for more then one and a half century.

This talk is going to arise within computability theory concerning Boolean algebras and primarily the existence of their strongly computable representations in terms of computable sequence of predicates on Boolean algebra introduced by Yurii Ershov in 1964. By strongly computable structure we understand a computable structure with computable first order diagram.

If $P_0, P_1, ...$ is the mentioned sequence of predicates then we consider the following problem: if $S \subseteq \{P_0, ..., P_n\}$ is a subset of initial segment of the sequence, B is computable, and all predicates from S are computable in B, then can we assert that B has strongly computable copy?

The important notion in theory of Boolean algebras is elementary characteristic. It is a triple of elements $(ch_1(B), ch_2(B), ch_3(B))$ from $\omega \cup \{\infty\}$ which fully describes elementary theory of Boolean algebra.

At Logic Colloquium 2010 we presented the complete answer for the stated problem if $ch_1(B) \neq \infty$. The following question is what do we have in the case of $ch_1(B) = \infty$? At Logic Colloquium 2011 we delivered some partial results for ω-pure Boolean Algebras B with $ch_1(B) = \infty$. This time we are going to discuss what this problem turns out to be in general case.

Acknowledgements. Supported by the Grants Council (under RF President) for State Aid of Leading Scientific Schools (grant NSh-276.2012.1).

Applications of proof mining in nonlinear ergodic theory

Ulrich Kohlenbach[1] and Laurenţiu Leuştean[2]

[1] Department of Mathematics, Technische Universität Darmstadt,
Schlossgartenstrasse 7, 64289 Darmstadt, Germany,
kohlenbach@mathematik.tu-darmstadt.de
[2] Simion Stoilow Institute of Mathematics of the Romanian Academy,
P. O. Box 1-764, 014700 Bucharest, Romania,
Laurentiu.Leustean@imar.ro

This talk reports on recent applications of proof mining in nonlinear ergodic theory [1, 2]. By *proof mining* we mean the use of tools from proof theory to extract effective quantitative information from ineffective proofs in analysis. This line of research, developed by the first author in the 90's, has its roots in Georg Kreisel's program on *unwinding of proofs*, put forward in the 50's.

We present effective and highly uniform rates of metastability (in the sense of Tao [5, 6]) on nonlinear generalizations of the classical von Neumann mean ergodic theorem, due to Saejung [3] for CAT(0) spaces and Shioji-Takahashi [4] for Banach spaces with a uniformly Gateaux differentiable norm.

These results are obtained by developing a method to eliminate the use of Banach limits from Saejung's and Shioji-Takahashi convergence proofs. In this way we get more elementary proofs to which general logical metatheorems can be applied to guarantee the extractability of effective bounds. The techniques of proof mining are applied for the first time to proofs that are based on the axiom of choice, required to prove the existence of Banach limits.

References

1. Kohlenbach, U., Leuştean, L.: Effective metastability of Halpern iterates in CAT(0) spaces. arXiv:1107.3215v1 [math.FA] (2011)
2. Kohlenbach, U., Leuştean, L.: On the computational content of convergence proofs via Banach limits. Philosophical Transactions of the Royal Society A (to appear)
3. Saejung, S.: Halpern's iterations in CAT(0) spaces. Fixed Point Theory Appl. (2010) Article ID 471781 13pp.
4. Shioji, N, Takahashi, W.: Strong convergence of approximated sequences for nonexpansive mappings in Banach spaces. Proc. Amer. Math. Soc. 125 (1997) 3641-3645
5. Tao, T: Soft analysis, hard analysis, and the finite convergence principle. http://terrytao.wordpress.com/2007/05/23/. Appeared in: Tao, T.: Structure and Randomness: Pages from Year One of a Mathematical Blog. AMS (2008)
6. Tao, T.: Norm convergence of multiple ergodic averages for commuting transformations. Ergodic Theory Dynam. Systems 28 (2008) 657-688

Is the Trilogy of Gap, Compression, and Honesty Theorems Still Significant at Type-2?

Chung-Chih Li

School of Information Technology
Illinois State University
Normal, IL 61790, USA

When we developed a notion for type-2 computation using the Oracle Turing Machine (OTM), we justified our model by arguing that many standard complexity theorems were sustained under our computing model. However, the difference between type-1 and type-2 computation is not trivial. Do those type-1 theorems bear the same intuitive meanings as their type-1 counterparts? For example, in classical complexity theory, the gap theorem, the compression theorem, and the honesty theorem form a famous trilogy in the field of abstract computational complexity theory. Do we have an equivalent trilogy under our notion of type-2 computation? In this presentation, I will try to partially answer this question.

We have defined a class of functions ($\mathbf{T}_2\mathbf{TB}$) as time bounds for clocking OTM. It is easy to argue that an analog type-2 honesty theorem can be proven under our clocked OTM model. However, the corresponding compression theorem and gap theorem seem falling apart due to the power of type-2 computation where we do not have a type-2 analog of Church-Turing Thesis. For example, while we have a weak analogs of gap theorem and compression, we can construct a rather strong effective operator to have an anti-gap theorem. We state these theorems as follows:

– Type-2 Honesty Theorem: *There exists a recursive function g such that $\{\alpha_{g(i)} | i \in \mathbf{N}\}$ is a measured set, and for all $\beta \in \mathbf{T}_2\mathbf{TB}$, if $\beta = \varphi_i$, then $\mathbf{C}(\beta) = \mathbf{C}(\alpha_{g(i)})$.*
– Weak Type-2 Gap Theorem: *Given any strictly increasing recursive function g, there exists $\beta \in \mathbf{T}_2\mathbf{TB}$ such that, $\mathbf{C}(\beta) = \mathbf{C}(g \circ \beta)$.*
– Weak Compression Theorem: *There is a recursive function r such that, for every $i \in \mathbf{N}$, if $\varphi_i \in \mathbf{T}_2\mathbf{TB}$, then $\varphi_{r(i)} \notin \mathbf{C}(\varphi_i)$.*
– Anti-Gap Theorem: *There is a recursive operator, $\Theta : \mathbf{T}_2\mathbf{TB} \to \mathbf{T}_2\mathbf{TB}$ such that, for each $\beta \in \mathbf{T}_2\mathbf{TB}$, we have $\Theta(\beta) \in \mathbf{T}_2\mathbf{TB}$ and $\mathbf{C}(\beta) \subset \mathbf{C}(\Theta(\beta))$.*

The honesty theorem promises that we can construct a measured set without losing any complexity class, where being measurable is required to remove the gap phenomena at type-1. At type-2, on the other hand, we do not need the complexity class to be measurable in order to uniformly increase the complexity class. In this regard, we may conclude that the type-2 analog of the honesty theorem is no longer needed, hence the classical trilogy does not share the same significance. Having said that, however, the type-2 honesty theorem is still appreciated in the sense that the family of our type-2 complexity classes is recursive.

An Automata Model for Computations Over an Arbitrary Structure

Jiamou Liu

School of Computing and Mathematical Sciences,
Auckland University of Technology, Auckland, New Zealand
jiamou.liu@aut.ac.nz

Abstract. An algorithm over an algebraic structure

$$\mathcal{S} = (D; f_0, \ldots, f_n, R_0, \ldots, R_k),$$

where each f_i is a function on D and R_i is a predicate on D, can be viewed as a sequence of instructions that use the operations and predicates of the structure. We introduce a finite automata model for performing certain algorithms over an arbitrary algebraic structure $\mathcal{S}$. Such an automaton is a finite state machine equipped with a finite number of registers, which are able to store elements in the domain D of the structure $\mathcal{S}$. The automaton processes finite strings where each position is labeled by an element of $\mathcal{S}$. During the computation, the automaton is able to test the input elements and the register values on relations $R_0, \ldots, R_k$, and perform basic operations $f_0, \ldots, f_n$ on the input and the register values. Our main motivation is that our model is the finite automata analogue of BSS machines over arbitrary structures. In this talk I will present some initial results such as the closure properties, validation problem and emptiness problem. In particular, the validation problem is closely related to the existential first-order logic of the structure $\mathcal{S}$, and the decidability of the emptiness problem depends on the number of registers of the automaton.

This work is joint with Bakhadyr Khoussainov and Aniruddh Gandhi.

Computational Syntax-Semantics Interface of Passive Forms with The Language of Acyclic Recursion

Roussanka Loukanova

Odensgatan 16A, 753 13 Uppsala, Sweden
rloukanova@gmail.com

Keywords: acyclic recursion, computational semantics, syntax semantics interface, underspecification, constraint based, feature-values, passive forms

Moschovakis' type theory L_{ar}^{λ} [3] is a proper extension of Gallin's TY_2 [1], and thus, of Montague's IL, [4]. The language of L_{ar}^{λ} has two kinds of variables — *recursion variables* (alternatively called *locations*) and *pure variables* — and an additional set of *recursion terms*. The recursion terms are formed by using a designated recursion operator. We present the syntax and semantics of the language of L_{ar}^{λ}. The denotational semantics of L_{ar}^{λ} is compositional on the structure of the L_{ar}^{λ} terms. The notion of intension in L_{ar}^{λ}, entirely different from Carnap intension in TY_2 and IL, is the algorithmic sense of language.

We present the potentials of L_{ar}^{λ} for computational syntax-semantics interface of human language (commonly denoted by NL) and semantic underspecification at the object level of L_{ar}^{λ}. We define the relation *render* between NL and L_{ar}^{λ} for passive forms in NL, by Constraint-Based Lexicalized Grammar (CBLG) apparatus (extending the technique in [2]) that covers a varieties of approaches, e.g., HPSG, LFG, and the grammatical framework GF. This work on L_{ar}^{λ} is contribution to formalization of Minimal Recursion Semantics (MRS) used in HPSG, by higher-order type theory, and broadly, to CBLG approach for inclusion of semantic representations. E.g., GF has a prominent placement in the computational applications of logic for language processing and is open for syntax-semantics development.

References

1. D. Gallin. *Intensional and Higher-Order Modal Logic.* North-Holland, 1975.
2. R. Loukanova. Semantics with the language of acyclic recursion in constraint-based grammar. In G. Bel-Enguix and M. D. Jiménez-López, editors, *Bio-Inspired Models for Natural and Formal Languages*, pages 103–134. Cambridge Scholars Publishing, 2011.
3. Y. N. Moschovakis. A logical calculus of meaning and synonymy. *Linguistics and Philosophy*, 29:27–89, 2006.
4. R. H. Thomason, editor. *Formal Philosophy: Selected Papers of Richard Montague*, ed. Richmond Thomason. Yale University Press, New Haven, Connecticut, 1974. Edited, with an introduction, by Richmond H. Thomason.

Categorical Universal Logic: Duality as Semantics

Yoshihiro Maruyama

Quantum Group, Dept. of Computer Science, Oxford University
maruyama@cs.ox.ac.uk

Categorical Universal Logic relativizes the logic of topos to monads, with the purpose of reaching a universal conception of logic, and of providing foundations of categorical semantics for various logical systems, including intuitionistic logic, substructural logics, quantum logics, (topological and convex) geometric logics, and many others.

For a topos $\mathbf{E}$, the subobject functor $\mathrm{Sub}_{\mathbf{E}}(\text{-}) : \mathbf{E}^{\mathrm{op}} \to \mathbf{HeytAlg}$ essentially provides the logic of the topos $\mathbf{E}$. It is a (higher order) hyperdoctrine in William Lawvere's terms, a tripos in Hyland-Johnstone-Pitts' terms, and a (higher order) fibration in Bart Jacobs' terms. We aim at placing such a concept in a much broader context, providing categorical semantics for different logics in a uniform manner, and clarifying a universal form of logic which does not rely on syntactic or semantic details; from this point of view, what we pursue is not really semantics for logic, but the very conception of logic, which we think is what the idea of hyperdoctrine is ultimately about.

From the viewpoint of algebraic logic, (the Lindenbaum-Tarski algebra of) intuitionistic propositional logic is the free algebra on the set of propositional variables. We thus regard propositional logics as free algebras (in varieties). And the generic notion of free algebra is given by the concept of monad in category theory. Monads on a category $\mathbf{C}$ can be seen as $\mathbf{C}$-structured free algebras For example, let $\mathbf{C} = \mathbf{Top}$, and then monads on $\mathbf{C}$ correspond to topological free algebras. Since free algebras in our logical context have no such additional structure, we focus on monads T on $\mathbf{Set}$, which are thought of as representing propositional logics.

For a monad T, we define a T-hyperdoctrine as a functor (an $\mathbf{Alg}(T)$-valued presheaf or a $\mathbf{C}$-indexed T-algebra) $F : \mathbf{C}^{\mathrm{op}} \to \mathbf{Alg}(T)$ satisfying adjointness conditions (and the corresponding additional conditions like Beck-Chevalley) to represent logical structures of interest, which may be quantifiers, equality, comprehension, object classifier, and so on. Here, $\mathbf{C}$ is supposed to have products at least, and exponentials are required as well if we are concerned with higher order logics. We are omitting those "logicality" conditions on the monad T that are necessary for expressing adjointness conditions. We consider that the concept of T-hyperdoctrine gives a universal form of logic (and set theory qua higher order logic).

In Categorical Universal Logic, we propose to see "Duality as Semantics". Duality theory can actually be used to construct models of T-hyperdoctrines. For instance, the well-known dual adjunction between frames and topological spaces gives us a T-hyperdoctrine for (topological) geometric logic. Such duality-based T-hyperdoctrines may be applied to prove consistency and independence results, as they can be used to obtain sheaf models in the classic case of intuitionistic logic. We can find a lot of duality T-hyperdoctrines in diverse contexts, some of which turn out to be particularly useful.

Dualities between Scott's continuous lattices and convex structures (algebras of the distribution monad or set-theoretical convexity spaces) provide T-hyperdoctrines for convex geometric logic, namely Filt-hyperdoctrines for the filter monad Filt, where we regard continuous lattices as the propositional logic of convex sets, or point-free convex structures as discussed in the author's previous paper (recall frames or locales give the propositional logic of open sets); from this perspective, Filt-hyperdoctrines give the quantified logic of convex sets, which we call convex geometric logic.

By means of a certain duality T-hyperdoctrine of quantum nature, we can solve those difficulties on universal quantifier and substituition that have been known in Heunen-Jacobs' categorical quantum logic based on dagger kernel categories. Duality T-hyperdoctrines have many more merits than can be mentioned here, like having object classifiers in very general situations.

Categorical Universal Logic also leads us to coalgebraic predicate logic. From the duality-as-semantics perspective, duality theory in coalgebraic propositional logic is relevant to the lifting of T-hyperdoctrine structures from propositional dualities to Algebra-Coalgebra dualities, which turns out to be always possible under certain conditions. Thus, we can interpret quantified modal logics via Algebra-Coalgebra dualities. This vividly illustrates our idea that duality for propositional logics is categorical semantics for their first order (and higher order) extensions.

Turing and Wittgenstein: from the Foundations of Mathematics to Intelligent machines

Stéphane Massonet

IBM, Belgium

In 1939, Ludwig Wittgenstein lectured at the University of Cambridge on the topic of the Foundations of Mathematics. Not only did Alan Turing lecture on a similar topic that same year, but he attended the philosopher's seminars. This encounter has generally been considered by critics and historians as a significant one, clearly underlining the differences in their respective approach to mathematics and philosophy of the mind.

We would like to revisit these lectures in order to gain a better insight on the issues discussed and understand how they led to the development of a dialogue between the mathematician and the philosopher. As we will see, the core of Wittgenstein's lectures on the foundation of mathematics is based on a study of the usage of grammar and the relationship of mathematics with everyday language. We will see how this approach forms the basis of his reading and his understanding of the concept of the Turing Machine. We will also look at how Turing used and extended some of the mechanical images of the human calculus in his later writings on Artificial Intelligence, when he attempted to conceive machine intelligence behind the question : “Can machines think ?”

Keywords : Philosophy of Mind, Computation, Foundations of Mathematics, Turing Machine, Machine intelligence, Artificial intelligence, Game theory.

Lossless compressors: Degree of compression and optimality

Armando Matos[1,3,⋆]
Andreia Teixeira[1,2,⋆⋆]

[1] Universidade Porto
[2] Instituto de Telecomunicações
[3] LIACC

Abstract. Lossless compressors are very useful in practice: they compress many larger files into smaller files without losing any information. Their principle is based on the exploration of some form of regularity, such as the existence of string repetitions. Many other forms of lossless compressors are conceivable; for instance, a very specialized compressor would detect if a file consists of a sequence of binary integers that were generated by the linear congruential method ([1]), and, if so, output a much shorter description of the file. To each compressor, corresponds a decompressor, which will be called *expander*. We study some theoretical aspects of lossless compressors. As background for the rest of the paper, we present an elementary method to transform every (lossless) compressor C into a *normalized compressor*, that is, a compressor in which no word is expanded by more than one bit; this idea is used to reformulate a particular result of Algorithmic Information theory and applied to a real world compressor. Then, the compressibility bounds of uniform distributions, using both general and normalized compressors, are studied. For a large class of problems, called "inversion problems", Levin mentioned in [2] the existence of a fixed algorithm with a time of the same order of magnitude of *any* algorithm that solves any other problem in the class. Inspired by Levin's universal optimal search, we define, using a universal expander Turing machine, an optimal compressor, apart from an additive constant, and within a time of the same order of magnitude.

References

1. D. E. Knuth, The Art of Computer Programming, Volume 2: Seminumerical Algorithms, third edition, [Section 3.2.1: The Linear Congruential Method, pp. 10-26], Addison-Wesley, ISBN-0-201-89684-2, 1997.
2. L. Levin, *Universal sequential search problems*, Problems of Information Transmission, (translated from Problemy Peredachi Informatsii (Russian)), 9:3, pp 265–266, 1973.

⋆ acm@dcc.fc.up.pt; All the authors are partially supported by CSI^2 (PTDC/EIA-CCO/099951/2008)
⋆⋆ andreiasofia@dcc.fc.up.pt; The author is also supported by the grant BIM/n°50/2011 of IT

Continuous reductions of regular languages

Filip Mazowiecki

University of Warsaw, Banacha 2, 02-097 Warszawa, Poland
fm248407@students.mimuw.edu.pl

In [1] Wagner described a hierarchy of ω-regular languages, which corresponds to the existence of continuous reductions between languages, i.e., continuous functions $f : A^\omega \to B^\omega$ such that $f^{-1}(M) = L$ for $L \subseteq A^\omega$, $M \subseteq B^\omega$. We would like to develop analogous theory for regular languages of finite words. While the space of infinite words has a natural metric definition (that of the Cantor space), the space of finite words does not have a canonical metric. We consider metrics defined by embeddings into the space of infinite words that preserve prefix convergence. This last condition guarantees that the natural connection between finite and infinite words is not lost in the process of embedding.

The space of finite words with a given metric can be considered a uniform space. We show that under mild assumptions embeddings give only two non-isomorphic uniform structures. In both cases the space is not compact, so the notions of continuity and uniform continuity diverge and the hierarchies for continuous and uniformly continuous reductions need to be described separately.

The first embedding gives a discrete metric and the hierarchies of all languages are simple. For the second embedding we define canonical automata, similar to those defined by Wagner in [1], and show that every regular language is equivalent to one of the languages recognized by them. In this case we describe only the hierarchy for regular languages and show an example of a non-regular language that is not equivalent to any of the regular languages with respect to any reduction. In the end we examine the hierarchy of Borel languages in Σ^ω and Σ^* induced by Lipschitz reductions. We show that in Σ^ω the hierarchy of Borel languages for 1-Lipschitz reductions is an extension of the hierarchy for continuous reductions. Then we show that it is similar in the case of Σ^* with metrics defined by embeddings.

The basic tool we use is a game characterization of the existence of reductions. Thus for every kind of reduction – continuous, uniformly continuous and Lipschitz – we define an appropriate game.

References

1. Klaus Wagner. On ω-regular sets. *Information and Control*, 43(2):123–177, 1979.

Bases for AC^0 and other Complexity Classes

Stefano Mazzanti

Università Iuav di Venezia
Fondamenta delle Terese 2206, 30123 Venezia, Italy
email: mazzanti@iuav.it

Abstract. The sets of AC^0, TC^0, NC^1 functions and other canonical complexity classes are defined as the substitution closure of a finite function set.

Keywords: concatenation recursion on notation, substitution basis.

1 Introduction

A finite function set F is a *substitution basis* for a function class C (and C is the *substitution closure* of F) when C can be defined using only the functions in F, the projection functions and the substitution operator.

Recently, the class $\mathcal{F}TC^0$ of functions computable by polysize, constant depth threshold circuits has been shown to be the substitution closure of $\{x+y, x\dot{-}y, x\wedge y, \lfloor x/y \rfloor, 2^{|x|^2}\}$ where $x \wedge y$ is the *bitwise and* of x and y [3].

In a previous paper [2], we showed that a function class closed with respect to substitution and *concatenation recursion on notation* admits a substitution basis, provided that it contains integer division. By applying this result to Clote-Takeuti characterizations of $\mathcal{F}TC^0$ and $\mathcal{F}NC^1$[1], we obtained the above mentioned basis for $\mathcal{F}TC^0$ and a new basis for $\mathcal{F}NC^1$.

In this paper, we improve the techniques and results of [2]. The existence of a basis for a function class does not need the division requirement anymore and a basis for $\mathcal{F}AC^0$ is introduced. Finally, by considering complete problems for canonical complexity classes under AC^0 reductions, substitution bases for canonical complexity classes are introduced.

References

1. P. Clote, Computation models and function algebras, in Handbook of Computability Theory, ed. E.R. Griffor, Elsevier Science B.V. (1999) 589-681.
2. S. Mazzanti, CRN elimination and substitution bases for complexity classes, preprint (2011), http://rice.iuav.it/311/1/bases18rice.pdf.
3. S. A. Volkov, Generating Some Classes of Recursive Functions by Superpositions of Simple Arithmetic Functions, Dokl. Akad. Nauk 415, 4 (2007) 439-440. English translation Dokl. Math. 76, 1 (2007) 566-567. Preliminary version in Russian with proofs available at http://volkov-articles.narod.ru/ffom_manyfloors.pdf.

Social Networks and Collective Intelligence

Manuel Mazzara[1], Antonio Marraffa[2], Luca Biselli[2], Simona De Nicola[3], Luca Chiarabini[4], and Georgios Papageorgiou[1]

[1] Newcastle University, UK
[2] Polidoxa.com, Germany
[3] University of Bologna, Italy
[4] University of Parma, Italy

Access to global information is a fact of primary importance in a global world. Internet contains a huge amount of documents and it has a big potential as a news media, but the key is in the mechanism in which information is accessed. The most important aspect is having access to the most possible independent and unbiased information source for any specific topic or, at least, having access to a variety of different sources. Indeed, despite the significant technological advances, we still cannot reach the best information in real time.

Our aim is combining user friendliness of search engines and trustworthiness of social networks in order to define an integrated platform enabling users to get trustworthy news on their favorite topics. We called this platform Polidoxa (from greek "poly", meaning "many or several" and "doxa" meaning "common belief or popular opinion"). Polidoxa works with a Trusted Rank Algorithm based on the definition of trusted relationship between users. The immediate contacts have more influence while the others see a reduction of their influence which is proportional to their distance. Polidoxa is designed to work as a stigmergic system, a strategy based on what can be found in biological systems. Social interaction and networking is enhanced by the collective intelligence, which is superior to the sum of knowledge of individuals and opinion trends can be predicted via swarm intelligent algorithms.

Polidoxa can offer a platform for discussion which elevates users to a higher level of knowledge, criticism and consciousness. The principle of the "agora" (it was a "Gathering place" or "Assembly" in ancient Greek city-states) is also behind the concept of Polidoxa and some of its design choices. The "agora" is the ancient place where people used to meet and discuss. The Socratic discussion represents the consistency check on shared information. Polidoxa reintroduces the "agora", this time is a "virtual agora" where all the advantage of the traditional "agora" are actually amplified by means of virtual networks, better known as social networks. Collective intelligence is the key to information and Internet has an enormous potential to fix the problem of trustworthiness, but the current instruments commonly used like Google and Facebook lack the most important concept in this field: they do not embed the notion of individual trustworthiness of a source. Polidoxa, instead, connects local knowledge making them usable for everybody and it is conceived to promote public awareness and discussion in total freedom, like in an open piazza.

Solving the Dirichlet Problem Constructively

D.S. Bridges and M. McKubre-Jordens*

University of Canterbury
Private Bag 4800
Christchurch 8140
New Zealand
douglas.bridges@canterbury.ac.nz;
maarten.jordens@canterbury.ac.nz

Abstract. Working within the Bishop-style constructive mathematical framework, we present sufficient reasonable constraints on a domain in n-dimensional Euclidean space which ensure that if the Dirichlet problem has weak solutions on arbitrary close internal approximations to a domain, then it has a (perforce unique) weak solution on that domain as well. We also give Brouwerian examples which show that, even on geometrically quite reasonable domains in 2-dimensional Euclidean space, the existence of solutions for the Dirichlet problem is an essentially nonconstructive result.
A corollary of the Brouwerian examples is that, perhaps surprisingly, there is no universal algorithm for computing solutions to the Navier-Stokes equations of fluid flow and that hence any existence result must be purely theoretical.

Keywords: constructive analysis; Dirichlet problem; Navier-Stokes equations; Brouwerian example; Markovs principle; omniscience principle.

* corresponding author

Computability of Narrative

Loizos Michael

Open University of Cyprus
loizos@ouc.ac.cy

Narratives are ubiquitous across human cultures, and building machines that comprehend them may lead to more natural human-machine interactions, and a deeper understanding of human cognition. We propose a logic-based approach to this challenge.

We adopt a rather generic view of narrative [1], as some partially-ordered collection of events and facts that can be *embedded* (i.e., assumed to have taken place at particular points) in time, so that the embedding is *consistent* with a given *domain*. Each domain comprises the background knowledge and beliefs of a particular listener of the narrative (including static laws like "whenever an entity is a penguin, it is also a bird", and causal laws like "turning the switch on causes the light to come on, if the fuse is okay"), and acts as the context within which the narrative is interpreted. In case consistency with an entire domain is not possible, the collection may still have some degree of *plausibility* as a narrative, depending on the part of the domain it is consistent with. This approach effectively allows the encoding of varying degrees of defeasible beliefs in a domain.

One may argue that plausibility is insufficient to serve as a universal determinant of narrativeness. It may account for the coherence of a narrative, but how about other properties: syntactical meaningfulness, or cohesiveness; description of a setting, a problem, and a resolution; relevance and informativeness for the reader; and so on? Our response is that all these properties are simply constraints (possibly defeasible ones) that a narrative is expected to respect, and may be accommodated as part of the listener's domain.

Determining whether a collection is indeed a narrative with respect to a domain is not immediate, as there are infinitely many embeddings that could be considered. It can be formally shown, however, that determining narrativeness is decidable. Furthermore, it is possible to choose the most plausible narrative from a given set of collections, and, therefore, to order narratives according to their plausibility. Each narrative can be also assigned a *canonical index* that allows its storage and retrieval. It follows, in particular, that narratives are computably enumerable, as one can produce them all in some order.

As with other narrative-related processes, question answering reduces to a computational problem: Search for a *model* (a truth-assignment to the facts and events at each time-point that agrees with the narrative and the domain), and check whether the fact that corresponds to a given question is made true within that model. If this is the case for at least one model / for all models, then the question is *possibly true* / *certainly true*.

Recent work shows how the *recognizing narrative similarity* task can be formalized and attacked within an extension of this work that accounts for authorial intentions [2].

References

1. Michael, L.: Computability of Narrative. In: 2nd Comp. Models of Narrative Workshop (2010)
2. Michael, L.: Similarity of Narratives. In: 3rd Comp. Models of Narrative Workshop (2012)

Ant Collectives as Computers

Loizos Michael

Open University of Cyprus
loizos@ouc.ac.cy

An ant collective is often used as the prototypical example of a system that is able to achieve more than the sum of its constituent parts. Indeed, both theoretical models and entomological studies support the conclusion that although the available sensors, deliberation, and actuators of each individual ant have limited capabilities, an ant collective as a whole may overcome these limitations. Ant collectives are known to be able, for instance, to sort different types of objects into piles, to find shortest paths between points of interest, and to dispatch workers to the tasks that are most important at the time.

Novel, and rather successful, general algorithms have been conceived and built upon the observed behavior of ants, giving rise to research areas such as Ant-Based Clustering and Ant Colony Optimization. Unlike, however, the well-studied problem of capitalizing the modus operandi and abilities of ant-like agents, a more conceptual problem that has received less attention is the determination of the principled capabilities of a typical collective of ants. This latter problem we have investigated in past and ongoing work.

Building on studies of ants, earlier work [1] proposed a biologically and physically plausible model for ants and pheromone, according to which: Ants enter / exit the system by dropping in / out at specified locations. Ants sense pheromone in their vicinity, and randomly choose their next location, giving preference to higher pheromone concentrations. Pheromone enters the system through pheromone pumps and ant secretions. An ant secretes pheromone whenever pheromone concentration at its current location exceeds some threshold, and this is not the case for some neighboring location. In doing so, the ant propagates the high pheromone concentration at its current location to neighboring locations through diffusion. Pheromone exits the system through dissipation.

The proposed model was shown sufficient for designing a gadget comprising paths, pumps, and ant sources / sinks, such that: when sufficiently many (resp., few) ants enter through a specified path, then sufficiently few (resp., many) ants exit through a certain other path. The gadget follows the design and operation of a typical resistor-transistor electronic inverter, establishing, thus, the principled ability to design larger gadgets within which ant behavior can simulate the workings of full-fledged logic circuits.

Subsequent work [2] sought to validate the aforementioned principled ability, by refining the original model, and implementing it through the development of a design and simulation tool. This latter work showed how circuits found in modern computers (e.g., adders, memory circuits, oscillators) can be effectively simulated by an ant collective.

References

1. Michael, L.: Ant-Based Computing. Artificial Life 15(3), 337–349 (Summer 2009)
2. Michael, L., Yiannakides, A.: An Ant-Based Computer Simulator. In: Proceedings of the 13th International Conference on the Synthesis and Simulation of Living Systems (July 2012)

WebSense: Making Sense of the Web

Loizos Michael

Open University of Cyprus
loizos@ouc.ac.cy

The Web, arguably the greatest marvel of the modern information era, is associated with what is, perhaps, an equally great paradox: its under-utilization. Billions of pages of natural language text, encoding essentially the collected knowledge of our species, are demoted to a mere *collection* of information, and accessed through tools that only identify pages containing posed queries, within which one *hopes* to find some answers.

It is imperative to have solid ways to exploit the *knowledge*, not just the information, that is woven into the Web. The time is ripe to build search engines able to compute and return *answers* to queries. And certain such efforts are, sure enough, already under way. Wolfram|Alpha, for instance, is a tool that offers tangible evidence of these efforts, as long as one focuses on axiomatized knowledge, a certain type of knowledge for which concrete representations and rules of inference are available, and well understood.

Even for such axiomatized knowledge, the task is by no means easy. Yet, one would hope for a search engine able to also capitalize the vast unaxiomatized knowledge found on the Web, encoding the collective beliefs, biases, misconceptions, and common sense of the human race. How can such knowledge be employed to respond to a query such as “they first met last summer at a common friend’s house” with the answer “they were not married at the time”, even if the query does not appear on any single web-page?

We report herein on our work over the past few years in this direction, resulting in a working prototype of such a search engine endowed with *websense*. The engine parses downloaded web-pages with state-of-the-art NLP tools, and the extracted semantic and syntactic information is then translated to a relational logic-based form [4]. Efficient and noise-resilient learning algorithms, able to cope with missing information in principled ways [3], populate a knowledge base with rules that determine the existence of relations among objects, as dictated collectively by the downloaded web-pages. Given a query in natural language, the engine first translates the query to a relational logic-based form, and then relational reasoning is employed to derive the inferences that follow from the knowledge base and that specific query [1]. The inferences are, finally, composed back into natural language sentences that form the answer to the query. In earlier work [2] we have formally shown that this process can be understood precisely in the sense that we have suggested: as the process of drawing websense inferences from a given query.

References

1. Michael, L.: Autodidactic Learning and Reasoning. Ph.D. thesis, Harvard University, Massachusetts, U.S.A. (May 2008)
2. Michael, L.: Reading Between the Lines. In: Proc. 21st IJCAI. pp. 1525–1530 (July 2009)
3. Michael, L.: Partial Observability and Learnability. Artif. Int. 174(11), 639–669 (July 2010)
4. Michael, L., Valiant, L.G.: A First Experimental Demonstration of Massive Knowledge Infusion. In: Proc. 11th Int. Conf. on Principles of KR&R. pp. 378–389 (September 2008)

Simultaneous Learning and Prediction

Loizos Michael

Open University of Cyprus
loizos@ouc.ac.cy

In a typical supervised learning scenario, an agent trains on labeled examples, constructs a model of the data, and applies it to predict the missing labels of new examples. In many real-world settings, however, neither such explicit labels nor the examples are fully observed, as assumed. Yet, one would hope that the existing, and extensive, work on supervised learning could be still rigorously applied and exploited in these settings.

Call *autodidactic learning* the setting where an agent trains on partially-observable examples, constructs a model of the data, and applies it to predict the missing values of those attributes that are *masked* in new partially-observable examples. A physician, for instance, with access to partial patient records may autodidactically learn rules governing the human physiology, and use them to predict unobserved (N.B. not unobservable) medical conditions of new patients. Earlier work investigated how autodidactic learning algorithms can be obtained from supervised algorithms for concept learning [2, 4].

Consider, then, an algorithm that autodidactically learns a rule to predict any single given attribute. How should this algorithm be used so that the performance of reliably completing missing information in new partially-observable examples is maximized?

Assume that a rule for each attribute is available. The application of the rules on an example is: *flat* if no rule has access to the predictions of other rules; *chained* if at least one rule is applied after another, so that the former can use the prediction of the latter as if it were part of the example. It can be shown that there exist situations where the chained application of rules strictly outperforms the flat application of any set of rules.

How can the provable benefits of rule chaining be attained? Learning the rules first and then chaining them increases the completed information compared to their flat application, but at the expense of the reliability of the predictions. *Simultaneous learning and prediction* achieves the best of both worlds: learn a rule for each attribute; use a flat application of these rules on the training examples to complete some missing information; repeat t times on the resulting examples. Chain all available rules in the order they were learned, and apply the chained set of rules to make predictions on future examples. By appropriate choice of the learning parameters, it can be shown that SLAP-ing outperforms a flat application of these rules while retaining the reliability of the predictions (cf. experiments and theory [1, 3]), and it is fixed-parameter tractable for parameter t.

References

1. Michael, L., Valiant, L.G.: A First Experimental Demonstration of Massive Knowledge Infusion. In: Proceedings of KR'08. pp. 378–388 (2008)
2. Michael, L.: Autodidactic Learning and Reasoning. Ph.D. thesis, Harvard University, Massachusetts, U.S.A. (2008)
3. Michael, L.: Reading Between the Lines. In: Proceedings of IJCAI'09. pp. 1525–1530 (2009)
4. Michael, L.: Partial Observability and Learnability. Artif. Intellig. 174(11), 639–669 (2010)

Finitary reducibility on equivalence relations

Russell Miller[1] and Keng Meng Ng[2]

[1] Queens College, 65-30 Kissena Blvd., Flushing, NY 11367 &
CUNY Graduate Center, 365 Fifth Avenue, New York, NY 10016, USA.
Russell.Miller@qc.cuny.edu, http://qc.edu/~rmiller
[2] Nanyang Technological University, 21 Nanyang Link, Singapore 637371.
KMNg@ntu.edu.sg, http://www.ntu.edu.sg/home/kmng/.

Keywords: computable reducibility, equivalence relations, finitary reducibility.

Numerous authors have considered the notion of computable reducibility (or m-reducibility) on equivalence relations on the natural numbers. This holds of equivalence relations E and F (written $E \leq_c F$) if there exists a computable total function f on ω such that $x\ E\ y$ if and only if $f(x)\ F\ f(y)$. Recent results include both the existence of such reductions and, for many pairs of equivalence relations, the impossibility of any such reduction.

Considering several of the proofs of non-reducibility, we have defined a weaker notion, finitary reducibility, to help analyze these negative results. We say that E is *n-arily reducible* to F, written $E \leq_c^n F$, if there are computable total functions $f_1, \ldots, f_n : \omega^n \to \omega$ such that, for all $j, k \leq n$ and all $i_1, \ldots, i_n \in \omega$, $i_j\ E\ i_k$ if and only if $f_j(i_1, \ldots, i_n)\ F\ f_k(i_1, \ldots, i_n)$. If the indices of such functions can be found uniformly for every n, then E is *finitarily reducible* to F, written $E \leq_c^\omega F$.

In this talk we will give examples of how these new notions can be used. For example, it was shown in [2] that neither of the relations E^{ce}_{min} and E^{ce}_{max} (defined by equality of maxima and minima in c.e. sets) is computably reducible to the other, but finitary reducibility gives us a better comparison: $E^{ce}_{min} \leq_c^\omega E^{ce}_{max}$, whereas $E^{ce}_{max} \not\leq_c^2 E^{ce}_{min}$. In fact, E^{ce}_{max} turns out to be complete among Π^0_2 equivalence relations under 3-ary reducibility, but not so under 4-ary reducibility. The relation of equality on c.e. sets ($i =^{ce} j$ if and only if $W_i = W_j$) is complete for finitary reducibility among Π^0_2 equivalence relations, which is of particular interest because it has been shown in [4] that there is no complete Π^0_2 equivalence relation under $\leq_c$.

References

1. Andrews, U., Lempp, S., Miller, J., Ng, K.M., San Mauro, S., Sorbi, A.: Computably enumerable equivalence relations (to appear).
2. Coskey S., Hamkins, J.D., Miller, R.: The hierarchy of equivalence relations on the natural numbers under computable reducibility, to appear in Computability.
3. Fokina, E.B.: Index sets for some classes of structures, Ann. Pure Appl. Logic 157 (2009) 2-3, 139-147.
4. Ianofski, E., Miller, R., Ng, K.M., Nies, A., Stephan, F.: Completeness among Σ^0_n and Π^0_n equivalence relations (to appear).

An integral test for Schnorr randomness and its applications

Kenshi Miyabe

Research Institute for Mathematical Sciences, Kyoto University

Some recent researches show that some classical theorems with "almost everywhere" can be converted to effective version with "for each random point". One example is that a real is Martin-Löf random iff each computable function of bounded variation is differentiable at the real. The author [1] has given a Kurtz randomness version where we find that an integral test is a useful tool to study the relation between algorithmic randomness and computable analysis. Here we give a version of Schnorr randomness. Consider a computable metric space (X, d, α) and a computable measure μ on it.

Definition 1. *An integral test for Schnorr randomness is a nonnegative lower semicomputable function* $t : X \to \overline{\mathbb{R}}^+$ *such that* $\int t d\mu$ *is a computable real.*

Theorem 1. *A point* z *is Schnorr random iff* $t(z) < \infty$ *for each integral test* t *for Schnorr randomness.*

As an application we have another effectivized version of a classical theorem.

Theorem 2. *Let* $f, g :\subseteq X \to \mathbb{R}$ *be the differences between two integral tests for Schnorr randomness. Then* $f(x) = g(x)$ *for each Schnorr random point iff* $||f - g||_1 = 0$.

The class of the differences between two integral tests for Schnorr randomness has some characterizations and is an important class.

Definition 2. *A function* $f :\subseteq X \to \mathbb{R}$ *is* L^1*-computable with an effective code if there exists a computable sequence* $\{s_n\}$ *of finite rational step functions such that* $f(x) = \lim_n s_n(x)$ *and* $||s_{n+1} - s_n||_1 \leq 2^{-n}$ *for all* n.

Definition 3. *A function* $f :\subseteq X \to \mathbb{R}$ *is* Schnorr layerwise computable *if there exists a Schnorr test* U_n *such that the restriction* $f|_{X \setminus U_n}$ *is uniformly computable.*

Theorem 3. *Let* SR *be the set of Schnorr random points. Then*

$$\begin{aligned} &\{f|_{\mathrm{SR}} \mid f \text{ is the difference between two integral tests for Schnorr randomness}\} \\ =&\{f|_{\mathrm{SR}} \mid f \text{ is an } L^1\text{-computable function with an effective code}\} \\ =&\{f|_{\mathrm{SR}} \mid f \text{ is Schnorr layerwise computable and its } L^1\text{-norm is computable}\} \end{aligned}$$

References

1. Miyabe, K.: Characterization of Kurtz randomness by a differentiation theorem, submitted

Program Extraction with Nested Inductive/Coinductive Definitions

Kenji Miyamoto* and Helmut Schwichtenberg
miyamoto@mathematik.uni-muenchen.de,
schwicht@mathematik.uni-muenchen.de

Mathematisches Institut der Ludwig-Maximilians Universität München

Abstract

We present our work on program extraction and a case study on uniformly continuous functions working in our proof system Minlog [4].

Ulrich Berger and Monika Seisenberger [1,2] inductively/coinductively defined a predicate of the uniform continuity and informally extracted Haskell programs from their constructive proofs of it. Our work enriches the Theory of Computable Functionals [3] and its computer implementation Minlog in order to formalize case studies by Berger and Seisenberger.

We extract from formal proofs programs which translate a uniformly continuous function on Cauchy reals in $[-1, 1]$ into a non-well founded tree representation, and vice versa. Via Kreisel's modified realizability interpretation, the extracted programs involve certain recursion and corecursion operators which come from nested inductive/coinductive definitions. The non-well founded tree representation of uniformly continuous functions is of ground type. In this way, we manage to understand uniformly continuous functions through approximating non-well founded objects.

References

1. U. Berger: From coinductive proofs to exact real arithmetic, in E. Grädel and R. Kahle (ed.) *Computer Science Logic*, 132–146, LNCS 5771, 2009.
2. U. Berger and M. Seisenberger: Proofs, programs, processes, in *Theory of Computing Systems Online First*, Springer-Verlag, 2012.
3. H. Schwichtenberg and S. Wainer: *Proofs and Computations*, Perspectives in Logic, ASL and Cambridge University Press, 2012.
4. The Minlog system. http://minlog-system.de/

* Kenji Miyamoto is supported by the Marie Curie Initial Training Network in Mathematical Logic – MALOA – From MAthematical LOgic to Applications, PITN-GA-2009-238381

How to Accept Turing's Thesis without Proof or Conceptual Analysis, without also Feeling Qualms

Masaaharu Mizumoto

1-1 Asahidai, Nomi, Ishikawa, 923-1292 Japan
Japan Advanced Institute of Science and Technology
mizumoto@jaist.ac.jp

In his (1936), Turing (i) presented what is called Turing's Thesis (TT) today, that every number that can naturally be regarded as computable can be computed by his universal Turing machine, and assuming it (ii) solved Hilbert's Entscheidungsproblem negatively. This much is taken as a matter of course within and outside mathematics today, and (i) and (ii) are two early great mathematical achievements of Turing, or so regarded. However, there are various views concerning the current status of (i), which affects our understanding of (ii). Though not exhaustive, the major attitudes towards (i) consist of:

a-1) TT remains a hypothesis, which is *in principle* not formally provable.
a-2) TT is a hypothesis, which is formally provable (or already formally proved).
b-1) TT is already established beyond doubt, as a matter of definition.
b-2) TT is already established beyond doubt, through the *conceptual analysis* of (human) computation given by Turing.

In this paper I will first briefly re-examine these views, and claim that each of them is not all satisfactory, leaving us with some qualms about accepting it without reservation. Then I shall present as a better option an alternative understanding of TT, based on Wittgenstein's philosophy of mathematics, large part of which was presented in his 1939 lectures, to which Turing himself attended.

According to Wittgenstein mathematics proceeds with acceptance of proofs, and each time we accept a proof, that fact changes mathematical system as a whole, causing a systematic change of relevant (or in fact *all*) mathematical *concepts*. Now Turing's argument for his thesis was basically a *non-mathematical* contribution to his solution of the Entscheidungsproblem. However, when the latter was accepted as a proof, our concept of "computable" underwent, unbeknownst to most of us, a conceptual change, so that it would *entail* computability by Turing machine. In other words, TT *became* a mathematical fact *as a result of* the fact that his demonstration of the insolubility of the Entscheidungsproblem was accepted as a mathematical proof. This may first look a queer view, turning justificatory order totally upside-down. I will defend this view by Wittgenstein's remarks on mathematics, thereby truly securing Turing's two achievements.

adfa, p. 1, 2011.

Von Neumann's Biased Coin Revisited

Laurent Bienvenu, Benoît Monin

LIAFA - CNRS & Université de Paris 7
Case 7014, 75205 Paris Cedex 13, France
{laurent.bienvenu,benoit.monin}@liafa.jussieu.fr

Abstract. Suppose you want to generate a random sequence of zeros and ones and all you have at your disposal is a coin which you suspect to be biased (but do not know the bias). Can "perfect" randomness be produced with this coin? The answer is positive, thanks to a little trick discovered by von Neumann. In this paper, we investigate a generalization of this question: if we have access to a source of bits produced according to some probability measure in some class of measures, and suppose we know the class but not the measure (in the above example, the class would be the class of all Bernoulli measures), can perfect randomness be produced? We will look at this question from the viewpoint of effective mathematics and in particular the theory of effective randomness.

Keywords: Algorithmic Randomness, Computability, Effective mathematics, Markov measure, Measure theory, Randomness extraction

Complexity of Model Checking in Modal Team Logic

Julian-Steffen Müller

Institut für Theoretische Informatik
Leibniz Universität Hannover
Appelstr. 4, 30167 Hannover, Germany
mueller@thi.uni-hannover.de

Abstract. Modal dependence logic (*MDL*) extends classic modal logic by an atomic dependence operator and was first introduced by Väänänen in [3]. Let $p_1, \ldots, p_n$ be propositional variables. Then we can express with the dependence atom that, on a set of worlds, the variable p_n is determined by the variables $p_1, \ldots, p_{n-1}$.
Recently it was shown by Lohmann and Vollmer that the satisfiability problem for *MDL* is *NEXPTIME*-complete ([2]). That the model checking problem is *NP*-complete was shown by Ebbing and Lohmann ([1]).
In this work we study an extension of *MDL*, because *MDL* does not have the expressive power to formulate that a dependence between variables does not hold. For this purpose modal team logic (*MTL*) extends *MDL* by a classical negation operator.
In this paper we consider the model checking problem for *MTL*. In the main result we will show that with the classical negation alternating quantifications, like in *QBF*, can be expressed. This will lead to a *PSPACE* completeness result for the *MTL* model checking problem.
We will also classify the model checking complexity for *MTL* operator fragments. These fragments will mostly be intractable, but we will show that there are also fragments which are tractable and actually parallelizeable.
Furthermore we take a deeper look into the influence of the classical negation on the complexity by constraining the nesting depth within *MTL* formulas. Constraining the nesting depth to k, will lead to fragments which are Σ_k^p or Σ_{k+1}^p complete.

References

1. Ebbing, J., Lohmann, P.: Complexity of model checking for modal dependence logic. CoRR abs/1104.1034v1 (2011), http://arxiv.org/abs/1104.1034v1
2. Lohmann, P., Vollmer, H.: Complexity results for modal dependence logic. In: Proceedings 19th Conference on Computer Science Logic. Lecture Notes in Computer Science, vol. 6247, pp. 411–425. Springer Berlin / Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-15205-4_32
3. Väänänen, J.: Modal dependence logic. In: Apt, K.R., van Rooij, R. (eds.) New Perspectives on Games and Interaction, Texts in Logic and Games, vol. 4, pp. 237–254. Amsterdam University Press (2008)

Solving application oriented graph theoretical problems with DNA computing *

Veronika Halász, László Hegedüs, István Hornyák, and Benedek Nagy

Department of Computer Science, Faculty of Informatics,
University of Debrecen, Debrecen, Hungary
veronika@macko.atomki.hu, hegedus.laszlo@inf.unideb.hu,
ihornyak@namafia.atomki.hu, nbenedek@inf.unideb.hu

Important features of networks, e.g., length of shortest paths, centrality, are defined in a graph theoretical way. Bipartite graphs are frequently applied, not only in computer science (e.g. PetriNets), but in other sciences as well. They are used to represent various problems, for example, in medicine or in economy. The relations between customers and products can be represented by bipartite graphs. Genes and various diseases can also form a bipartite graph. Here we present a DNA computing approach for solving the mentioned graph theoretical problems. As a counterpart of [1], we present an algorithm that computes all shortest paths between all pairs of nodes in a graph (that may represent a social network, or some other network). From the results of the algorithm we can compute the centrality and eccentricity of a vertex, and also the centrality of an edge. In medical sciences bipartite graphs are used to denote the connection between diseases and causes, or genes and characteristics, etc. [2]. Thus scientists are interested in the direct paths from v_i to v_j, or all direct paths from v_i to all other vertices, or all direct paths between all vertices. We show a graph transformation algorithm: Starting from a bipartite graph we obtain its projection [3] by DNA computation, i.e., a graph with labeled edges having only vertices from one of the sets of the bipartition. Applications related to economy, e.g., marketing are also presented. We present another algorithm for projection that produces only paths of the desired length (and so it is faster and more efficient).

References

1. Hossein Eghdami and Majid Darehmiraki. Application of DNA computing in graph theory. *Artificial Intelligence Review* (2011), (online first)
2. Anna Bauer-Mehren, Michael Rautschka, Ferran Sanz, and Laura I. Furlong. DisGeNET: a Cytoscape plugin to visualize, integrate, search and analyze genedisease networks. *Bioinformatics* 26/22 (2010), 2924–2926.
3. Katharina Anna Zweig, Michael Kaufmann. A systematic approach to the one-mode projection of bipartite graphs. *Social Network Analysis and Mining* 1 (2011) 187–218.

* The work is supported by the TÁMOP 4.2.1/B-09/1/KONV-2010-0007 and TÁMOP-4.2.2/B-10/1-2010-0024 projects. The project is implemented through the New Hungary Development Plan, cofinanced by the European Social Fund and the European Regional Development Fund.

Computability, physics and logic

Ranjit Nair
Centre for Philosophy and Foundations of Science
E-36 Panchshila Park
New Delhi 110017
(e-mail: nair@cpfs.res.in)

The use of terms such as 'computability' and of Gödel's theorem by mathematical physicists hss drawn criticism from mathematical logicians who emphasize the precise bounds of these ideas. Within a formal system S with a finite number of axioms and syntactic rules of reasoning or inference rules, in which a certain amount of elementary arithmetic can be done, Gödel showed that there are undecidable sentences (first incompleteness theorem) and the consistency of S cannot be proved within S (second incompleteness theorem). Stephen Hawking conceded his first bet made in 1980 concerning the completeness of physics or the attainment of a 'theory of everything' (TOE) in 20 years in 2002, in 'Gödel's theorem and the end of physics'. The same bet was repeated with the author in 2001 and with the first concession on logical grounds it would appear *prima facie* that the second bet was also conceded. However, from a rigorous application of the incompleteness theorems, all that follows is that elementary arithmetic accompanying the putative TOE is incomplete. Hawking's conception of the centrality to self-referential sentences in Gödel's theorem which he maps on to physics turns out to be misleading as the theorems can be proved without invoking self-refentiality. Is there then no connection between incompleteness in Gödel's sense and in Hawking's sense? In my paper I suggest that the central feature shared is the complete absence of semantics and a dependence on syntactic rules. While it was not patently wrong for Gödel to have left semantics out entirely, theoretical physics as ordinarily understood, requires semantic interpretation. Perhaps, however, that is Hawking's point. Theoretical physics may be fundamentally of an algorithmic nature, without semantic interpretations. Arguably the Copenhagen interpretation of quantum mechanics was committed to such a view, hence the connections must not be so cavalierly dismissed.

[Topics: Philosophy of science and computation, Physics and computability]

An Undergraduate Course on the Life and Work of Alan Turing

Michael Olinick
Middlebury College
Middlebury, VT USA

May 25, 2012

Abstract: We describe a liberal arts course on the life and work of Alan Turing directed at first-year students with only a modest background in mathematics and computer science.
The initial segment focuses on ciphers in World War I and II, developing and teaching the tools of cryptology to understand the achievements of Turing and his Bletchley Park colleagues in cracking Enigma.
We continue with developing the mathematics needed to appreciate his historic paper "On Computable Numbers" and to create some elementary Turing machines. A third major section of the course examines the Turing Test and issues of artificial intelligence.
We conclude with a multimedia presentation on artistic responses to Turing's work and life including music, opera, poetry, and film. Course readings include biographies, novels, short stories, essays, and research papers.

Information Visualization for Cognitively Guided Health Risk Assessment

R. Padman[1], D.B. Neill[1], and C. Harle[2]

[1] The Heinz College, Carnegie Mellon University, Pittsburgh, USA. rpadman@cmu.edu, neill@cs.cmu.edu
[2] The University of Florida, Gainesville, USA. charle@phhp.ufl.edu

Global obesity rates have doubled since 1980 and are linked to higher rates of cardiovascular disease, cancer and diabetes, with an estimated 3 million deaths worldwide every year [1]. Assessing and responding to many patients' risks of chronic diseases and related complications are complex, high-dimensional, information processing problems faced by time-constrained clinicians. Innovative algorithms and tools which combine statistical machine learning, information visualization and electronic health data may reduce clinicians' information processing load and improve their ability to assess risk of disease onset and related complications. Information visualization utilizes the high bandwidth processing capabilities of the human visual system to more efficiently perform interactive data exploration and glean important insights [1]. A critical element in visualization is the incorporation of an expert user in the interpretation of data. This may make visualization methods particularly useful for the primary care setting where clinicians desire the flexibility of customizing assessments to the needs of their unique patient populations. To our knowledge, our research is the first study on computationally driven, contextualized, visualization techniques for improving chronic disease risk assessment at the point of care [2].

In this research, we explore statistical and machine learning methods commonly used for dimensionality reduction, including Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA) as a means for finding informative two-dimensional (2-D) projections and classifying patient data composed of arbitrary numbers of variables that are relevant to diabetes-related risk assessment [2]. Included in this step is the identification of appropriate data normalization procedures that conform with the selected dimensionality reduction methods and the disparate measurement of the data attributes. We establish a set of feasible methods for pre-processing and projecting high-dimensional patient data to 2-D plots so that multiple visual enhancements that may augment a user's analysis can be incorporated into the framework: (a) Procedures for overlaying decision boundaries that provide stratification into risk groups are defined using well-known classification techniques from the statistical machine learning literature; (b) Attracting anchor points and specifications for plotting them to intuitively reflect an attraction metaphor have been developed; (c) Additional use of color and/or shape that follow standard visual design techniques and may help highlight patient groups or important risk factors are specified as elements in the visual models. Results show that the framework may generate models which visually classify a large patient population with accuracy comparable to common statistical methods.

The primary goal is to develop intelligent visual data analysis tools that can be integrated with existing approaches to clinical data management and evaluation, in order to provide practitioners with usable systems that deliver critical information and new insights for responding to chronic disease risk among their patients. The methodology and tool proposed here offer an interactive interface through which clinicians can visually access, explore and compare risk predictions for a large cohort of patients in the context of many risk factors. This contextualization may make risk predictions more relevant, interpretable and clinically actionable. Aggregating results over patients, explaining the relationships between individual risk factors, and presenting information in a clinically useful format have not been sufficiently addressed in prior research. Such cognitively guided capabilities have the potential to move statistical risk models closer to the domain of primary care practice and the goal of meeting information needs to improve care quality. The proposed solutions may benefit multiple stakeholders.

References

1. B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. in: Proc. IEEE Visual Languages; 1996; Boulder, CO p. 336-343
2. C. Harle, D. Neill, R. Padman. An Information Visualization Approach to Classification and Assessment of Diabetes Risk in Primary Care, Proceedings of the 3rd INFORMS Workshop on Data Mining and Health Informatics (DM-HI 2008), J. Li, D. Aleman, R. Sikora, Eds.

The Turing Test and Computational Creativity

Alison Pease, Simon Colton, and John Charnley

Computational Creativity Group
Department of Computing, Imperial College
180 Queens Gate, London SW7 2RH, United Kingdom.
www.ccg.doc.ic.ac.uk

Computational Creativity is the AI subfield in which we study how to build computational models of creative thought in science and the arts. From an engineering perspective, it is desirable to have concrete measures for assessing the progress made from one version of a program to another, or for comparing and contrasting different software systems for the same creative task. The Turing Test is of particular interest to CC for two reasons. Firstly, unlike the general situation in AI, the TT, or variations of it, *are* currently being used to evaluate candidate programs in CC. Thus, the TT is having a major influence on the development of CC. This influence is usually neither noted nor questioned. Secondly, there are huge philosophical problems with using a test based on imitation to evaluate competence in an area of thought which is based on originality. While there are varying definitions of creativity, the majority consider some interpretation of novelty and utility to be essential criteria. For instance, one of the commonalities found by Rothenberg in a collection of international perspectives on creativity is that "creativity involves thinking that is aimed at producing ideas or products that are relatively novel",[1] and in CC the combination of novelty and usefulness is accepted as key. In Plucker and Makel list "similar, overlapping and possibly synonymous terms for creativity: imagination, ingenuity, innovation, inspiration, inventiveness, muse, novelty, originality, serendipity, talent and unique".[2] The term 'imitation' is simply antipodal to many of these terms.

In our talk we describe the Turing Test and versions of it which have been used in order to measure progress in Computational Creativity. We show that the versions proposed thus far lack the important aspect of interaction, without which much of the power of the Turing Test is lost. We argue that the Turing Test is largely inappropriate for the purposes of evaluation in Computational Creativity, since it attempts to homogenise creativity into a single (human) style, does not take into account the importance of background and contextual information for a creative act, encourages superficial, uninteresting advances in front-ends, and rewards creativity which adheres to a certain style over that which creates something which is genuinely novel. We further argue that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently untenable to apply any defensible version of the Turing Test.

As an alternative to Turing-style tests, we introduce two descriptive models for evaluating creative software, the FACE model which describes creative acts performed by software in terms of tuples of generative acts, and the IDEA model which describes how such creative acts can have an impact upon an ideal audience, given ideal information about background knowledge and the software development process.[3] These alternative measures constitute a beginning in our efforts to avoid some of the pitfalls of the TT: they do not discriminate against a creativity which may be specific to computers, they take contextual information into account via the framing aspect of the FACE model, they reward genuine advances in CC and the genuinely novel over pastiche. Perhaps most importantly, we believe that they are workable measures which will enable us to measure intermediate progress and make falsifiable claims about our programs. We demonstrate the practicability of the descriptive models with regard to a poetry generation system.[4]

---

[1] A. Rothenberg: *Creativity and Madness.* The John Hopkins University Press, Baltimore, USA, 1990.

[2] J. A. Plucker and M. C. Makel: *Assessment of creativity.* In J. C. Kaufman and R. J. Sternberg, editors, *The Cambridge Handbook of Creativity*, pages 48–73. Cambridge University Press, USA, 2010.

[3] S. Colton, A. Pease, and J. Charnley: *Computational Creativity Theory: The FACE and IDEA descriptive models.* In 2nd International Conference on Computational Creativity, 2011; and A. Pease and S. Colton: *Computational Creativity Theory: Inspirations behind the FACE and the IDEA models.* In 2nd International Conference on Computational Creativity, 2011.

[4] S. Colton, J. Goodwin, and T. Veale: *Full FACE poetry generation.* In Proceedings of the Third International Conference on Computational Creativity, 2012.

Summary of an ethnographic study of the third Mini-Polymath project

Alison Pease and Ursula Martin

{Alison.Pease, Ursula.Martin}@eecs.qmul.ac.uk
School of Electronic Engineering and Computer Science,
Queen Mary, University of London
Mile End Road London E1 4NS
UK

In his article *Computing Machinery and Intelligence*, Alan Turing proposed to consider the question, "Can machines think?". We consider the question, "Can machines do mathematics, and how?" Turing suggested that intelligence be tested by comparing computer behaviour to human behaviour in an online discussion. We hold that this approach could be useful for assessing computational logic systems which, despite having produced formal proofs of the Four Colour Theorem, the Robbins Conjecture and the Kepler Conjecture, have not achieved widespread take up by mathematicians. It has been suggested that this is because computer proofs are perceived as ungainly, brute-force searches which lack elegance, beauty or mathematical insight. One response to this is to build such systems which perform in a more human-like manner, which raises the question of what a "human-like manner" may be.

Timothy Gowers recently initiated Polymath, a series of experiments in online collaborative mathematics, in which problems are posted online, and an open invitation issued for people to try to solve them collaboratively, documenting every step of the ensuing discussion. The resulting record provides an unusual example of fully documented mathematical activity leading to a proof, in contrast to typical research papers which record proofs, but not how they were obtained.

We consider the third Mini-Polymath project, started by Terence Tao and published online on July 19, 2011. We examine the resulting discussion from the perspective: what would it take for a machine to contribute, in a human-like manner, to this online discussion? We present an account of the mathematical reasoning behind the online collaboration, which involved about 150 informal mathematical comments and led to a proof of the result. We distinguish four types of comment, which focus on mathematical concepts, examples, conjectures and proof strategies, and further categorise ways in which each aspect developed. Where relevant, we relate the discussion to theories of mathematical practice, such as those proposed by Pólya and Lakatos, and consider how their theories stand up in the light of this documented record of informal mathematical collaboration. We briefly discuss automated systems which currently find examples, form concepts and conjectures, and generate proofs; and conclude that Turing was right when he commented: "We can only see a short distance ahead, but we can see plenty there that needs to be done."

A Lower Bound on the Minimal Resources for Measurement-only Quantum Computation

Simon Perdrix

[1] CNRS
[2] LIG, University of Grenoble, France

We show a simple and conceptually new proof that the model of measurement-only quantum computation (MOQC) is a universal model of quantum computation. The simulation of unitary transformations (which are reversible) using measurements (which are not reversible) requires an additional working space. As a consequence, universal resources for the MOQC model are described as (i) a family of elementary observables and (ii) the number of ancillary qubits which are used to simulate any quantum evolution by composing measurements according to the elementary observables.

It has been proved that there exists a family of three observables which is universal using one ancillary qubit only [1], whereas there exists a family of two observables only which is universal in the presence of two ancillary qubits [2], leaving as an open question the existence of a universal family of two observables when a single ancillary qubit is available.

We introduce necessary and sufficient conditions for two successive measurements to be information preserving, which means that the action of the second measurement is reversible. Using this characterisation of information preserving measurements we prove that there is no family of two observables which is universal when a single ancillary qubit is available.

References

1. Simon Perdrix. *Towards minimal resources of measurement-based quantum computation.* New Journal of Physics, 9 206, 2007.
2. Yasuhiro Takahashi. *Simple Sets of Measurements for Universal Quantum Computation and Graph State Preparation.* Theory of Quantum Computation, Communication, and Cryptography. Lecture Notes in Computer Science Volume 6519, 2011, pp 26-34.

On the Formal Constructive Theory of Computable Functionals TCF$^+$

Iosif Petrakis and Helmut Schwichtenberg

Mathematisches Institut
Ludwig-Maximilians Universität
München 80333
petrakis@mathematik.uni-muenchen.de
schwicht@mathematik.uni-muenchen.de

Abstract

Our aim is to present the current stage of formation of the constructive formal theory of computable functionals TCF$^+$.

While the idea of a formal theory of computable functionals based on the partial continuous functionals as its intended domain goes back to Scott's LCF [Scott 1969/1993], Schwichtenberg's formal theory TCF [Schwichtenberg and Wainer 2012], a common extension of Plotkin's PCF and Gödel's system **T**, uses, in contrast to LCF, non-flat free algebras as semantical domains for the base types. These algebras are given by their constructors, which can be proved in TCF to be injective with disjoint ranges. Moreover, the underlying logic of TCF is minimal.

The passage from TCF to TCF$^+$ is forced by our need to have a formal theory better adjusted to the intended model. Since a partial continuous functional of a type ρ over some base algebras is an ideal of the corresponding information system C_ρ, we would like to represent within our formal theory not only the functionals themselves but also their finite approximations, i.e., tokens and formal neighborhoods contained in them. The system TCF$^+$ is such a formal theory first developed in [Huber et al. 2010].

We present an updated version of TCF$^+$ and of the proofs within TCF$^+$ of a generalization of Kreisel's density theorem and Plotkin's definability theorem. We also point to new case studies that could be examined within TCF$^+$.

Selected References

1. S. Huber, B. Karadais and H. Schwichtenberg: Towards a formal theory of computability, in R. Schindler (ed.) *Ways of Proof Theory: Festschrift for W. Pohlers*, 251-276, Ontos Verlag, 2010.
2. H. Schwichtenberg and S. Wainer: *Proofs and Computations*, Perspectives in Logic, ASL and Cambridge University Press, 2012.
3. D. Scott: A type-theoretic alternative to ISWIM, CUCH, OWHY, Manuscript, Oxford University, 1969; later published in Theoretical Computer Science, 121:411-440, 1993.

Computable C-classes

Sergei Podzorov

Sobolev Institute of Mathematics
Akad. Koptyug pr., 4, 630090, Novosibirsk, Russian Federation
podz@math.nsc.ru

Keywords: C-class, Many-One Reducibility, Computable Numbering, Reducibility of Numberings.

Let Λ be any finite nonempty set of indices. We say that $A = \{A_\lambda\}_{\lambda\in\Lambda}$ is *Λ-sequence* if $A \subseteq \mathbb{N}$ for any $\lambda \in \Lambda$. For any computable function φ and Λ-sequence A we write $\varphi^{-1}(A)$ for the Λ-sequence $\{\varphi^{-1}(A_\lambda)\}_{\lambda\in\Lambda}$. Given two Λ-sequences A and B we say that A is *m-reducible* to B if $A = f^{-1}(B)$ for some total computable function f.

Let Q be a class of Λ-sequences. Then Q is called *C-class* if Q contains universal Λ-sequence w.r.t. m-reducibility and $\varphi^{-1}(A) \in Q$ for any $A \in Q$ and any computable φ. C-classes are introduced in [1, ch. 3, §1]; it can be proved that the universal Λ-sequence in any C-class is unique up to computable isomorphism.

We say that $\nu = \{\nu_\lambda\}_{\lambda\in\Lambda}$ is a (computable) numbering of the class Q of Λ-sequences if ν_λ is (computable) numbering of some family of c.e. sets and $\{\nu x = \{\nu_\lambda x\}_{\lambda\in\Lambda} : x \in \mathbb{N}\} \subseteq Q$. For any two numberings ν and μ of Q we write $\nu \leqslant \mu$ if $\nu_\lambda = \mu_\lambda \circ f$ for any λ and some total computable function f. A computable numbering ν of Q is called *principal* if $\mu \leqslant \nu$ for any computable numbering μ of Q.

Let us call *an equality* any equality of the form $t_1(\lambda_1, \ldots, \lambda_k) = t_2(\lambda_1, \ldots, \lambda_k)$ where t_1, t_2 are the terms in the language with two binary operations $\cup$ and $\cap$, constant symbol $\varnothing$ and variables from Λ. If e is the equality of the such form and A is Λ-sequence we write $A \models e$ for $\mathbb{N} \models (t_1(A_{\lambda_1}, \ldots, A_{\lambda_k}) = t_2(A_{\lambda_1}, \ldots, A_{\lambda_k}))$. For any set Γ of the equalities we define $Q(\Gamma)$ to be the collection of all such Λ-sequences A that $A \models e$ for every $e \in \Gamma$ and A_λ is c.e. for any $\lambda \in \Lambda$.

For any finite poset P and any numbering α of P we say that α is *computable* if there exists computable function $\langle t, x\rangle \mapsto \alpha^t x \in P$ such that $\nu^0 x \geqslant \nu^1 x \geqslant \ldots$ and $\alpha x = \lim_t \alpha^t x$. A computable numbering of P is said to be *principal* if any computable numbering of P is reducible to it.

The author have proved that for any Γ the next conditions are equivalent: (i) the class $Q(\Gamma)$ of Λ-sequences is a C-class; (ii) there is a principal computable numbering of the class $Q(\Gamma)$; (iii) there is a principal computable numbering of P_Γ, where P_Γ is a special finite poset defined from Γ. It follows that $Q(\Gamma)$ is always a C-class; it also can be proved that any computable C-class is a class of the form $Q(\Gamma)$ for some suitable Γ.

References

1. Ershov, Yu. L.: Numbering theory. Nauka, Moscow (1977) [in Russian]

On Analogues of the Church-Turing Thesis in Algorithmic Randomness

Christopher P. Porter
christopher.p.porter@gmail.com

University of Notre Dame
Department of Mathematics
255 Hurley Hall
Notre Dame, IN 46556

Abstract. According to the Church-Turing Thesis, the informal notion of an effectively calculable function has the same extension as the notion of a Turing computable function. Is there an analogue of the Church-Turing Thesis that holds for some definition of algorithmic randomness for infinite sequences? While several analogues have been suggested, I will argue (i) that each of these suggestions is problematic, and (ii) that, rather than single out one definition of algorithmic randomness as capturing the so-called intuitive conception of randomness, a more promising approach is one according to which multiple non-equivalent definitions of algorithmic randomness play an important role in illuminating the concept of randomness.

Keywords: the Church-Turing Thesis, algorithmic randomness, philosophy of mathematics

Computational model based on meteorological data for forecast of influenza-like activity

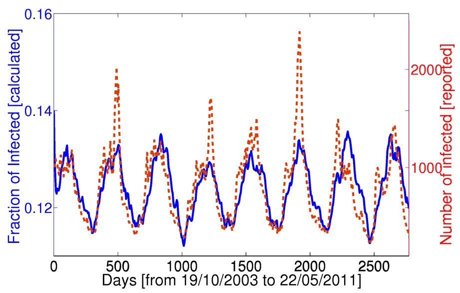
Eugene B. Postnikov

Theoretical Physics Department, Kursk State University,
Radishcheva 33, 305000 Kursk, Russia
`postnicov@gmail.com`

Although the seasonality of respiratory diseases (common cold, influenza, RSV, etc) is a quite ubiquitous phenomenon, the development of computational models, which allow to reproduce it is still an actual unsolved problem of mathematical epidemiology and dynamical systems theory.

Basing on known in medical literature correlation between activity influenza-like diseases and meteorological (air temperature, humidity, illumination) conditions, the SIRS (Susceptible–Infected–Recovered– Susceptible) approach is used $\dot{S} = -kIS + \theta^{-1}R, \dot{I}dt = kIS - \tau^{-1}I$, $\dot{R} = \tau^{-1}I - \theta^{-1}R$ $(S + I + R = 1)$ with the variable parameter $k = k_0 \left[1 + \kappa\left(T(t)\right)\right]$, which depends on air temperature $T(t)$ daily varying during seasons.

It has been shown that that this system can be transformed into the second-order non-autonomous ODE with free the term $R_s\theta^{-1}\tau^{-1}\kappa\left(T(t)\right)$, where R_s is a fixed point for R in the case $k = k_0 = \text{const}$. In other words, the proposed coordinate transformation reveals the explicit form of outer excitation, which enforce its intrinsic period to the epidemic oscillations and determines the shape of latter as a kind of resonant filtering. As well, this reveals the origin of co-called dynamical resonance in stochastic SIRS model [Dushoff et al., 2004], since the phenomenon of 1:1 resonant excitation between intrinsic oscillations and single-frequency oscillations of the reaction rate.



To validate the obtained model, the data on flu dynamics obtained from `Google Flu Trends` are compared with the forecast evaluated via proposed ODE system, where corresponding weather conditions (daily mean temperature) are taken from `European Climate Assessment & Dataset`. The processing of these curves confirms the proposed mathematical model, see, for example, the Figure above quite satisfactory reproducing normal flu level in Berlin (localized significant exceptions corresponds to avian and swine flus outbreaks). Thus, these results open the opportunities for automated computational forecast of normal estimated flu activity using both data of current trends and weather forecast.

A verificationist modal language for contextual computations

Giuseppe Primiero

FWO – Research Foundation Flanders
Centre for Logic and Philosophy of Science
University of Ghent (Belgium)
Blandijnberg 2, room 231, 9000 Gent
IEG, University of Oxford (UK)
`Giuseppe.Primiero@UGent.be`

The relation between modal logic $S4$ and intuitionistic logic is notoriously given via the interpretation of necessity as provability. Constructive $S4$ has been explored in the form of both Kripke and categorical semantics. Less considered in the literature are modal translations of the contextual notion of derivability known from natural deduction calculi and type theories. Even less so are languages including *both* local and global validity relations, roughly corresponding to the idea of derivability from undischarged and discharged assumptions.

In this paper, we present a modal language for contextual computing, corresponding to the fragment of constructive KT with necessity and possibility operators to interpret absolute and contextual computations as different modes of verifying the truth of propositions. Its semantics $\mathcal{L}_{cc}$ is given as the union of two fragments: $\mathcal{L}^{ver}$ for absolute computations has formulas verified in a set of knowledge states with models defined by an order relation on them and a verification function; the extension to $\mathcal{L}^{ctx}$ is obtained by introducing an appropriate notion of contextual verification, simulating truth under contents that are not directly computable in $\mathcal{L}^{ver}$, but are considered admissible. The pre-order on states is now strictly determined by the inclusion relation of contexts for states. An informational context Γ over K_i is admitted if and only if the evaluation functions it induces are not contradictory to any of the contents validated in $\mathcal{L}^{ver}$ for any knowledge state from which K_i is accessible. Modalities are used to express extensions of contexts as sets of such assumptions in order to define local and global validity. This semantics has a (weak) monotonicity property, depending on satisfaction of processes in contexts. In the corresponding axiomatic system $cKT_{\Box\Diamond}$, a restricted version of the deduction theorem for globally valid formulas holds. Soundness and completeness are proven and decidability is shown to hold for the necessitation fragment of the language by an additional restricted finite model property.

$\mathcal{L}_{cc}$ and the axiomatic counterpart $cKT_{\Box\Diamond}$ are inspired by applications of logics for modeling knowledge processes in the context of exchange of *unverified* or *uncertain* information. Possible applications are in modeling of trusted communications of uncertain information and contextual verification methods in distributed and staged computation.

The Computational Strengths of α-length Infinite Time Turing Machines

Benjamin Rin[1]

Logic and Philosophy of Science, University of California, Irvine - Irvine CA, USA
brin@uci.edu

Abstract. In [2], open questions are raised regarding the computational strengths of so-called ∞-α-Turing machines, a family of models of computation resembling the infinite time Turing machine (ITTM) model of [1], except with α-length tape (for $\alpha \geq \omega$). Let T_α refer to the model of length α (so T_ω is just the ITTM model). I define a notion of computational strength, letting $\succ$ stand for "is computationally stronger than". I show the following: (1) $T_{\omega_1} \succ T_\omega$. (2) There are countable ordinals α such that $T_\alpha \succ T_\omega$, the smallest of which is precisely γ, the supremum of clockable ordinals (by T_ω). In fact, there is a hierarchy of countable machines of strictly increasing strength corresponding to the transfinite (weak) Turing jump operator ∇. (3) There is a countable ordinal μ_0 such that for every countable $\mu \geq \mu_0$, neither $T_{\omega_1} \succ T_\mu$ nor $T_\mu \succ T_{\omega_1}$ — that is, the machines T_μ and T_{ω_1} are computation-strength incommensurable. The same holds true if T_{ω_1} is substituted by any larger machine.

Keywords: infinite time Turing machine, ITTM, transfinite computation, supertask, computability

References

1. Hamkins, J., Lewis, A.: Infinite Time Turing Machines. Jour. Sym. Log. 65, 567–604, (2000)
2. Koepke, P.: Ordinal Computability. Lecture Notes in Comput. Sci., 5365, 280–289 (2009)

Two New Reducibilities Based on Enumeration Orders

Safilian Aliakbar[1] and Didehvar Farzad[2]

[1] safiliaa@mcmaster.ca
Department of Computing and Software, McMaster University, Hamilton, Canada
[2] didehvar@aut.ac.ir
Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran

Abstract. Various works have been done in computability theory to contrast and compare decidable sets and computable enumerable sets. As a simple fact, we know that among all enumerable sets, only decidable sets can be sorted computably in an ascending usual order.
The question that naturally arises for any c.e. set A is "What computable partial orders are accepted by A?"In this paper, the attempts are made to expose a theory to answer the above question.
Some other questions arise in the shadow of the above question, the most important ones are listed in the following:
1. How to investigate c.e. sets which accept the same partial orders in the above sense?
2. Consider two non-decidable c.e. sets A and B. Is it possible the partial orders accepted by A be the subset of partial orders accepted by B?
Throughout this paper, we show that answering the first question requires defining an equivalence relation and to answer the second one we have to define a partial ordering among c.e. sets. Therefore, a reducibility among c.e. sets named Enumeration Order Reducibility and the related equivalence relation are defined.
In continuing, we found out that adding some finite numbers of elements to a non-decidable c.e. set may results a set that is not in the enumeration order equivalence class of the origin set. Regarding this point, some complexities arise. To overcome them, we define a new type of reducibility among c.e. sets.
The main goal of the present paper, is to find some properties of the introduced relations among c.e. sets and compare them to the other famous relations in computability theory. As an example, we show that these partial orderings are finer than Turing reducibility.

Nonstandard Analysis: a New Way to Compute

Sam Sanders

Department of Mathematics, Ghent University
sasander@cage.ugent.be

Abstract

Constructive Analysis, introduced in 1967 by Errett Bishop ([1]), is the redevelopment of Mathematics based on *algorithm* and *proof* along the lines of the famous BHK (Brouwer-Heyting-Kolmogorov) interpretation. *Constructive Reverse Mathematics* is a spin-off from Harvey Friedman's famous *Reverse Mathematics* program ([8]), based on Constructive Analysis ([3, 4]).

We identify a fragment of Nonstandard Analysis ([6]) which captures Bishop's Constructive Analysis. The counterparts of *algorithm* and *proof* in Nonstandard Analysis are played by *Ω-invariance* and *Transfer*. Transfer expresses Leibniz' law that $\mathbb{N}$ and ${}^*\mathbb{N}$ satisfy the same properties. Furthermore, an object is *Ω-invariant* if it does not depend on the *choice* of the infinitesimal in its definition. Incidentally, the latter is exactly the way infinitesimals are used in Physics. Moreover, the latter discipline tends to limit itself to Mathematics formalizable in Constructive Analysis ([2]).

We obtain a large number of equivalences from Constructive Reverse Mathematics in our constructive version of Nonstandard Analysis and discuss implications of our results. In particular, we discuss how our approach is the dual of Palmgren and Moerdijk ([5]) towards *Reuniting the Antipodes* ([7]).

References

[1] Errett Bishop, *Foundations of constructive analysis*, McGraw-Hill Book Co., New York, 1967.
[2] Fred Richman and Douglas Bridges, *A constructive proof of Gleason's theorem*, J. Funct. Anal. **162** (1999), no. 2, 287–312.
[3] Hajime Ishihara, *Reverse mathematics in Bishop's constructive mathematics*, Philosophia Scientiae (Cahier Spécial) **6** (2006), 43-59.
[4] ———, *Constructive reverse mathematics: compactness properties*, From sets and types to topology and analysis, Oxford Logic Guides, vol. 48, 2005.
[5] Ieke Moerdijk and Erik Palmgren, *Minimal models of Heyting arithmetic*, J. Symbolic Logic **62** (1997), no. 4, 1448–1460.
[6] Abraham Robinson, *Non-standard analysis*, North-Holland, 1966.
[7] Peter Schuster, Ulrich Berger, and Horst Osswald (eds.), *Reuniting the antipodes: constructive and nonstandard views of the continuum*, Synthese Library, vol. 306, Kluwer Academic Publishers, Dordrecht, 2001.
[8] Stephen G. Simpson, *Subsystems of second order arithmetic*, 2nd ed., Perspectives in Logic, Cambridge University Press, Cambridge, 2009.

Shape a Language for $\mathcal{IFS}$

Anargyros Sarafopoulos

NCCA, Bournemouth University,
Talbot Campus, Fern Barrow, Poole UK
{asarafop}@bournemouth.ac.uk

Abstract. *Shape* is a language for controlling agents designed to represent geometric transformations of compact subsets in a metric space. *Shape* provides a spatial reasoning[6] framework which allows agents to take decisions based on the structure of their surrounding metric. Like biological cells, these agents can subdivide and replicate creating new transformations and new subsets, while their control program or "genetic code" is written in *Shape*. *Shape* is demonstrated on the inverse problem for Iterated Function Systems (IFS) fractals[1]. *Shape* provides a useful computational framework for studying development of fractal structures, as well as a novel indirect representation[2, 4] for spatial generative models[5, 3].

Keywords: Evolutionary Representations, IFS Fractals, Agents, Biologically Inspired Computing, Computer Graphics, Spatial Reasoning

References

1. M. F. Barnsley. *Fractals Everywhere: 2nd edition.* Academic Press, 1993.
2. P. Bentley and S. Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In *gecco-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 35–43. 4th Annual Genetic Programming Conference, Morgan Kaufmann, 1999.
3. P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants.* The Virtual Laboratory. Springer, 1996.
4. A. Sarafopoulos and B. Buxton. Empirical comparison of evolutionary representations of the inverse problem for iterated function systems. In M. Ebner, M O'Neill, Anikó Ekárt, L. Vanneschi, and A. I. Esparcia-Alcázar, editors, *Proceedings of the 10th European Conference on Genetic Programming*, volume 4445 of *Lecture Notes in Computer Science*, Valencia, Spain, 2007. Springer-Verlag.
5. John M. Snyder and James T. Kajiya. Generative modeling: a symbolic system for geometric modeling. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, pages 369–378, New York, NY, USA, 1992. ACM.
6. J. Van Benthem. *Modal Logic for Open Minds.* Csli Studies in Computational Linguistics. University of Chicago Press, 2010.

The ω-Turing degrees

Andrey Sariev

Faculty of Mathematics and Computer Science, Sofia University,
5 James Bourchier Blvd., 1164 Sofia, Bulgaria,
acsariev@gmail.com

Abstract. In this paper the study of the partial ordering of the ω-Turing degrees is initiated. Informally, the considered structure is derived from the structure of ω-enumeration degrees described by Soskov [1] by replacing the usage of the enumeration reducibility and the enumeration jump in the definitions with Turing reducibility and Turing jump respectively. The main results include a jump invertion theorem, existence of minimal elements and minimal pairs.

Keywords: ω-enumeration degrees, Turing reducibility, jump hierarchies

References

1. I.N. Soskov: The ω-enumeration degrees. Journal of Logic and Computation, to appear.
2. I.N. Soskov, H. Ganchev: The jump operator on the ω-enumeration degrees. Annals of Pure and Applied Logic, to appear.

Structural versus Quantitative Characteristics of Information in Information Processing: Information Integration

Marcin J. Schroeder

Akita International University, Akita, Japan

Abstract. Discussions of Church-Turing thesis, as well as that of hypercomputation have only fragmentary foundations. Although the concept of computation is usually clearly defined, the issue of what is beyond computation or what can be reduced to computation remains vague. It seems obvious that any discussion of the limits of computation has to be formulated in a conceptual framework broader than that of computation. It is commonly assumed that such a more general framework can be found in the concept of information, and computation can be considered a kind of information processing. However, there is no consensus regarding definition of information, and its study focused thus far exclusively on its quantitative characteristics, such as Shannon's entropy or algorithmic measures of Kolmogorov-Chaitin. In the context of computation, the latter approach is closing a vicious circle. Computation is supposed to be explained in terms of information processing. Information is being studied in terms of computation. Even more serious problem in using information as a fundamental concept for the study of limits for computation was the lack of well developed structural theory of information. Present paper is presenting an approach to the study of information processing based on the concept of information and its mathematical formalism developed in earlier works of the author. Its formalism, formulated in terms of closure spaces and their general algebraic analysis, has an advantage of uniting the selective and structural manifestations of information. Another advantage, relevant to the issue of the limits of computation, is the possibility to employ Tarski's consequence operator (specific example of a closure space with the finite character property) as a tool for the analysis of the role of logic in the process of computing. Then, we can recognize another vicious circle in the discussion of the limits of computation. Failing attempts to go beyond Turing's computation have been always assuming that the description of what could be beyond computation has to be formulated in terms of classical, linguistic logic which is based on closure operator of finite character. However, such an assumption automatically precludes non-computability. An alternative is to consider logic of more general forms of information. Moreover, information integration, which is considered as defining process of consciousness, involves closure spaces which are not necessarily of finite character. This opens a very broad perspective on processes which are beyond Turing's computability.

Keywords: Information, Information Integration, Information Processing, Computation, Logic of information

Who is the Human Computer?

Oron Shagrir

Philosophy and Cognitive Science
The Hebrew University, Jerusalem 91905, Israel

Alan Turing (1936) famously provided an analysis of effective computability by formulating a set of restrictive constraints ("axioms") on the computing agent. It is widely agreed that Turing had in mind an idealized human agent, and that he "makes no reference whatsoever to calculating machines" (Gandy 1988: 83-84). My aim is to explore two very different understandings of the concept of a human computer; I call them the cognitive and the non-cognitive approaches. According to the cognitive approach, a human computer is restricted by the limitations of certain human cognitive capacities. The claim need not be that these limitations apply to human mental processes in general, but to the cognitive abilities involved in calculation. The non-cognitivist, in contrast, thinks that a human computer is restricted to certain finite means, regardless of whether or not these means reflect the limitations of human cognitive capacities. These means are simply part of the concept of effective computation as it is properly used and as it functions in the discourse of logic and mathematics.

The restrictive constraints that Turing formulated can be respectively understood in two ways. The cognitivist might see them as reflections of certain limitations of human cognitive capacities. These limitations ground or justify the restrictive constraints. According to the cognitive approach, computability is constrained by Turing's constraints because these restrictive constraints reflect the limitations of human cognitive capacities. The non-cognitivist thinks that the restrictive constraints do not, and need not, necessarily reflect cognitive limitations. The non-cognitivist offers no other justification for the constraints. In fact a call for further justification has no place at all in the analysis of computability, according to the non-cognitivist.

I will argue that the founders of computability and their interpreters take a stand between the approaches.

Closure Properties of Quantum Turing Languages *

Yun Shang[1,2], Xian Lu[1], and Ruqian Lu[1,2,3]

[1] Institute of Mathematics, AMSS, CAS, Beijing, P.R.China
[2] NCMIS, AMSS, CAS, Beijing, P.R.China
[3] CAS Key Lab of IIP, Institute of Computing Technology, CAS, Beijing, P.R.China
shangyun602@163.com, luxian@amss.ac.cn, rqlu@math.ac.cn

In classical computing theory, recursively enumerable languages are known as type-0 languages and accepted by Turing machines. These languages enjoy a good set of well-defined closure properties. It has been proved that the Kleene * of a recursively enumerable language is a recursive enumerable language again. Also the intersection, union and concatenation of two recursive enumerable languages are recursive enumerable again.

Are all these results still valid when shifting from languages based on classical logic to those based on quantum logic? We have found that the answer is no. While the underlying algebra of classical logic is Boolean algebra, that of quantum logic (in the sense of von Neumann) is orthomodular lattice where the distributive law fails to hold. The question as to whether or not a symbol string is a sentence of a language does not have an answer of yes/no type as in the classical case. Rather, the answer of such questions now takes an element of the orthomodular lattice as its value. Unfortunately, only the union operation remains valid in the (orthomodular) lattice version. As for intersection, concatenation and Kleene * operations of two quantum recursive enumerable languages, we have proved that they are not recursive enumerable in lattice valued sense. Similar to classical complement operation, orthocomplement operation for orthomodular lattices does not enjoy the closure property.

Things become more complicated if we come from the (sharp) quantum logic over to unsharp quantum case where the underlying algebraic model is extended lattice-ordered effect algebra (or lattice-ordered QMV algebra). The situation is more serious since also the non-contradiction law and excluding of the middle law both fail to be valid. As a positive result we have proved that the closure property is true for intersection operation. This is different from the case of orthomodular lattice. On the other hand we have also proved that the closure property for disjoint sum (that is some generalized union), concatenation, Kleene * and orthocomplement operation does not hold. The first three properties are true if and only if the truth valued lattice satisfies some kind of distributive law.

From the above results, we can conclude that the underlying physical properties determine the quantum logic (sharp or unsharp), while the latter determines the algebraic models of quantum Turing machine.

* This work was supported by NSFC projects under Grant No.61073023. 973project under Grant NO. 2009CB320701,K.C.Wong Education Foundation, Hong Kong, National Center for Mathematics and Interdisciplinary Sciences, CAS

Geometric Models of Computation

by Joseph Shipman and John H. Conway

We describe a general framework for models of computation that generate numbers from previously generated numbers by geometrical operations. Most of these involve "construction tools" that find a subset of the roots of polynomials whose coefficients lie in fields generated by previously constructed numbers. In this view, numbers are regarded as "constructible" only if they are given by a finte sequence of such operations, rather than by infinite processes involving limits. Complex numbers are considered to be constructed if their real and imaginary parts have been.

Constructions may involve restrictions or extensions of the classical Greeks' tools of ruler and compass and technique of neusis or "verging", and later techniques such as origami, nomograms, rigid frameworks, curve-drawing tools, and linkages. We define complexity measures in terms of the number of operations of various types needed to complete a construction, and compare general constructions involving a bounded number of steps with non-uniform constructions of particular numbers. We establish some results about angle n-section, nth-rooter tools, regular polygons, and general nth-degree polynomial equations.

The subfields of the algebraic numbers associated with such models can be characterized in various ways, such as degrees of extensions, cyclotomy, commutativity and solvability and composition factors of Galois groups, closedness under taking real or complex algebraic conjugates, or number of terms in permitted polynomials. The geometrical operations permitted in such a model can be characterized in additional ways, such as number of degrees of freedom in the configuration space of the construction tools, and the types of curves that can be drawn explicitly (rather than implicitly characterized by points on them).

We will demonstrate relationships between these characterizations, and extend them to non-geometrical operations such as price-yield functions and ultraradicals. Questions of definability, decidability and (classical) computational complexity are investigated. Extensions to transcendental fields are possible using techniques such as generating pi by rolling a paper cone in three dimensions, but we must then assume Schanuel's conjecture in order to prove basic results. We also discuss extensions to characteristic p and the usefulness of having a computable order relation, and other results by the authors relating the degrees of polynomials with no constructible roots. A historical survey connects this work to earlier results of Alperin, Baragar, Demaine, Gleason, Kempe, Lang, Peaucellier, et al.

The use of language and logic in computing technology

Lindsay Smith, Vito Veneziano, Paul Wernick

University of Hertfordshire, UK

Introduction: An emerging perception of computing as technology rather than science (Baldwin 2011) is based on a view that emphasises the role of tool use in computing.
Background and the problem domain: Any new technological development is subject to unpredictable circumstances, and computing is no exception. For example, the question as to whether computing can be explained in scientific terms is still unanswered and perplexing. Nevertheless, computing technology plays – and seems likely to continue to play – a 'game-changing' role in the state of human affairs. The source of the associated 'game change' will inevitably determine a particular technological turn of events. We assume that any significant change in the technologies in use is likely to be a source of change in the world in which it operates. In particular, the need for software makes computer-based technologies different from others. Software exists, and is constructed, exclusively in a specific artificial, non-natural, environment. Computer-based tools take this "context-insensitive" software and place it into human 'context sensitive' environments and set them to work in (assumed) harmony with other tools; software-based in-car satellite navigation software travels along the road with the car driver, courtesy of the car's wheels. Computing technology has thus split human use of artefacts from their construction environment(s). This restricts our ability to improve technology by shifting focus between tool use and construction. As a result, it can be very difficult to be certain that software, correct in a construction environment will prove to be 'correct' when used in an artefact.
We use conflicting paradigms of computing to work out a logical analysis of the problem domain: The above analysis has identified a potential technologically-based mismatch between the environments of software construction and use. This 'mismatch' is not explicit in computing paradigms and has provoked disputes over the ontological status and valid purpose(s) of programming languages (PL). The differing constraints acting on PL as against natural languages (NL), for example, only implicitly appear in an extensive literature review and analysis of computing paradigms (Eden 2007). Our approach to increasing an understanding of PL (software) is pragmatic, using logical analysis to counterpoise two opposing views of PL, rationalist and technocratic (Eden 2007). Due to the lack of previous work in this area, we have opposed 21st-century computing concepts – the classes and objects used in object-oriented programming (OOP) – with the 20th-century philosophical failure to demonstrate mathematical systems as inevitably logically self-consistent (Gefwert 1998).
Conclusion: By considering "Russell's Antinomy" (Gefwert 1998), we conclude that "inconsistency-tolerance" (Decker 2009) in PL is a hypothetical possibility for PLs and systems developed using them. This tends to support technocratic views of computing that are directly opposed by a rationalist concern for consistency that tends, for instance to value programs that have been formally specified over those that have not. We will align the technocratic view of PL with Wittgenstein's view on paradoxes (Chihara 1977), and the opposing rationalist view of PL with Russell (and Turing), as the basis for extending our analysis.

References

Baldwin, D. (2011), 'Is computer science a relevant academic discipline for the 21st century?', *Computer* **44**(12), 81 –83.

Chihara, C. (1977), 'Wittgenstein's analysis of the paradoxes in his lectures on the foundations of mathematics', *The Philosophical Review* **86**(3), 365–381.

Decker, H. (2009), How to contain inconsistency, or why wittgenstein only scratched the surface, *in* 'E-CAP 2009 European conference on Computing and Philosophy'.

Eden, A. (2007), 'Three paradigms of computer science', *Minds and Machines* **17**(2), 135–167.

Gefwert, C. (1998), *Wittgenstein on mathematics, minds and mental machines*, Ashgate Aldershot.

Smith, L. (2009), Meeting stakeholder expectations of software, or looking for the 'black swan' in software requirements, *in* 'E-CAP 2009 European conference on Computing and Philosophy'.

P systems controlled by topological spaces

Erzsébet Csuhaj-Varjú[1], Marian Gheorghe[2,3], and Mike Stannett[2]

[1] Department of Algorithms and Their Applications, Faculty of Informatics, Eötvös Loránd University, Pázmány Péter st. 1/c, 1117 Budapest, Hungary
csuhaj@inf.elte.hu
[2] Department of Computer Science, The University of Sheffield, Regent Court, 211 Portobello, Sheffield S1 4DP, United Kingdom
{M.Stannett, M.Gheorghe}@dcs.shef.ac.uk
[3] Department of Computer Science, University of Piteşti
Str Targu din Vale, Piteşti, Romania

Membrane computing is a vivid research field in natural computing and unconventional computing, studying a computational paradigm including a large variety of models, called membrane systems or P systems. During the years, the framework has been intensively investigated, especially with respect to its computational power and complexity aspects. Promising applications of the theory in biology, distributed computing, linguistics, graphics have also been identified.

A P system consists of a set of compartments i.e., regions delimited by membranes, where the regions contain multisets of objects. In the generic model, the compartments form a nested, tree-like structure. There are transformation and communication rules associated to the regions for describing the interactions between the objects and a strategy of evolving the system. It can easily be seen that P systems introduce in a very natural way a specific topology of the system described, whereby membranes define compartments with local objects and interacting rules, together with specific links between compartments. These links describe communication channels allowing connected compartments to exchange objects. Although this topology is flexible enough to model various natural or engineering systems, there are cases when a fine grain topological structure is requested.

In this paper we continue our previous investigations using topological spaces as a control mechanisms for membrane systems, where a topological space was used as a framework to control the evolution of the system with respect to a family of open sets that is associated with each compartment. This approach provides a fine grain description of local operations occurring in each compartment by restricting the interactions between objects to those from a certain vicinity. Continuing the initial study on the influence of an arbitrary topology on the way basic membrane systems compute, we examine the effect of the topological space control on P systems with different static and dynamic underlying graph structure and with different topologies. More precisely, we study topologically controlled computation for purely communicating P systems (symport-antiport), tissue P systems and population P systems, as well as P systems with membrane division and creation.

Linear Aspects of Intersection

Anastasia Veneti[1,⋆] and Yiorgos Stavrinos[2,⋆⋆]

[1] Dept. of History and Philosophy of Sciences, U. of Athens, 15771 Zografou, Greece, tassiana98@gmail.com

[2] Graduate Program in Logic and Algorithms (*ΜΠΛΑ*), Dept. of Mathematics, U. of Athens, 15784 Zografou, Greece, g.stavrinos@gmail.com

Linear logic deals with two kinds of conjunction, *multiplicative* $\otimes$ and *additive* $\&$. For these connectives both sequents $A \otimes B \Rightarrow A \& B$ and $A \& B \Rightarrow A \otimes B$ are not derivable. Another system with two kinds of conjunction (and disjunction) is *Intersection and Union Logic* IUL [3,6,5] which aims to give a logical foundation for intersection and union types [1]. In IUL, conjunction $\wedge$ is an *asynchronous* connective and has a *multiplicative* definition whereas intersection $\cap$ being *synchronous* is necessarily *additive* [3]. For these connectives, $A \wedge B \Rightarrow A \cap B$ is not derivable whereas $A \cap B \Rightarrow A \wedge B$ is and thus intersection $\cap$ behaves as a special *synchronous* conjunction.

To investigate further the nature of these connectives we define a translation $(\_)^\circ$ of IUL in linear logic and prove a full embedding. In the translation of $\cap$ we take into account the *additive* aspect of $\cap$ and therefore $(A \cap B)^\circ = A^\circ \& B^\circ$, whereas for $\wedge$ and $\otimes$ we consider *general elimination rules* [4,2] and thus we translate $(A \wedge B)^\circ = !A^\circ \otimes !B^\circ$. Since $!A \otimes !B \Rightarrow A \& B$ is derivable (and not conversely), the embedding with $\&$ is in a sense tighter than the embedding with $\otimes$, which is dual to the relation of $\cap$ to $\wedge$.

Keywords: intersection and union types, intersection and union logics, linear logic

References

1. Barbanera F., Dezani-Ciancaglini M., and de'Liguoro U., Intersection and Union Types: Syntax and Semantics, *Information and Computation* 119, 202–230 (1995)
2. Negri S., Varieties of linear calculi, *Journal of Philosophical Logic* 31(6), 569–590 (2002)
3. Pimentel E., Ronchi Della Rocca S., and Roversi L., Intersection Types: a Proof-Theoretical Approach, *Technical Reports of Structures and Deduction*, ICALP'05 workshop, 189–204 (2005)
4. von Plato J., Natural deduction with general elimination rules, *Archive for Mathematical Logic* 40, 541–567 (2001)
5. Stavrinos Y. and Veneti A., Towards a logic for union types, *Fundamenta Informaticae: special issue on Intersection Types and Related Systems*, accepted for publication (2012)
6. Veneti A. and Stavrinos Y., Towards an intersection and union logic, *4th Workshop on Intersection Types and Related Systems*, Torino (2008)

⋆ Author co-financed by the European Union (ESF) and Greek national funds through the Funding Program "Heracleitus II".

⋆⋆ Author partially supported by grant SARG 70/3/11098 of the University of Athens.

Computational Models as Mechanistic Explanations

Catherine Stinson[1,2]

[1] Universität Tübingen, Centre for Integrative Neuroscience
Ottfried-Müller-Str. 25, 72076 Tbingen, Germany
catherine.stinson@cin.uni-tuebingen.de
[2] University of Pittsburgh, Department of History & Philosophy of Science
1017 Cathedral of Learning, Pittsburgh, PA 15260, USA

Abstract. I examine computational modeling in cognitive neuroscience with the joint aims of expanding the scope of research on the epistemology of computational modeling, and resolving some outstanding problems concerning the role of computational models in cognitive science.
The literature on the philosophy of simulation and modeling has focused predominantly on examples taken from physics or economics. In fields like cognitive neuroscience, computational models bear a different relationship to theory, for the simple reason that there are no fundamental theories in cognitive neuroscience.
Discussions of computational modeling in the cognitive sciences have largely ignored epistemological questions. Nevertheless, some epistemological puzzles can be extracted from the literature on the relative merits of classical and connectionist AI architectures. There is an unresolved tension in these discussions between computational models as bits of mathematical theory and as physical implementations. There is also a confusing mixture of claims about neural plausibility being a virtue, while at the same time, simplicity and idealization are deemed essential.
I show how these apparently contradictory pairs of roles and desiderata can be reconciled. In particular, if the neo-mechanist picture of explanation is adopted, the connectionist project appears quite coherent. Mechanisms are simultaneously theory-like explanatory apparatus, and physical implementations. Furthermore, a mechanism need not include details at all levels of analysis in order to serve its explanatory purpose. They are nearly always sketchy or schematic. The result is a pluralist picture where computational models play a variety of roles from being theories, to aiding in the construction of theories, to being tools that evaluate theories. I discuss some recent examples of computational modeling work in cognitive neuroscience that illustrate these points.

Keywords: computational models, connectionism, mechanistic explanation

The Analysis of Evolutionary Algorithms: Why Evolution is Faster With Crossover

Dirk Sudholt

Department of Computer Science, University of Sheffield
211 Portobello, Sheffield S1 4DP, UK
d.sudholt@sheffield.ac.uk

Abstract. Evolutionary algorithms represent a natural computation paradigm, mimicking the natural evolution of species in order to "evolve" good solutions for optimisation problems. Starting with a random "population" of candidate solutions, search operators like mutation and crossover are used to construct new, innovative solutions, while selection operators drive this artificial evolution towards favouring solutions with high "fitness", i. e., solution quality. Evolutionary algorithms are very popular in many engineering disciplines as they are generic, easy to apply, and often perform surprisingly well. They are a method of choice in black-box optimisation, where the knowledge about the problem at hand is too limited to design a tailored algorithm.

In the past decades there has been a long and controversial debate about when and why the crossover operator is useful. The so-called "building-block hypothesis" assumes that crossover is particularly helpful if it can recombine good "building blocks", i. e. short parts of the genome that lead to high fitness. However, all attempts at proving this rigorously have been inconclusive. As of today, there is no rigorous and intuitive explanation for the usefulness of crossover.

In this talk we provide such an explanation. For functions where "building blocks" need to be assembled, we prove rigorously that a simple evolutionary algorithm with crossover is twice as fast as the fastest evolutionary algorithm using only mutation. The reason is that crossover effectively turns fitness-neutral mutations into improvements by combining the right building blocks at a later stage. This also leads to surprising conclusions about the optimal mutation rate.

Keywords: Natural computation, evolutionary computation, genetic algorithms, runtime analysis, recombination, crossover, theory

Iterative Quantum Tree Search

Luís Tarrataca and Andreas Wichert

GAIPS/INESC-ID
Technical University of Lisbon
{luis.tarrataca,andreas.wichert}@ist.utl.pt

Abstract. We discuss an $O(\sqrt{b^d})$ iterative extension to Grover's algorithm focused on tree search.

Whilst the appropriateness of quantum random walks for searching a graph $G = (V, E)$ seems fairly natural, the application of Grover's algorithm to the same problem is not immediately apparent. In order to obtain graph-like search behaviour the oracle O_{f_d} employed needs to contain information that is specific to the graph. One possible strategy resides in evaluating if a node alongside a sequence of edges leads to a goal state, as illustrated in Expression 1, where the query register is decomposed into two components, namely $|s\rangle$ containing the initial node and a sequence of edge transitions where each pair $(e_k, e_{k+1}) \in E$. The associated function f_d of the oracle employed is depicted in Expression 1, where S_g represents a set of goal nodes and d the depth of the shallowest solution.

$$|q\rangle|a\rangle = |s\rangle|(e_0, e_1), (e_1, e_2), \cdots, (e_{d-2}, e_{d-1})\rangle|a\rangle \tag{1}$$

$$f_d(s, (e_0, e_1), \cdots, (e_{d-2}, e_{d-1})) = \begin{cases} 1, & \text{if } ((e_0, e_1), \cdots, (e_{d-2}, e_{d-1})) \in S_g \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Tree search problems represent acyclic connected graphs where each vertex has a set of children, whose cardinality is represented through variable b, and at most one parent node. With tree search each additional level of depth adds exponential b^d nodes to the search. When the depth of the shallowest solution is unknown some form of iteration needs to be applied. Accordingly, we can apply Grover's algorithm alongside Expression 2 in order to evaluate superpositions spanning all possible edge combinations up to depth-level d. If a measurement yields a solution, then the algorithm terminates, otherwise the depth limit is iteratively increased and Grover's algorithm is applied with a new O_{f_d}. This procedure which can be performed indefinitely. The overall complexity will be $\sum_{k=0}^{d} \sqrt{b^d} = O(\sqrt{b^d})$ remaining essentially unchanged from that of the original quantum tree search algorithm discussed in [1].

References

1. L. Tarrataca and A. Wichert, "Tree search and quantum computation," *Quantum Information Processing*, vol. 10, no. 4, pp. 475–500, 2011. 10.1007/s11128-010-0212-z.

The Morals of the St. Petersburg Game:
Why Statistical Distributions Matter in Prospect Appraisal

Mariam Thalos and Oliver Richardson

University of Utah, Salt Lake City, United States of America
m.thalos@utah.edu, orichardson95@gmail.com

Abstract

In spite of its infinite expectation value, the St. Petersburg game is a gamble without supply in the real world, but also one without demand at apparently very reasonable asking prices. We offer a rationalizing explanation, in terms of St. Petersburg's fractal probability distribution function (quite heavy-tailed), for why the St. Petersburg bargain is unattractive on both sides (to both house and player) in the mid-range of prices (finite but upwards of about \$4). Our analysis—featuring (1) the already-established fact that the average of finite sequences of the St. Petersburg game grows with sequence length, but is unbounded and (2) our own simulation data showing that the debt-to-entry fee ratio rises exponentially—shows why this is just as it should be: why both house and player are quite rational in abstaining from the St. Petersburg game. The house is unavoidably exposed to very long sequences (with very high averages, and so very costly to them), while contrariwise even the well-heeled player is not sufficiently capitalized to be able to capture the potential gains from long sequences of play (short sequences, meanwhile, enjoy low means and so are not worth paying more than \$4 to play, even if a merchant were to offer them at such low prices). Both sides are consequently rational in abstaining from entry into the St. Petersburg market in the mid-range of asking prices. We utilize the concept of *capitalization vis-à-vis a gamble* to make this case.

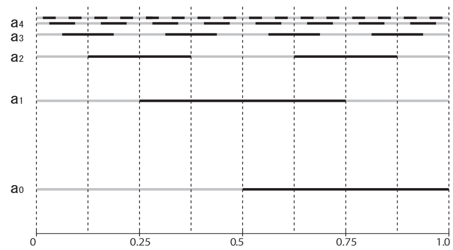
Our treatment demonstrates that fair asking prices, which are fundamentally solutions typically reached in negotiation (real or virtual) between differentially affected parties, with different strategic orientations and different "business models" vis-à-vis the relevant transaction, are incompletely analyzed in traditional appraisals of risky prospects, and no less so in the fictional case of the St. Petersburg gamble. Traditional analysis is especially incomplete in cases characterized by probability distribution functions with divergent, unknown or undefined means—cases with quite substantial representation in the marketplace and high-profile public policy issues.

Multi-Head Extension of the Turing Machine for Computation over Continuous Spaces

Hideki Tsuiki (tsuiki@i.h.kyoto-u.ac.jp)

Kyoto University, Yoshida-Nihonmatsu, 606-8501 Kyoto, Japan

Keywords: Real muber computation; Dimension; Multi-head machine



Let $\mathbb{T} = \{0, 1, \perp\}$ where $\perp$ is the bottom character which means undefinedness. We call an infinite $\mathbb{T}$-sequence with at most n copies of $\perp$ an $n\perp$-sequence and denote by $\mathbb{T}^\omega_n$ the set of $n\perp$-sequences. We have the Scott topology on $\mathbb{T}^\omega$ and its subspace topology on $\mathbb{T}^\omega_n$.

In [Tsuiki 2002, TCS] and [Gianantonio 1999, TCS], they independently introduced a topological embedding of the unit interval $\mathbb{I}$ into $\mathbb{T}^\omega_1$. The above figure depicts this embedding φ_G which we call the Gray-embedding, where, for $a_n = \varphi_G(x)(n)$ $(n = 0, 1, \ldots)$, gray and black lines show that a_n is 0 and 1 on the lines, respectively, and a_n is $\perp$ at their endpoints. Therefore, $\varphi_G(1/2) = \perp 10^\omega$ for example and it gives a unique $\mathbb{T}^\omega_1$-representation to each number. Through this embedding, one can consider that a machine which can input/output a $\mathbb{T}^\omega_1$-sequence is operating directly on real numbers.

However, a machine gets stuck if it tries to read in the value of the bottom cell. In order that a machine can input the sequence after a bottom, the author introduced a machine called an IM2-machine which has two heads on each input/output tape. If the cell-value at one of the input head is not bottom, then it inputs the value and the heads move to the first two unread cells of the tape. If both of the cells under the heads have values, it has two possible behaviors and therefore it is indeterministic. Here, I use the word "indeterministic" instead of "nondeterministic" because our machine should have a correct output not depending on the way it reads the input. In this way, if there is a bottom on the input tape, one head waits for the value of the cell eternally and the other one reads in the rest of the sequence. An IM2-machine outputs an $1\perp-$sequence in a similar way starting with the tape state $\perp^\omega$. It is proved that the computability notion induced on the reals by an IM2 machine with Gray-embedding is equal to the standard one. Note that we have a unique representation to each number, where the representation used by Turing in his "Correction" and more generally admissible representations of real numbers are redundant.

After this, the author studied generalizations of this result to other spaces. In particular, it is proved in [Tsuiki 2004, MSCS] that a separable metrizable space is n-dimensional if and only if it can be embed in $\mathbb{T}^\omega_n$. It means that the dimension of a space, which is a purely mathematical notion, is characterized by the number of extra heads a machine needs to have to input from that space.

Dynamics of Agent Interaction: In search of strategies for optimal information sharing

Ashish Umre and Ian Wakeman

University of Sussex, Brighton, UK
A.Umre@sussex.ac.uk

Keywords: Social Foraging, Game Theory, Collective Behaviour, Social Cognition, Information Sharing, Social Insect Behaviour, Complex Interactions

Social learning is an effective way to reduce uncertainty about the environment, helping individuals to adopt adaptive behaviour cheaply. Although this is evident for learning about temporally stable targets, such as acquisition of an avoidance of toxic foods, the utility of social learning in a temporally unstable environment is less clear, since knowledge acquired by social learning may be outdated. An individual can either depend entirely on its own foraging information (individual forager) or that provided by the environment or shared by other agents. We are interested in scenarios where individual foraging might be a useful and effective strategy and how the topology and distribution of resources in the network/environment might affect this. We investigate the adaptive value of social learning in a dynamic environment both theoretically and empirically.

Some group-living species have evolved effective social mechanisms for reducing uncertainties in their environments. Examples include is a system of reciprocal exchange and sharing of resources such as food. These social mechanisms secure a stable supply of resources by collectively buffering uncertainties associated with their acquisition. In the biological world, evolution has created a large number of distributed systems which balance utility and resource usage. Can we better understand the incentive structures of distributed applications on the internet through examination of biologically inspired algorithms?

Foraging can be modelled as an optimisation process where an animal seeks to maximise the energy (information) obtained per unit time spent foraging. We have overviewed research in foraging theory, relevant to both individual and collective foraging, and significance of these areas to optimisation.

The work [1]has advanced through examination of the behavioural, evolutionary and game-theoretic underpinnings of well-known biological social systems, and used to understand distributed applications through the creation of a simulation environment, in which the similarities between existing and proposed adaptive information dissemination protocols and the foraging behaviour of ants, bees and similar creatures is modelled to investigate the dynamics of social interaction in the context of resource discovery and information dissemination.

[1] Umre, A., Wakeman, I., *Economics of Cooperation: Social Foraging in Distributed Systems*, Journal of Research on Computing Science: Advances in Artificial Intelligence Theory. Alexander Gelbukh & Raul Monroy (Eds.) **16**, (2005).

Emergence of Spatial Patterns in a Model of Wetland Ecosystem: Application to Keoladeo National Park (KNP), India

Ranjit Kumar Upadhyay* and Sharada Nandan Raw

Department of Applied Mathematics, Indian School of Mines,Dhanbad - 826 004, Jharkhand, India
ranjit_ism@yahoo.com (RKU); shardaraw@gmail.com (SNR)

Abstract. An attempt has been made to understand the spatiotemporal dynamics of good biomass (fishes and floating vegetation like Nymphoides indicum, Nymphaea nouchali etc.), bad biomass (Paspalum distichum and its family) and bird population in the biotic system of the wetland part of Keoladeo National Park (KNP), India. Spatiotemporal distributions of species biomass are simulated using the diffusivity assumptions realistic for natural wetland systems. We observed that growth rate and carrying capacity of water availability for the bad biomass, higher value of the carrying capacity and half saturation constant of the good biomass are responsible for the good health of the wetland ecosystem. The study also demonstrates that spatial movements of good biomass acquires stable stationary patterns in the presence of bad biomass which performs swinging motion and selects a steady state spatial pattern, thereby ensuring the persistence as well as extinction of multiple species in space and time. The patterns observed seem to have a similar structure that is uniformity in space, with plateau for good biomass, concentration onto two distinct humps for bad biomass, and extinction for bird population. This will usually be the case in real world scenarios, unless the parameter controlling the death rate of bird population is sufficiently decreased.The mechanisms evolved for the space time survival of these species are thus regulated by interspecific spatial interaction.

The present study suggests that the bad biomass species pull towards dynamic stability and the spatial movement leads towards spatial instability, resulting in the emergence of good biomass and bird population species in inhomogeneous biomass distributions over space and time. Thus, the species of the bad biomass present within the biomass community itself can be viewed as a potential self-regulating candidate, which, combined with physical movement of the biomass species and the structure of the biomass distribution, boosts the emergence of the species in the wetland ecosystem. The overall results may potentially explain the sustainability of biodiversity and the spatiotemporal emergence of good biomass and bird population under the influence of bad biomass combined with their physical movement in the wetland ecosystem.

Keywords: Flood plain wetland; Good biomass; Bad biomass; Spatial patterns.

Mathematical model of chaotic oscillations and oscillatory entrainment in glycolysis originated from periodic substrate supply

Andrey Yu. Verisokin[1], Darya V. Verveyko[2], Eugene B. Postnikov[3]

Theoretical Physics Department, Kursk State University,
Radishcheva 33, 305000 Kursk, Russia
[1]ffalconn@mail.ru, [2]allegroform@mail.ru, [3]postnicov@gmail.com

A special feature of glycolysis is its oscillatory character, which is close connected with an efficiency of the energy metabolism of almost any living organisms [1]. The parameter of a glycolytic system, which can be practically controlled, is the substrate influx into a sphere of reaction that determines a continuing interest to study its details, see e.g. [2].

In our work, we focus on the study of sequential variation of the periodic influx within the classical model of glycolysis: two-dimensional Selkov system. Extensive numerical experiments demonstrate an entrainment of solutions for this qubic non-linear system by the influx and subdivision of parameter plane via Arnold tongues. We determine detailed dynamical regimes exploring the oscillation type inside and between Arnold tongues via Lyapunov characteristic exponents.

This analysis shows that the influx periodicity leads to rich set of oscillatory dynamics: i) limit cycles in the domains of entrainment (Arnold tongues); ii) chaotic regime (strange attractors) between the tongues; iii) stable two-dimensional tori on the borders of tongues.

Thus, the obtained results explain experimental findings [3] detecting oscillations having periods multiple-times the input period as well as chaotic oscillations. Finally, it should be pointed out that the recent research [4] reveals the possibility to model new experimentally found spatio-temporal patterns by Selkov model with non-uniform influx. Thus our results on temporal oscillations have a perspective to be generalized for study of mechanisms of more complex structures studying by modern biophysics.

References

1. Chandra, F.A., Buzi, G., Doyle, J.C.: Glycolytic Oscillations and Limits on Robust Efficiency. Science 333, 187–192 (2011)
2. Torre, J.A., Lemos, M.C., Cordoba, A.: Control of Cellular Glycolysis by Perturbations in the Glucose Influx. In: F. Masulli, R. Tagliaferri, G.M. Verkhivker (eds.): CIBB 2008, LNBI vol. 5488, pp. 132-143. Springer, Berlin–Heidelberg (2009).
3. Markus, M., Kuschmitz, D., Hess, B.: Chaotic dynamics in yeast glycolysis under periodic substrate input flux. FEBS Lett. 172, 235–238 (1984)
4. Lavrova, A., Schimansky-Geier, L., Postnikov, E.B.: Phase reversal in the Selkov model with inhomogeneous influx. Phys. Rev. E 79, 057102 (2009)

Computer simulation of three-dimensional reaction-diffusion models. Case study: glycolytic reaction in open spatial reactor

Darya V. Verveyko[1], Andrey Yu. Verisokin[2]

Theoretical Physics Department, Kursk State University,
Radishcheva 33, 305000 Kursk, Russia
[1]allegroform@mail.ru, [2]ffalconn@mail.ru

Study of behaviour dynamics for large number of processes in physical, chemical and biological systems is closely connected with systems of nonlinear differential equations containing diffusion terms. Such reaction-diffusion equations are used for the description of dynamical spread processes in population dynamics and chemical front propagation as well as for pattern formation (see e.g. [1]).

In the present work we consider the computational analysis of processes connected with glycolytic reaction [2] taking place in an open chemical reactor. The main point of the simulation is taking into account spatial three-dimensionality of the experimental system. For this reason we consider reaction-diffusion system with unidirectional reaction in the bulk supplied by feedback terms stated as boundary conditions on the lower boundary of the reactor.

The numerical solution of proposed model stated mathematically [3] confirms the existence of the experimentally observed glycolytic travelling waves [2]. The analysis of the curvature of the reagents distribution curves proves the kinematic character of the observed waves. But their origin relates to the diffusion of reagents in a vertical reactor cross-section. This phenomenon distinguishes glycolytic travelling waves in an open spatial reactor from travelling reaction-diffusion waves occurring in Fisher - Kolmogorov model. Study of the solutions for the concerned reaction-diffusion model in the case of different diffusion coefficients allows to find the regimes of Turing structures.

Thus, we can conclude that considerable model successfully describes the experimentally observed nonlinear regimes and allows to explain the reasons of such a behaviour.

References

1. Murray, J. D.: Mathematical Biology: II. Spatial Models and Biomedical Applications: Springer (2003).
2. Bagyan, S., Mair, T., Sukhorski, Y., Hauser, M. J. B., Straube, R.: Spatial Desynchronization of Glycolytic Waves as Revealed by Karhunen Loève Analysis. J. Phys. Chem. B 112, 14334–14341 (2008)
3. Postnikov, E. B., Verisokin, A. Yu., Verveyko, D. V., Lavrova, A. I.: Self-sustained biochemical oscillations and waves with a feedback determined only by boundary conditions. Phys. Rev. E 81, 052901 (2010)

The Linguistic Turing Machine

Jeffrey Watumull*
*Massachusetts Institute of Technology (watumull@mit.edu)

Cognition is demonstrably computational and computation is mathematical by definition: procedures are run to determine the symbolic outputs (values) of functions given symbolic inputs (arguments); in the domain of the brain, the running of procedures is referred to informally as "thinking." In a successful model of this process, functions would be "*completely* determined" (Turing) by rules and representations so "perfectly explicit" (Chomsky) as to be automated, for "*if you can't program it, you haven't understood it*" (Detusch). The successful model would be *descriptively* and *explanatorily adequate*: it would completely describe *what* the system of rules and representations is and completely explain *how* internal and external factors determine the genotype-to-phenotype expression of the cognitive system. The model might even go *beyond* explanatory adequacy to answer the biggest question of all: *Why* does the system assume this one form out of the infinity of conceivable forms? Decomposing these big questions into smaller solvable problems, I propose to model mathematically important aspects of intelligent thought in the domain of language. In particular, I intend to adopt a rigorous and empirical definition of "language" as an *I-language*: a cognitive computational system—an *intensional* function—*internal* to an *individual* of the species *Homo sapiens sapiens.* I assume that I-language, as a computational system, can be defined as a type of Turing machine; modeling the former as a type of the latter would precisify the nature of linguistic computation so as ultimately to be unified with a yet to be formulated model of neurobiological computation.

A linguistic Turing machine LTM is the 5-tuple $(Q, \Gamma, \delta, \#_S, \#_H)$

Q : set of states/instructions (i.e., linguistic principles and parameters)

Γ : set of symbols for syntactic objects (e.g., lexical items, phrases, etc.)

δ : $Q \times \Gamma \to Q \times \Gamma$ (i.e., transition function from state/symbol to state/symbol by a search/merge procedure)

$\#_S$ $(\in \Gamma)$: start (boundary) symbol

$\#_H$ $(\in \Gamma)$: halt (boundary) symbol

LTM recursively generates syntactic structures mappable via formal semantics and rule-based morphology-phonology to interfaces with conceptual-intentional and sensory-motor systems, respectively. LTM is thus a system of *discrete infinity* analogous to the natural numbers: a finite system that in principle can "strongly generate" an infinite set of hierarchically structured expressions by recursively combining discrete elements. Descriptive adequacy: I propose to describe some of the rules and representations of language by conducting typological research on linguistic universals; if LTM is the correct model of the mathematical system implemented in the brain, then its rules and representations are predicted to be universal (species-typical). Explanatory adequacy: I propose to explain (prove) how a PAC algorithm (with an oracle) defined in terms of strong generative capacity enables the rules and representations of LTM to be acquired. Beyond Explanatory Adequacy: I propose to investigate whether the system of rules and representations of LTM can be derived from mathematics of computability and complexity. (NB: I-language could be uncomputable, with profound implications.) For this I need to revamp the Chomsky Hierarchy from specifying strings (weak generation) to specifying structures (strong generation). To the extent that this research program succeeds, unification of computational cognitive science and computational neuroscience becomes possible. It could thus render commensurable the computational ontologies of linguistics and neuroscience. I thus need to define the linguistic primitives formally so that "linking hypotheses" (not mere correlations) to neurobiological primitives can be formed. If I-language were precisified as a form of TM, our imagination for how language functions in abstract computation and how that relates to concrete computation in the brain would be profoundly expanded—so as, perhaps, to compass the truth. "[Imagination] is that which penetrates into the unseen worlds around us, the worlds of Science [. . .]. Those who have learned to walk on the threshold of the unknown worlds [. . .] may then with the fair white wings of Imagination hope to soar further into the unexplored amidst which we live" (Lady Lovelace).

Alan Turing and Maurice Wilkes

Peter Wegner[1]

Brown University

Alan Turing was born in June 1912, just one year before Maurice Wilkes. His Cambridge undergraduate program from 1931-34 coincided with that of Wilkes, and they knew each other personally. Maurice Wilkes, who supervised my graduate program in "Numerical Analysis and Automatic Computing" during 1953-54, informed me of Turing's death which occurred in June 1954 while I was taking the final examination for my post-graduate diploma. I continued to interact with Maurice throughout my career, and invited him to lecture at Brown when he was working in Boston after completing his stint as head of the Cambridge computer science department.

Wilkes and I both admired Turing's technical contributions but questioned the characterization of the Turing machine as a comprehensive model of computing and problem-solving. Maurice asserted that Turing, although a great thinker, was unable to create a Turing machine because his ability to direct collaborators was weaker than his ability to formulate ideas. He praised my questioning of the view that the Church-Turing thesis provided a complete basis for computation and accepted my assertion that Turing machines are a powerful but incomplete model of computing, and that modern computing machines are more powerful than Turing machines.

When I visited Sir Maurice at his Cambridge home in 2008 (two years before his death), we discussed both the power and the limitations of Turing's model of computation, as well as the view that a complete model of computation (just as a complete model of mathematics, physics, or science as a whole) does not yet exist. We can expand our understanding of computation and other scientific methodologies, but are as yet unable to develop complete descriptions of underlying disciplines.

Dynamic Semantics of τN -Theories

R. L. Wojtowicz[1], S. Bringsjord[2], and J. Hummel[3]

[1] Baker Mountain Research Corp.; Yellow Spring, WV; ralphw@bakermountain.org
Shepherd University; CME Department; Shepherdstown, WV; rwojtowi@shpherd.edu
[2] Rensselaer Polytechnic Institute; Department of Computer Science and Department of Cognitive Science; Troy, NY; Selmer.Bringsjord@gmail.com
[3] University of Illinois at Urbana-Champaign; Department of Psychology; Champaign, IL; jehummel@cyrus.psych.illinois.edu

Abstract. In this paper we review τN-theories as environments that are suitable for expressing concepts that arise in the Incompleteness Theorems. We describe discrete- and continuous-time dynamic systems as categories in which to build models of such theories and we explore structure-preserving mappings between them. This work is motivated by a program aimed at developing new logic-based artificial intelligence capabilities that blend analogical reasoning with machine deduction and new logic-based machine learning techniques that symbolically manipulate complex structures.

Keywords: category, dynamic system, higher-order logic, incompleteness

Stochastic reaction and diffusion on growing domains: understanding the breakdown of robust pattern formation

Thomas E. Woolley[1,2], Ruth E. Baker[1], Eamonn A. Gaffney[1], and Philip K. Maini[1,3]

[1] Centre for Mathematical Biology, Mathematical Institute, University of Oxford, 24-29 St Giles', Oxford, OX1 3LB, UK.
[2] Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, University of Oxford, 24-29 St Giles', Oxford, OX1 3LB, UK.
[3] Oxford Centre for Integrative Systems Biology, Department of Biochemistry, University of Oxford, South Parks Road OX1 3QU, UK.

Abstract. All biological patterns, from population densities to animal coat markings, can be thought of as heterogeneous spatial-temporal distributions of reactive agents. Many mathematical models have been proposed to account for the emergence of this complexity, but, in general, they have consisted of deterministic systems of differential equations, which do not consider the stochastic nature of population interactions. One particular, pertinent, criticism of these deterministic systems, is that the exhibited patterns can often be highly sensitive to changes in initial conditions, domain geometry, parameter values, etc. Due to this sensitivity, we seek to understand the effects of stochasticity and growth on biological patterning paradigm models. In this paper, we extend spatial Fourier analysis and growing domain mapping techniques to encompass stochastic Turing patterning systems. Through this we find that the stochastic systems are able to realise much richer dynamics than their deterministic counterparts, in that patterns are able to exist outside the standard Turing parameter range. Further, it is seen that the inherent stochasticity in the reactions appears to be more important than the noise generated by the growth. Finally, although growth is able to generate robust pattern sequences in the deterministic case, we see that stochastic effects destroy this mechanism of pattern doubling. However, through Fourier analysis we are able to suggest a reason behind this lack of robustness and identify possible mechanisms by which to reclaim it.

woolley@maths.ox.ac.uk

Degrees of Relations on Ordinals

Matthew Wright

University of Chicago, Chicago, IL, USA

Downey, Khoussainov, Miller, and Yu showed that the degree spectrum of any computable unary relation on $(\omega, <)$ either contains only the computable degree, or contains $\emptyset'$—that is, if the relation is not intrinsically computable, then it contains $\emptyset'$. However, their proof does not extend to n-ary relations, nor does it extend to relations on structures other than ω. We will show that their results can be extended in both of these directions, first by showing that the same result holds for n-ary relations on ω, and then showing that a more general version holds of unary relations on arbitrary computable ordinals.

The first extension of the original result is as follows:

Theorem 1. *Let R be any computable n-ary relation on $(\omega, <)$ that is not intrinsically computable. Then there is a computable copy $\mathcal{M}$ of $(\omega, <)$ such that $R^{\mathcal{M}}$ computes $\emptyset'$.*

While the construction used in the proof follows the same basic structure as the construction used by Downey, Khoussainov, Miller, and Yu, several combinatorial lemmas are required in this setting which were not required in the unary case—in particular, we use both Ramsey's theorem and the theory of well quasi orderings in setting up the construction.

The second extension is the following:

Theorem 2. *Let α be a computable ordinal, and R a computable unary relation on a copy of $(\alpha, <)$ satisfying certain computability conditions. Let ν be any computable ordinal, and assume that for all $\mu < \nu$ there is a copy $\mathcal{B}$ of α such that $R^{\mathcal{B}} \not\leq_T \emptyset^{(\mu)}$. Then there is a copy $\mathcal{A}$ of α such that $R^{\mathcal{A}} \equiv_T \emptyset^{(\nu)}$.*

That is, the degree spectrum of a computable relation on a computable ordinal always has a maximum degree, and that degree is always $\emptyset^{(\alpha)}$ for some ordinal α. The proof of this theorem relies on results about back-and-forth relations in order to build copies of ordinals, using results of Ash and Knight.

Part restrictions: adding new expressiveness in Description Logics

Mitko Yanchev ⋆ e-mail: yanchev@fmi.uni-sofia.bg

Sofia University "St. Kliment Ohridski"
5 James Bourchier blvd., 1164 Sofia, Bulgaria

Abstract. Description Logics (DLs) are logical formalism widely used in knowledge-based systems. DLs both explicitly represent knowledge in form of taxonomy, and infer new knowledge out of the presented structure by means of a specialized inference engine. The representation language, called *concept language*, comprises expressions with only unary and binary predicates, called *concepts* and *roles*. Concept languages differ mainly in the constructors adopted for building complex concepts and roles, and they are compared with respect to their expressiveness, as well as with respect to the complexity of reasoning in them. The language $\mathcal{AL}$ is usually considered as a "core" one, having the basic set of constructors: $\neg A$ (atomic negation), $C \sqcap D$ (intersection), $\forall R.C$ (universal role quantification), and $\exists R.\top$ (restricted existential role quantification).
We introduce new concept constructors, called *part restrictions* (denoted by $\mathcal{P}$), which are capable to distinguish a part of a set of successors. These are $MrR.C$ and (the dual) $WrR.C$, where r is an arbitrary rational number in (0,1), R is a role, and C is a concept. The intended meaning of $MrR.C$ is "More than r-part of all R-successors of the current object has the property C". Part restrictions essentially enrich the expressive capabilities of Description Logics.
We explore the complexity of the main reasoning task in Description Logics—checking the subsumption between concepts. We prove NP-completeness of subsumption in the basic Description Logic with part restrictions $\mathcal{ALP}$, and in semantically equivalent to it $\mathcal{ALEP}$, adopting also full existential role quantification ($\mathcal{E}$). The hardness part of the proof uses simulation of the later constructor via M-constructor ($\mathcal{E}$-$\mathcal{P}$-simulation). Correctness of the simulation is shown in the language $\mathcal{ALCP}$, adopting in addition to $\mathcal{ALP}$ full existential role quantification and union of concepts ($\mathcal{U}$) (or full negation—$\mathcal{C}$). For the "in" part of the proof an appropriate completion calculus based on tableau technique is used.
We use $\mathcal{E}$-$\mathcal{P}$-simulation also to prove PSPACE-hardness of subsumption in $\mathcal{ALUP}$. Then, we consider Description Logic $\mathcal{ALCQP}$ ($\mathcal{Q}$ is for qualified number restrictions, more expressive variant of number restrictions, $\mathcal{N}$). $\mathcal{ALCQP}$ is a syntactical variant of modal logic GGML, for which the satisfiability problem is PSPACE-complete. This yields the PSPACE-completeness of subsumption in $\mathcal{ALCQP}$ and in its sublanguages $\mathcal{ALUP}$, $\mathcal{ALCP}$, and $\mathcal{ALCNP}$.
Key Words: Description Logics, Part restrictions, Complexity.

⋆ The work on this presentation was granted by Science Fund of SU, contract no. 176/2012.

MIND as IRIS versus Mind as Machine: An Alternative Analogy for the Mind

Aziz F. Zambak

Department of Philosophy, Yeditepe University, Istanbulx
aziz.zambak@gmail.com

Abstract. Mind-as-machine is the essential metaphor of cognitive science employing computer-based terminology in the explanation of mental phenomena. The mind-as-machine metaphor seems very practical and attractive to psychologists and AI researchers since it provides a scientific ground for the intelligibility of mind. Moreover, the mind-as-machine metaphor indicates a technological possibility for a non-biological form of the human mind. The notion of computer has dominated the science of cognition and this domination forces us to describe a mental phenomenon only in a representational system (e.g., symbol-processing system) that restricts seeing the conditions of/for human mind. In addition to that, describing a mental phenomenon in purely mechanistic terms prevents seeing the agentive character of the human mind. The mind-as-machine metaphor has two basic forms: the representational form and the Cartesian form. In the representational form, the mind is considered as a mechanistic device that can be represented in an algorithm. In AI, this algorithm is described in a very restricted range of specific problems (e.g., expert systems), and for AI it is possible to enlarge these algorithms for every cognitive skill of the human mind. In the Cartesian form, AIs central paradigm, cognition as computation, refers to two independent entities such as brain as hardware and mind as software (program). Therefore, modern AI considers that internal brain processes can be functionalized and embodied in a separated and self-governing computational algorithm. In other words, AI conceives software as an independent and sufficient entity in order to realize the particular functions of the brain (i.e., hardware).
We defend the idea that the methodology of cognitive science (e.g., mind-as-machine metaphor) does not imply any principle to the model of mind in AI since AI requires a distinctive and original modeling strategy peculiar to machine intelligence. In other words, the theoretical and methodological issues in cognitive science are not the subject matter of AI in order to develop a model of mind. Therefore, AI should situate its methodological position out of the discussion of whether the mental phenomenon and cognitive skills can be explained in a computer-based and mechanistic terminology. What AI should do is develop a mental analogy associated with the conditions of/for the human mind. We propose mind-as-IRIS [Idealized-Reflective-Informational-System] as a proper analogy for the mental models in AI, rather than the mind-as-machine analogy.

The modality "Future" for Probably true propositions

Paola Zizzi
Department of Psychology, University of Pavia,
Piazza Botta, 6, 27100 Pavia, Italy
paola.zizzi@unipv.it

Abstract

A quantum logic should be the logic underlying a quantum system, whose state is described in terms of some pairs of conjugate variables, which satisfy some uncertainty relations. It follows that a logic is quantum if and only if the propositions themselves of its language obey some (logical) uncertainty principle. In this regard we will consider the atomic propositions of a quantum logical language, or quantum object-language (QOL), that can be asserted, in the quantum metalanguage (QML) [1], with an assertion degree α, which is a complex number (within the formalism of sequent calculus, such assertions are denoted by $\vdash^{\alpha} A$). This fact requires that the atomic propositions in the QOL are endowed with a fuzzy modality "Probably" [2] and have fuzzy (partial) truth-values [3]. The latter, moreover, sum up to one. In general, such a set of probabilistic propositions is a subset of a bigger set, including also non-probabilistic ones. We found an uncertainty relation between the (partial) truth-values $v_i = |\alpha_i|^2$ *(i =1,2…n)*, where *n* is the number of probabilistic propositions, and the total number *N* of propositions [4]. Also, we defined as "quantum coherent propositions" those propositions of the QOL, all having the same partial truth value $|\alpha|^2$, which minimize the logical uncertainty relation mentioned above. This definition follows from the fact that the corresponding assertions in the QML having assertion degree α can be physically interpreted as the coherent states $|\alpha\rangle$ of Quantum Optics (Glauber states) [5] , which are eigenstates of the annihilation operator a, with eigenvalue α, that is: $a|\alpha\rangle = \alpha|\alpha\rangle$. We recall that in Quantum Theory the displacement operator $D(\alpha) = \exp(\alpha a^{\dagger} - \alpha^* a)$ is the unitary operator which displaces the vacuum state $|0\rangle$ to the coherent state $|\alpha\rangle$, that is: $D(\alpha)|0\rangle = |\alpha\rangle$. We look for the logical analogous of the operator $D(\alpha)$ in order to build up an assertion with assertion degree α starting from an assertion with assertion degree $\alpha = 0$. The latter is the assertion of a logical proposition *A* with truth value $v = 0$, that is, a false proposition: $\vdash^{\alpha=0} A \equiv A^{\perp} \vdash$. Then, given a (linear) flux of time $\tau = (T,<)$ as in Temporal Logic [6], we consider the valuation π on τ, namely, $\pi : (T \to (\Psi \to [0,1]))$, where Ψ denotes the set of propositions Pp_i ("Probably" p_i). Therefore, if at time t a proposition Pp_i is false, we have: $\pi(t)(Pp_i) = 0$, $\pi(s)(Pp_i) = v_i$, where $v_i \in [0,1]$ with the constraint $\sum_i^n v_i = 1$, and $\{s \in T\, t < s\}$ is the future of *t*. In this way, it is possible to build up, along the flow of time, propositions with partial truth values starting from a false proposition. It is to be remarked that in Quantum Theory, the so-called bosonic transformation [7] is a displacement of the annihilation operator: $D^{\dagger}(\alpha) a D(\alpha) = a + \alpha \equiv a(\alpha)$. Then, the original vacuum state $|0\rangle$ is not anymore annihilated by the displaced annihilation operator: $a(\alpha)|0\rangle = \alpha|0\rangle$. The old vacuum can be then viewed as a coherent vacuum state, in the sense that it is an eigenstate of the displaced annihilation operator with eigenvalue α. This means that the original vacuum does not logically correspond anymore to the assertion of a false proposition. One can however define a new vacuum state: $|0(\alpha)\rangle = D|0\rangle$, which is annihilated by $a(\alpha)$. Notice that the original vacuum state $|0\rangle$ can be obtained from the coherent vacuum $|0(\alpha)\rangle$ by the action of the inverse $D^{-1}(\alpha) = D(-\alpha)$ of the displacement operator. In logical terms, $D^{-1}(\alpha)$ corresponds to the inverse of the valuation, namely π^{-1}. This entails that we can recover a false proposition from a coherent proposition with probabilistic fuzzy truth value $|\alpha|^2$ by means of the inverse of the map π, that is: $\pi^{-1}(t)(Pp_i) = v_i$, $\pi^{-1}(s)(Pp_i) = 0$, which is equivalent to interchange the modality "Future" with the modality "Past".

References

[1] P. Zizzi, "From Quantum Metalanguage to the Logic of Qubits". PhD Thesis, arXiv:1003.5976 (2010).
[2] H´ajek P., Godo L., Esteva F., Probability and Fuzzy Logic. In Proc. Of Uncertainty in Artificial Intelligence UAI'95, (Besnard and Hanks, Eds.) Morgan Kaufmann. San Francisco, 237–244 (1995).
[3] L. A. Zadeh et al. " Fuzzy Sets, Fuzzy Logic, Fuzzy Systems", World Scientific Press (1996).
[4] P. Zizzi, "The uncertainty relation for quantum propositions", arXiv: 1112.2923, submitted to IJTP.
[5] R. J. Glauber, "Coherent and incoherent states of radiation field", *Phys. Rev.* **131,** 2766-2788 (1963).
[6] A. N. Prior, "Time and Modality", Oxford: Clarendon Press (1957).
[7] H. Umezawa, "Advanced Field Theory: Micro, Macro and Thermal Physics". American Institute of Physics (1993).