DEPARTMENT OF
COMPUTER
SCIENCE

Proceedings of the University of Oxford Department of Computer Science Student Conference 2012

Organizing Committee: Miriam Backens, Krzysztof Bar, Ventsi Chonev, Adrian Duncan, Lu Feng, Thomas Gibson-Robinson

16 November 2012

Foreword

The 2012 Department of Computer Science Student Conference was held on the 16th November in the department. This year we had a record number of submissions with 26 abstracts and 8 posters submitted. Most pleasingly, the 15 abstracts that were accepted represented research from across each one of the department's research themes (aside from Algorithms, which had only just been formed).

The conference was not only an interesting and enjoyable event, but also offered an opportunity to practice useful skills for many of those involved. Many of the speakers were giving their first ever academic talk (and doing so most successfully). Additionally, a number of the reviewers were themselves DPhil students, and no doubt gained valuable reviewing experience. We hope that many of the conference attendees will have found out about research that is being done in other research groups in the department. We believe that one of the most important aspects of the student conference is its ability to bridge the subject divide within the department.

The conference would have not been a success without the help of a significant number of people to whom the organising committee would like to express their thanks to. In particular, we would like to thank Prof. James Worrell, who gave the keynote talk of the student conference (having presented a paper himself at a Departmental Student Conference in the late 1990s) and judged several of the prizes. We would also like to thank Julie Sheppard and Wendy Adams who put a huge amount of time and effort into organising the conference. We are very grateful to our selection committee of DPhil students, post-doctoral researchers and some more senior academics, who had the difficult job of reviewing the submissions. Most importantly, we would like to thank everyone who prepared an abstract or poster and enabled us to put together such a wonderful showcase of departmental research.

We hope that all who attended the conference found it an enjoyable experience and would like to wish next year's organisers good luck.

Thomas Gibson-Robinson

Prizes

Best Abstract: Andrew Paverd

Best Poster: Steven Ramsay and Robin Neatherway

Best Talk: Martin Lester

Organization

Organizing Committee

Miriam Backens, Krzysztof Bar, Ventsi Chonev, Adrian Duncan, Lu Feng, Thomas Gibson-Robinson

Selection Committee

Ioannis Agrafiotis, Miriam Backens, Rafel Bordas, Pierre Bourhis, Luke Cartey, Ventsi Chonev, Conrad Drescher, Adrian Duncan, Thomas Dunton, Sara Dutta, Shamal Faily, Lu Feng, Thomas Gibson-Robinson, Michael Goldsmith, Edward Grefenstette, Daniel Harvey, Duncan Hodges, Ernesto Jimenez-Ruiz, David Kay, Sophie Kershaw, Markus Kroetzsch, Katie Leonard, Martin Lester, Despoina Magka, Alex Merry, Nick Moffat, Robin Neatherway, Giorgio Orsi, José Pedro Magalhães, Rui Soares Barbosa, Tamas Szekely, Jamie Vicary, Jim Whitehead II

Keynote Speaker

James Worrell

Thanks

Special thanks to Julie Sheppard and Wendy Adams for all of their help organizing the conference. Thanks to Sara Dutta and Ernesto Jimenez-Ruiz for chairing sessions.

Conference Program

LTL model checking on Interval Markov Chains

Michael Benedikt, Rastislav Lenhardt, and James Worrell

Department of Computer Science, University of Oxford, United Kingdom

Introduction

Interval Markov chains (IMCs) generalise ordinary Markov chains by having variables that represent undetermined transition probabilities. Particular values within the interval boundaries can be substituted for the variables, yielding an ordinary Markov chain. IMCs arise naturally in the modelling and verification of probabilistic systems. For example, in an open system some transition probabilities may depend on an unknown environment; we may also want to model systems in which transition probabilities are only approximately known or are parameters that can be optimised.

We consider the problem of computing optimal (either maximum or minimum) probabilities for interval Markov chains to satisfy LTL specifications. We describe an expectation-maximisation (EM) algorithm to solve this problem. Our algorithm can be seen as a variant of the classical Baum-Welch algorithm on hidden Markov models, but it differs from the latter in several key respects. We prove convergence of the algorithm and introduce a publicly available on-line tool to perform such analysis of the systems.

Overview of Algorithm

Our algorithm starts with translation of LTL property φ to an unambiguous Büchi automaton. We use unambiguous automata, because we know that they will have at most exponentially many states in $|\varphi|$ comparing to double exponentially many states when translating to deterministic automata. The advantage of this approach has been confirmed by our implementation, where we could observe that we get much smaller unambiguous automata for many LTL properties. (see e.g. [BLW11] for more details how unambiguous automata can help in model-checking). Then we form a cross product M with an interval Markov chain.

The core of our algorithm is a calculation for updating the probabilities of variables in the cross product M. We denote by X a set of variables and by X_i groups of variables leaving the same vertex in the original IMC.

EM(M cross product, n number of iterations)

$\boldsymbol{v}_0(x) \leftarrow$ initial values of variables;
while $i < n$ **do**
  $\boldsymbol{v}_{i+1} = update(\boldsymbol{v}_i, M)$;
  i++;
end while

Given a valuation $\boldsymbol{v}$: $X \to \mathbb{R}$, we show that we can compute a new valuation $\boldsymbol{v}' = \mathit{update}(\boldsymbol{v}, M)$ of the variables in M. For any variable $x \in X_i$, let $E_x = \{e \in E : \ell_E(e) = x\}$ and let $E_i = \{e \in E : \ell_E(e) \in X_i\}$, where function $\ell_E(e)$ assigns edge e either fixed probability or variable. Let $\mathbf{E}[Z_e]$ be the expected number of times the edge e is taken before the trajectory reaches accepting set F. Then we define $\boldsymbol{v}'(x)$, the new value of x, to be

$$\boldsymbol{v}'(x) \stackrel{\text{def}}{=} (1 - \mu_i) \cdot \frac{\sum_{e \in E_x} \mathbf{E}[Z_e]}{\sum_{e \in E_i} \mathbf{E}[Z_e]}.$$

This equals the expected number of times for a trajectory to take an x-labelled edge divided by the expected number of times to take an X_i-labelled edge, where both expectations are conditioned on reaching F (and the result is renormalised by a factor taking into account all outgoing transitions with fixed probability). This works only for the transition interval boundaries $[0, 1]$. Therefore we had to adapt this technique to support other boundaries.

Tool Available Online

A beta version of our tool can be accessed at http://tulip.lenhardt.co.uk. It is equipped with several examples showing how it can be used. For example, to find mixed strategies in some economic games, to evaluate properties specifying competing goals or to synthesise optimal parameters for probabilistic systems. The tool takes as input a labelled interval Markov chain with properties specified either by LTL formulas or directly by unambiguous Büchi automata. It performs a specified number of iterations and outputs approximation to maximum probability with which IMC satisfies the property together with the values within the intervals for which the maximum is achieved.

Our implementation of the translation from LTL to unambiguous Büchi automata is similar to the approach of [GPV$^+$95], which translates LTL to non-deterministic automata. For both, automata and cross product construction, we use several techniques to reduce the state space: collapsing vertices, removing the final states in non-accepting strongly connected components and probabilistic bisimulation. Our expectation maximisation algorithm runs in cubic time.

Acknowledgement We would like to thank the authors of PRISM for their parser, LTL2dstar for their LTL formula simplifier, JAMA for solving systems of linear equations and authors of Graphviz for being able to visualise models, automata and cross products.

References

[BLW11] Michael Benedikt, Rastislav Lenhardt, and James Worrell. Two variable vs. linear temporal logic in model checking and games. In *CONCUR*, pages 497–511, 2011.

[GPV$^+$95] Rob Gerth, Doron Peled, Moshe Y. Vardi, R. Gerth, Den Dolech Eindhoven, D. Peled, M. Y. Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *In Protocol Specification Testing and Verification*, pages 3–18. Chapman & Hall, 1995.

Playing Stochastic Games Precisely

Taolue Chen[1], Vojtěch Forejt[1], Marta Kwiatkowska[1],
Aistis Simaitis[1], Ashutosh Trivedi[2], Michael Ummels[3]

[1] Department of Computer Science, University of Oxford, Oxford, UK
[2] University of Pennsylvania, Philadelphia, USA
[3] Technische Universität Dresden, Germany

We study stochastic two-player games where the goal of one player is to achieve *precisely* a given expected value of the objective function, while the goal of the opponent is the opposite. Potential applications for such games include controller synthesis problems where the optimisation objective is to maximise or minimise a given payoff function while respecting a strict upper or lower bound, respectively. We consider a number of objective functions including reachability, ω-regular, discounted reward, and total reward. We show that precise value games are not determined, and compare the memory requirements for winning strategies. For stopping games we establish necessary and sufficient conditions for the existence of a winning strategy of the controller for a large class of functions, as well as provide the constructions of compact strategies for the studied objectives.

In this paper we take a different stand from the well-established notion of viewing players as optimisers which, even though useful in many applications, is inadequate for the problems requiring precision. Among others, such precision requirements may stem from: a) controller design under strict regulatory or safety conditions, or b) optimal controller design minimising or maximising some payoff function while requiring that a given lower or upper bound is respected. For instance, consider the task of designing a gambling machine to maximise profit to the "house" while ensuring the minimum expected *payback* to the customers established by a law or a regulatory body [6,1]. Given that such a task can be cast as a controller synthesis problem using stochastic games, the objective of the controller is to ensure that the machine achieves the expected payback *exactly* equal to the limit set by the regulatory body—higher paybacks will result in a substantial decrease in profits, while lower paybacks will make the design illegal. There are examples from other domains, e.g., ensuring precise 'coin flipping' in a *security protocol* (e.g., Crowds), keeping the expected voltage constant in *energy grid*, etc.

In order to assist in designing the above-mentioned controllers, we consider the problem of achieving a *precise* payoff value in a stochastic game. More specifically, we study games played over a stochastic game arena between two players, Preciser and Spoiler, where the goal (the winning objective) of the Preciser is to ensure that the expected payoff is *precisely* a given payoff value, while the objective of the Spoiler is the contrary, i.e., to ensure that the expected value is anything but the given value. We say that the Preciser wins from a given state if he has a winning strategy, i.e., if he has a strategy such that, for all strategies

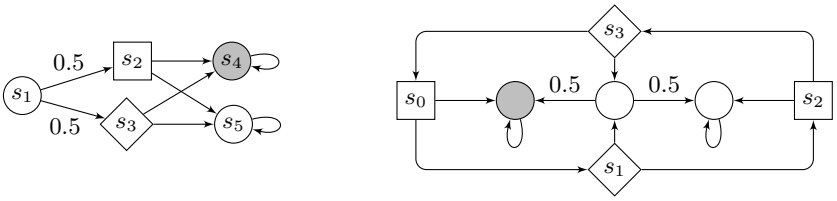


Fig. 1. Two stochastic game arenas where we depict stochastic vertices as circles and vertices of players Preciser and Spoiler as boxes and diamonds, respectively.

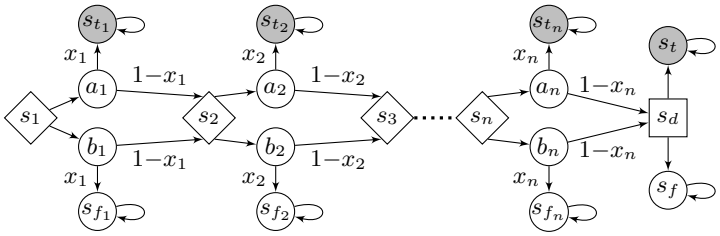


Fig. 2. Exponential deterministic update memory for Preciser

of Spoiler, the expected payoff for the given objective function is *precisely* a given value x. Similarly, the Spoiler wins from a given state if she has a strategy such that, for all strategies of Preciser, the payoff for the given objective function is not equal to x.

Contributions. This is an extended abstract of [3]. The contributions of the paper can be summarised as follows.

- We show that stochastic games with precise value objectives are not determined even for reachability objectives (e.g., a game in Figure 1 (left) with objective $x = 0.5$), and we compare the memory requirements for different types of strategies. We show that at least exponential (in the size of the game) memory is required if deterministic memory update function is chosen (e.g., as is the case in the game from Figure 2 with objective $x = 0.5$), however, if we allow stochastic update strategies linear memory suffices.
- We solve the *controller synthesis* problem for precise value in stopping games for a large class of functions and provide a construction for compact winning strategies. For example, it follows from our results that for reachability objectives in stochastic two-player game, Preciser has a strategy to *exactly* achieve all values between $\inf\sup(s)$ and $\sup\inf(s)$ if and only iff in all states of the game reachable by optimal strategies $\inf\sup(s) \leq \sup\inf(s)$. Where $\inf\sup$ and $\sup\inf$ denote, respectively, the smallest and largest reachability probabilities achievable by Preciser regardless of strategy of Spoiler in state s. The strategy memory is linear in the number of states in the game, i.e., for each state it contains a memory element for $\inf\sup$ and $\sup\inf$. We also show that decision problem of whether there exists a winning strategy for Preciser is in NP $\cap$ CO-NP for games with objectives that admit pure memoryless strategies for both players.

Related work. Stochastic games have been studied in detail in the literature [4,5,2], but we are not aware of any other work studying the precise value problem for any objective function.

References

1. A. N. Cabot and R. C. Hannum. Gaming regulation and mathematics: A marriage of necessity. *John Marshall Law Review*, 35(3):333–358, 2002.
2. K. Chatterjee and T. A. Henzinger. A survey of stochastic ω-regular games. *J. Comput. Syst. Sci.*, 78(2):394–413, 2012.
3. T. Chen, V. Forejt, M. Kwiatkowska, A. Simaitis, A. Trivedi, and M. Ummels. Playing stochastic games precisely. In *Proc. CONCUR'12*, pages 348–363, 2012.
4. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.
5. A. Neyman and S. Sorin, editors. *Stochastic Games and Applications*, volume 570 of *NATO Science Series C*. Kluwer Academic Publishers, 2004.
6. State of New Jersey, 214th legislature, as amended by the General Assembly on 01/10/2011. `http://www.njleg.state.nj.us/2010/Bills/S0500/12_R4.PDF`, November 2010.

Information Flow Analysis for JavaScript
via a dynamically typed language with staged metaprogramming

Martin Lester

Department of Computer Science, Parks Road, Oxford, OX1 3QD, UK
`http://mjolnir.cs.ox.ac.uk/web/slamjs/`

Web applications written in JavaScript are regularly used for dealing with sensitive or personal data. Consequently, reasoning about their security properties has become an important problem, which is made very difficult by the highly dynamic nature of the language, particularly its support for *runtime code generation* through the **eval** construct.

We have developed a *static information flow* analysis for a dynamically typed, functional language with *staged metaprogramming* that has semantics similar to JavaScript. Our analysis works by extending the well-known Control Flow Analysis (CFA) with an abstraction for staged code and information flow constraints. We have formally proved the soundness of our analysis in the mechanised theorem prover Coq and implemented it in OCaml [6].

This is the first information flow analysis for a language with staged metaprogramming and the first formal soundness proof of a CFA-based information flow analysis for a functional language. We argue that our work is applicable to an information flow analysis for full JavaScript, including **eval**, and could easily be transferred to other CFA-based analyses.

Information Flow and Noninterference The study of *information flow* security stems from the observation that programs contain both *direct* and *indirect* information flows [2].

Consider evaluation of the program $(\textsf{if}(h)\{l\}\,\textsf{else}\{0\})$. The result may be l, so there is a *direct* flow from l to the result. However, we might also infer from the result whether h is **true** or **false**. As h affects the result of the program through control flow, but does not directly contribute to the result, there is an *indirect* flow from h to the result.

Early work on information flow focused on augmenting program execution with a monitor to *taint* variables that contained high-security data [3]. But this method is weak at handling indirect flows, which may arise from program branches not executed in all runs.

There is a wide body of research concerning information flow analysis and security in statically typed languages [7], but relatively little for dynamically typed languages.

A popular information flow security property is *noninterference* [4]. Suppose that the inputs and outputs to a program are partitioned into different security levels. For example, some may be high-security (or *high* for short) while others are low-security (or *low*). A program satisfies noninterference if its high inputs do not affect its low outputs.

In the context of a Web application, a high input might be a text input box for a credit card number and a low output might be an unencrypted connection to a webserver. Such an application would satisfy a noninterference analysis if unencrypted transmissions could never reveal anything about the credit card number. Our analysis identifies which program values (inputs) might affect the result (output), so it can be used to verify noninterference.

Staged Metaprogramming JavaScript's **eval** construct takes a string and executes it as if it were a piece of program code. This is difficult to analyse for three reasons. Firstly, an analysis must determine what code strings may be passed to the **eval** and what code they may represent. Secondly, there may be infinitely many possible code strings, so there may be infinitely many subprograms to analyse. Thirdly, code executed by **eval** operates under different scoping rules. Not only must the analysis handle a mixture of *dynamic and static variable scoping*, but code using **eval** does not respect α-equivalence.

Metaprogramming constructs are poorly understood, but use of code strings makes **eval** particularly tricky. As a step towards a more principled analysis, we use a language with Lisp-style staged metaprogramming: programs can construct (**box**), splice together (**unbox**) and execute (**run**) code templates, but code values will always be syntactically valid programs.

Analysis In order to express which values are high, low and intermediate, we extend our language with security markers $\textsc{h}, \textsc{l}, \textsc{i}, \ldots$. They have no computational role; they simply indicate the security level of a value.

Our analysis is based on the popular dataflow analysis CFA [8]. CFA assigns each subexpression of a program a distinct label to track which values it may evaluate to. It generates constraints between labels to express *control and data flow* within the program.

First we extend the analysis to handle staged metaprogramming. Applying an idea from recent work [1], we model code values in a similar way to functions, but with different scoping rules: code values use the scope where they are run, not defined. Next we add constraints to track direct and indirect *information flows*. Here is an example without staging:

$$((\textsf{fun}(x)\{\textsc{i} : (\textsf{fun}(y)\{x\})\})(\textsc{h} : 1))(\textsc{l} : 2) \to^* 1$$

with labels on each subexpression:

$$(((\textsf{fun}(x)\{(\textsc{i} : (\textsf{fun}(y)\{x^0\})^1)^2\})^3(\textsc{h} : 1^4)^5)^6(\textsc{l} : 2^7)^8)^9$$

$$\begin{array}{ccccccc} 4 & \to & 5 & \to & x & \to & 0 \\ & \nearrow^{\textsc{h}} & & \searrow^{\textsc{i}} & & & \\ & & 3 & \rightsquigarrow & 6 & \overset{\searrow}{\rightsquigarrow} & 9 \\ & & & \searrow^{\textsc{i}} & & \nearrow^{\textsc{l}} & \\ & & 1 & \to & 2 & & \end{array} \qquad 7 \to 8 \to y$$

The information flow constraints, direct ($\to$) and indirect ($\rightsquigarrow$), are shown on the right. The result (labelled 9) depends directly on $\textsc{h}$, indirectly on $\textsc{i}$ and not on $\textsc{l}$. The indirect flows result from applying functions, which can reveal the identity of the applied function.

Next we have an example with staging:

$$\begin{array}{l} (\textsf{let}\ x = (\textsc{l} : 1^0)^1\ \textsf{in} \\ (\textsf{let}\ c = (\textsf{box}\, x^2)^3\ \textsf{in} \\ (\textsf{let}\ x = (\textsc{h} : 2^4)^5\ \textsf{in} \\ (\textsf{run}\ c^6)^7)^8)^9)^{10} \to^* 2 \end{array}$$

$$\begin{array}{ccccccccccc} 4 & \to & 5 & \to & x@7 & \leftrightarrow & x@2 & \to & 2 & \to & 7 & \to & 8 & \to & 9 & \to & 10 \\ & \nearrow^{\textsc{h}} & & & & & & & & & & & & & & & \\ & & & & c@2 & \leftrightarrow & c@8 & \to & 6 & \overset{\nearrow}{\rightsquigarrow} & & & \textsc{l} & & & \\ & & & & & \nearrow & & & & & & & & \searrow & & & \\ & & & & 3 & & & & & & & & 0 & \to & 1 & \to & x@9 \end{array}$$

Here **box** x is defined in the scope of the first **let** $x = \ldots$, which is labelled 9; the label $x@9$ tracks the values of x in this scope. But **box** x is run in (and so uses) the scope of the second **let** $x = \ldots$ ($x@7$). Thus the result (labelled 10) depends on $\textsc{h}$, but not $\textsc{l}$.

Future Work We believe all the pieces are now in place for an interesting, principled analysis of JavaScript with **eval**, but it will take a significant effort to combine them. Our analysis only handles a subset of JavaScript's features and is quite coarse in its abstraction of basic datatypes, but our ideas could be applied to a state-of-the-art CFA-based analysis. We also need to show how to transform string-based **eval** soundly into staged metaprogramming; recent research on direct analysis of **eval** suggests a way of doing that [5]. Finally, there are many infrastructural issues concerning how such an analysis would be used in practice.

References

1. Choi, W., Aktemur, B., Yi, K., and Tatsuta, M. Static Analysis of Multi-staged Programs via Unstaging Translation. In *POPL* (2011), pp. 81–92.
2. Denning, D. E. A Lattice Model of Secure Information Flow. *CACM 19*, 5 (1976), 236–243.
3. Fenton, J. S. Memoryless subsystems. *Comput. J. 17*, 2 (1974), 143–147.
4. Goguen, J. A., and Meseguer, J. Security Policies and Security Models. In *IEEE Symposium on Security and Privacy* (1982), pp. 11–20.
5. Jensen, S. H., Jonsson, P. A., and Møller, A. Remedying the Eval that Men Do. In *ISSTA* (2012), pp. 34–44.
6. Lester, M., Ong, L., and Schäefer, M. Information Flow Analysis for a Dynamically Typed Language with Staged Metaprogramming. Submitted to POST 2013.
7. Pottier, F., and Simonet, V. Information Flow Inference for ML. *TOPLAS 25*, 1 (2003).
8. Shivers, O. Control-Flow Analysis in Scheme. In *PLDI* (1988), pp. 164–174.

A Debugger for Communicating Scala Objects

Extended Abstract

Andrew Bate

andrew.bate@cs.ox.ac.uk

Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

Keywords. Concurrency, Scala, CSO, deadlock detection, debugging.

Concurrent programming is an important paradigm for structuring programs and can simplify the design of multi-task systems. Such systems consist of a collection of semi-independent tasks, each with independent data. Identifying and programming each task as a single process is easier than struggling to design and maintain one monolithic process that is responsible for all the tasks [2]. This separation of concerns leads to a natural exploit of the inherent concurrency of the problem.

However, reasoning about message-passing concurrency is still difficult and requires significant effort on the part of the programmer. Many of the techniques employed in debugging sequential programs do not readily apply to concurrent programs, since the interleavings of actions from different processes are not known externally. For example, allowing each process to print to the console when it enters a particular state is of limited use in debugging because it is not known whether those processes entered those states in the same order as the messages were printed. At present, considerable time is spent attempting to reason about some execution of a concurrent program, with limited guarantees that the information obtained is accurate. Consequentially, large multi-threaded applications have often been discouraged [5].

The software tool we have developed seeks to build upon the advantages of message-passing concurrency using Communicating Scala Objects (CSO) [7] and tries to address the concerns outlined above. It permits visualising and reasoning about the runtime behaviour of such concurrent programs.

In particular, the tool developed extracts information about the trace of communications and the network of processes, and constructs models representing the evolution of the concurrent program over time. It also allows *behavioural specifications* to be defined in terms of trace patterns of special events and these are akin to programming with assertions. These specifications can then be checked at runtime. *Stateful specifications*, which are permitted by our tool, scale well with both the length of the trace history and the number of threads. For example of such a specification, we refer the reader to Section 2.3 of [1].

The tool provides comprehensive diagrams depicting the extracted information. The *sequence diagram* shows the time-ordered sequence of communications between processes and also depicts the events logged by processes for behavioural specifications. In addition to the sequence diagram, the event trace of all communications and special events for behavioural specifications is shown in the lower left-hand column.

The *communication network* is the graph that results from instantiating a node for every process and that connects each pair of processes that share a channel. The process *composition trees* illustrate the syntactic way in which the processes were composed together to form the network. Both of these diagrams can be automatically generated by our tool for the program under inspection.

These diagrams provide an aide-mémoire for the programmer, can form part of the documentation of the program, and assist with debugging. For example, if the network diagram unexpectedly contained a cycle, it would be immediately obvious that the network was not as the programmer intended. Futhermore, diagramming the interaction between the running processes can provide an intrinsic explanation of the extensional behaviour of a run of the program.

The tool also dynamically detects deadlock in the program under inspection via periodic analysis of the graph of ungranted requests between processes. This mechanism is optimised so as to only run when the communication graph induced from the processes contains a cycle—a necessary condition of deadlock [6]. Furthermore, this procedure is capable of detecting deadlock in any sub-network of the communication graph. Without such a debugging mechanism, deadlock can be virtually indistinguishable from livelock.

The contract of CSO restricts the usage of certain components, such as limitations on the sharing of channels between processes. Incorrect usage can lead to deadlock or data loss [7]. Our tool will advise the programmer of illegal construct usage. Without such a tool, these errors can be extremely difficult to debug, since the error is not in the programmer's intended design but is in an inadvertent use of CSO.

Real-time data about the runtime environment, including the resource usage of the program, is also provided to the user. This information helps the user to further distinguish between deadlock and livelock.

The overhead of the debugger to the running program is small [1]. This is crucial to the usefulness of such a tool: the timing details and interleavings of the actions of the program under inspection should be altered as little as possible; altering timing details can have a significant impact on whether deadlock will manifest [2].

This work has been published at the *Communicating Process Architectures Conference 2012* [1]. Future work will involve support for the Actors framework of the Scala standard library [4] and incorporating our debugger with the set of tools available for CSP++ [3].

Whilst, for critical applications at least, it is often still necessary to specify the behaviour of the system in a process algebra such as CSP and perform model checking, we believe that our tool can provide an invaluable insight into the behaviour of an implementation.

References

1. Bate, A., and Lowe, G. A Debugger for Communicating Scala Objects. In *Communicating Process Architectures* (2012).
2. Ben-Ari, M. *Principles of Concurrent and Distributed Programming*, 2nd ed. Addison-Wesley, 2006.
3. Gardner, W. B. *CSP++: An Object-Oriented Application Framework for Software Synthesis from CSP Specifications*. PhD thesis, Victoria, B.C., Canada, 2000.
4. Haller, P., and Odersky, M. Actors that Unify Threads and Events. In *Proc. of the 9th Int. Conf. on Coordination Models and Languages* (2007), COORDINATION'07, Springer-Verlag.
5. Muller, H., and Walrath, K. Threads and Swing. *Sun Developer Network* (2000). `http://java.sun.com/products/jfc/tsc/articles/threads/threads1.html`.
6. Roscoe, A. *Understanding Concurrent Systems*, 1st ed. Springer-Verlag New York, Inc., 2010.
7. Sufrin, B. Communicating Scala Objects. In *Communicating Process Architectures* (2008).

Exploring Multitask Learning for Steganalysis

Julie Makelberge

Oxford University Department of Computer Science
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

Steganalysis is the detection of hidden data. There is a lot of literature that incorporates machine learning techniques in steganalysis, but these often consider a single training and testing source. This is problematic because it does not reflect the real world situation. In conventional settings, it is unusual for one actor to generate enough data to be able to train a personalized classifier. On the other hand, in a network there will be many actors, between them generating large amounts of data. Other approaches[5] pool the training data. The approach of [2] is to look for differences between actors, but not to train at all. This paper introduces a new technique for multi-source steganalysis (in this topic, each source is called an "actor"). In this work, we use multitask learning[1] to account for differences between actors' image sources, while still sharing domain (globally-applicable) information.

A specific multitask learning technique tackles this problem by learning separate feature weights for each actor, and sharing information between the actors through regularization. In this case, of course, the training data also distinguishes between the actors. This way, the domain information that is obtained by considering all actors at the same time is not disregarded, but the weights are nevertheless personalized. This paper explores whether this kind of multitask learning improves accuracy of detection, by benchmarking this learner against previous work.

Multitask Logistic Regression

Since this is the first time this technique is applied to the field of steganalysis, the simplest form of multitask learning is used. This work is based on logistic regression, which has been applied to steganalysis[4] but is not common. Logistic regression attempts to model the probability that instance $\mathbf{x}$ comes from class C_1 or C_2, using a weight vector $\mathbf{w}$, by $\log \frac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})} = \mathbf{w}^T\mathbf{x}$. Given a number of training instances $(\mathbf{x}^{(i)}, t^{(i)})$, $i = 1, \ldots, n$, ($t^{(i)}$ is zero or one according to the class) the weights are learned by minimizing the negative log-likelihood function. Logistic regression becomes a binary classifier by picking the most likely class.

In the multi-actor setting, we use the notation $\mathbf{x}^{(a,i)}$ to mean data from actor a and item i, with $x_j^{(a,i)}$ extracting the j-th feature. The m actors are denoted by $a = 1, \ldots, m$. We follow the multitask learner of [3], learning a different weight vector for each actor, using $\mathbf{w}^{(a)}$ to indicate the weights associated with actor a and W the complete parameter matrix with each $\mathbf{w}^{(a)}$ as rows. The total negative log-likelihood is

$$\mathcal{L}(W) = -\sum_{a=1}^{m}\sum_{i=1}^{n} \log P(C_1 \mid \mathbf{x}^{(a,i)})^{t^{(a,i)}} (1 - P(C_1 \mid \mathbf{x}^{(a,i)}))^{1-t^{(a,i)}}.$$

However, simply minimizing $\mathcal{L}(W)$ is equivalent to training a logistic regression classifier for each actor separately; there is no transit of information between them. To make this a multitask learner, we add a regularization term, minimizing $\mathcal{G}(W) = \mathcal{L}(W) + \mathcal{R}(W)$ where

$$\mathcal{R}(W) = \sum_{j=1}^{d} \left(\frac{u_j^2}{\sigma_1^2} + \frac{1}{\sigma_2^2} \sum_{a=1}^{m} (w_j^{(a)} - u_j)^2 \right).$$

The vector **u** represents global weights (which, at the optimum, are simply a function of the local weights) and this regularization corresponds to a prior in which **u** arises from a Gaussian distribution and the difference between each of the personalized weights and the global weight is then also assumed to come from a Gaussian distribution.

Experiments and Results

We wish to compare multitask logistic regression against alternative steganalysis classifiers, given training data from multiple actors. To replicate this scenario, our dataset consists of images that were taken from one of the leading social networking sites. The people uploading the pictures are considered to be the actors, as they will be using different cameras. This makes available hundreds of actors each with up to 4000 JPEG cover images. Plain logistic regression, regularized with a Gaussian prior, is used as a base case for comparison of results. In all cases we use BFGS as the numerical minimizer for the weights: this is nontrivial as the multitask learner is optimizing over many dimensions (one per actor and feature).

We create stego data by embedding using nsF5. In this abstract, we present results with a payload of 0.1bpnc (bits per nonzero DCT coefficient). Here, we tested 26 actors with 50 images per actor, 100 images per actor and 200 images per actor, optimizing the weights using 5-fold cross-validation. We implemented both regularized logistic regression, using pooled training data and personalized data, and a regularized multitask learner. The table presents the results and it indicates that multitask learning may provide new advances in steganalysis, for situations when multiple actors are using different image sources and few images are available per actor.

	50 images/actor	100 images/actor	200 images/actor
Pooled	81.15%	81.73%	82.45%
Personalized	80.19%	82.14%	86.39%
Multitask	84.04%	85.58%	86.59%
z-score multitask > pooled	1.15	2.37*	2.01*
z-score multitask > personalized	1.48	2.03*	0.06

Table 1. Comparison of classifiers tested. The percentages indicate the accuracy of the different models on the testing data. The comparison between these percantages of accuracy is made using a z-test. This test generates a z-score, which indicates whether the comparison is statistically significant. Any z-score above 1.96 is statistically significant.

References

1. Caruana, R., Pratt, L., and Thrun, S. Multitask learning. *Machine Learning 28*, 1 (1997), 41–75.
2. Ker, A. D., and Pevný, T. Identifying a steganographer in realistic and heterogeneous data sets. In *Media Watermarking, Security, and Forensics XIV* (2012), N. D. Memon, A. M. Alattar, and E. J. Delp III, Eds., vol. 8303 of *Proc. SPIE*, SPIE, pp. 0N01–0N13.
3. Lapedriza, Á., Masip, D., and Vitrià, J. A hierarchical approach for multi-task logistic regression. In *Proc. 3rd Iberian Conference on Pattern Recognition and Image Analysis* (2007), Springer, pp. 258–265.
4. Lubenko, I., and Ker, A. D. Steganalysis using logistic regression. In *Media Watermarking, Security, and Forensics XIII* (2011), N. D. Memon, J. Dittmann, A. M. Alattar, and E. J. Delp III, Eds., vol. 7880 of *Proc. SPIE*, SPIE, pp. 0K01–0K11.
5. Lubenko, I., and Ker, A. D. Steganalysis with mismatched covers: do simple classifiers help? To appear in Proc. ACM Multimedia and Security Workshop MMSec'12, 2012.

Hardware Security for Device Authentication in the Smart Grid

Andrew J. Paverd Andrew P. Martin

University of Oxford Department of Computer Science
Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom

The *Smart Grid* is envisaged to be a significant upgrade of the public energy distribution infrastructure, in which modern computational and communication technologies are used to maximize the efficiency of the energy grid. By enabling two-way communication between all nodes, the smart grid will facilitate unprecedented levels of automation in energy management systems. This will extend to residential energy management and will include automated remote control of individual appliances in the smart home. Given this enhanced functionality, the security of the communication between smart grid entities is a primary concern.

Device authentication is a particularly important aspect of communication security in the smart grid. In particular, Metke and Ekl [3] explain how public key infrastructure (PKI) can provide significant benefit to the smart grid. Kuntze et al. [1] argue that the smart grid requires a clear distinction between the authentication of devices and users. In this context, some devices may have multiple users whilst others may frequently operate unattended. The latter presents a significant challenge for device authentication. Most secure communication protocols such as Transport Layer Security (TLS) require every device to possess a unique secret such as an RSA private key. On general purpose devices such as PCs, these keys may be vulnerable to be compromised by malware and so must be adequately protected. However, current approaches for protecting device private keys almost always require the presence of a user to provide a password for decryption or a secure token on which the key is stored. These are infeasible in the smart grid since devices will usually be controlled automatically. This results in the *unattended start-up problem* in which a device must be able to start up and reach a fully functional state without user intervention.

Trusted Computing (TC) provides two potential solutions for this problem using the Trusted Platform Module (TPM): *Binding* a key to a TPM means that the key can only be used on a specific system. *Sealing* a key ensures that it can only be used by a specific system which is in a predefined state. However, binding a key is insufficient to protect it from potential abuse by malware because this mechanism cannot distinguish between malware and authorized applications. Additionally, sealing a key against an entire system state is not practical given the frequently-changing nature of software on general purpose systems.

We have developed a novel solution to this problem based on TC technology which overcomes these limitations. The fundamental concept is that the device's private key is sealed against a trusted execution environment (TEE) rather than the entire system. We have demonstrated this principle through a proof-of-concept implementation of a system for protecting the private key used in a TLS handshake. For this implementation, we modified the PolarSSL library to incorporate this functionality and used the *Flicker* research project [2] to provide a hardware-enforced isolated execution environment which serves as the TEE.

During the initialization phase of the system, the device private key is either input into the TEE or generated by the TPM. Two primary restrictions on the use of the key are defined in this phase: Firstly, the TEE is provided with the hash-based identifiers of the applications which may use the key. Secondly, the TEE is provided with a certificate

authority (CA) certificate which will be used to verify the certificates of all other connecting devices. Using the TPM, the private key is then sealed against the current state of the TEE and the specific usage restrictions for the key. This initialization phase takes place once for each key and results in an encrypted data structure which can be stored outside the TEE.

The main operational phase of the system takes place when a TLS session is being established and the use of the private key is required. For this system, it is assumed that the private key is an RSA signing key and that ephemeral Diffie-Hellman parameters are used to generate the master secret since this is a common configuration. It is also assumed that the device takes the client role in the TLS handshake as this would usually be the case for an appliance in a smart home communicating with an external smart grid entity. In this configuration, the client proves its possession of the private key by using this key to sign a digest of the TLS handshake messages.

When this signature is required, the system temporarily suspends the host OS and launches the TEE. As part of the TEE launch process, the hash-based identifier of the application requesting the signature is implicitly input to the TEE and the TEE verifies that this application is permitted to use the key. The TLS handshake messages are also input to the TEE and are parsed to obtain the server's certificate and verify that this is signed by the specified CA. If these verification steps are successful, the private key is unsealed and used to sign the handshake digest. The host OS is then resumed and the required signature is returned to the requesting application to complete the TLS handshake protocol. During this procedure, the private key never leaves the TEE in unencrypted form.

One metric used to evaluate this system is to measure the additional time taken by this enhanced functionality. Since the host OS is suspended for the duration of the TEE execution, an important objective in this work was to minimize the time taken by the software component executing within the TEE. In the proof-of-concept implementation, this execution time was measured to be approximately 980 ms. This consists predominantly of the time taken by the TPM to perform the *unseal* operation, which represents the minimum possible execution time for a system using this technology. Although this would be a noticeable delay for a user, the primary objective of the system is to facilitate use of the private key in an unattended scenario. Therefore, the primary requirement with respect to performance was that the execution time should be sufficiently short to operate within the constraints of the TLS handshake protocol. Comprehensive testing has shown that the proof-of-concept system meets this performance requirement.

Overall, this system demonstrates the feasibility of this concept and provides a mechanism for protecting device private keys in an unattended environment. It is anticipated that this functionality will be required to support the high level of automation in the smart grid. In general, it could be argued that this concept allows for information to be sealed to a specific purpose rather than to a specific system state. Future work on this topic will investigate how this concept can be applied to other forms of confidential information.

References

1. Kuntze, N., Rudolph, C., Cupelli, M., Liu, J., and Monti, A. Trust infrastructures for future energy networks. In *IEEE PES General Meeting* (July 2010), IEEE, pp. 1–7.
2. McCune, J. M., Parno, B. J., Perrig, A., Reiter, M. K., and Isozaki, H. Flicker: an execution infrastructure for TCB minimization. In *Eurosys '08 Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems* (Apr. 2008), vol. 42, p. 315.
3. Metke, A. R., and Ekl, R. L. Security Technology for Smart Grid Networks. *IEEE Transactions on Smart Grid 1*, 1 (June 2010), 99–107.

Neighbourhood Watch

Network Coding Efficiency Depends on Good Neighbours

Martin Strohmeier,
Supervised by Ivan Martinovic

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

Network coding has emerged as an exciting new topic in networking research. It has attracted considerable attention since it allows network throughput to be increased under certain conditions while keeping bandwidth usage steady [1]. Nodes overhearing several messages may create linear combinations before re-broadcasting the received messages in combined form. Receivers can use several received combined messages to restore all individual messages from senders. Thus, the information flow can be increased while using the same number of transmissions. [1]

Network coding has been enthusiastically promoted as a means of improving throughput, especially in wireless networks where the inherent broadcast nature of the medium seems predisposed to employ the concept. Since its introduction in 2000, network coding has been successfully applied to improve various applications, e.g. peer-to-peer content distribution systems [3] and plain wireless networks [4]. Whilst our research focuses on applications of network coding in wireless networks, the findings may well apply to other types of networks.

Existing work has largely concentrated on the fundamental aspects of network coding and mostly neglected constraints found in practical network deployments. One of these constraints is a given network's neighbourhood. Generally speaking, the throughput benefit of coding over traditional routing in wireless settings depends on the ability of nodes to overhear each others' transmissions. Unfortunately, in practical scenarios it cannot be assumed that all nodes in physical transmission range can or should overhear broadcasted packets. Some reasons can be found in the network design such as transmission scheduling or security.

However, in many practical deployments it is possible to control which nodes are able to overhear transmissions. It is therefore important to analyse and quantify the possible impact of neighbourhood size and composition on network coding efficiency. By deriving theoretical upper bounds for randomized networks, we show that network coding can still be effective with small neighbourhoods if the composition is optimal. However, finding the optimal neighbourhood composition is NP-hard and heuristics can be necessary for some larger-scale, dynamic application scenarios. We propose such heuristics and show that these are able to find solutions with an acceptable efficiency loss compared to the optimal solution.

One concrete example would be a network that uses a medium access control (MAC) where only nodes that are part of the same cluster can overhear transmissions. This can e.g. be the case when utilizing a round-based TDMA schedule. To ensure transmission efficiency, this cluster size may be limited. Thus, even though many nodes may be in transmission range, not all nodes will be able to overhear transmissions since they are part of different clusters. In other words, while cluster size and network layout might be constrained, it may still be possible for a network designer to choose precisely how to group nodes into clusters.

Moreover, security is perhaps the most important problem in connection with network coding. It is not advisable to equip all nodes in a network with the same key; a single lost key

[1] For a full primer on the underlying network coding theory, see [2].

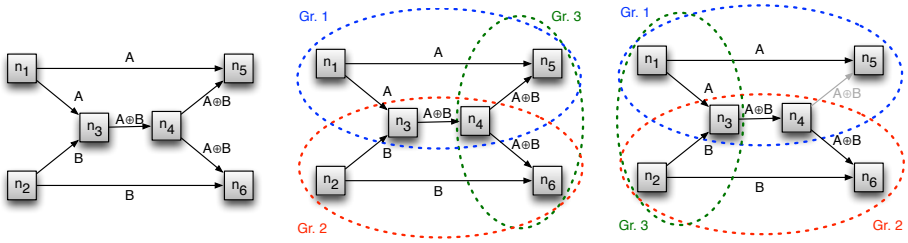


Fig. 1. Example of network coding with different limitations on neighbourhoods. The plain butterfly network (left) sends two messages A and B from n_1 and n_2, respectively, to **both** sinks n_5 and n_6 with only one round of transmissions (exhausting all link capacities). The wrong choice of groups (right) impacts the information flow: n_4 can send only to one of the sinks at a time, effectively taking away the advantage over traditional routing. The better choice (middle) preserves the coding gain.

should never lead to an immediate and complete loss of protection for the whole network. Thus, to increase protection it can make sense to form smaller groups of nodes that each share a key. Only nodes sharing a common key are able to use an overheard transmission for network coding purposes. The number of nodes sharing a common key may be limited in this scenario but it is still possible to decide how nodes are grouped together (cf. Fig. 1 for an example). Effectively, this means that there is a trade-off between network coding efficiency and security, which needs to be quantified.

As a consequence of the previous observations, we argue that it is often unreasonable to assume that all neighbours in a node's range can overhear its transmissions. From a network coding perspective it would therefore be advisable to group nodes such that network throughput is maximized. In our work, we analyse and quantify the impact of neighbourhood size and composition on network coding efficiency. We employ linear programming as a means to derive theoretical upper bounds for various different classes of networks. Specifically, we make the following contributions:

- **Neighbourhood Size:** We quantify the impact of neighbourhood size on network coding efficiency in terms of throughput and demonstrate the trade-off with security.
- **Neighbourhood Composition:** We give neighbourhood composition heuristics which enable us to find near optimal neighbourhood compositions quickly.
- **Real-World Analysis:** We demonstrate the impact of neighbourhood size and composition on coding efficiency using real-world wireless network configurations.

Future research includes simulations of practical scenarios to support the derived graph-theoretic upper bounds, and validate the efficiency of the developed heuristic algorithms.

References

1. Ahlswede, R., Cai, N., Li, S.-Y., and Yeung, R. Network information flow. *Information Theory, IEEE Transactions on 46*, 4 (July 2000), 1204 –1216.
2. Fragouli, C., and Soljanin, E. Network coding fundamentals. *Foundations and Trends in Networking 2* (January 2007), 1–133.
3. Gkantsidis, C., and Rodriguez, P. Network coding for large scale content distribution. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* (March 2005), vol. 4, pp. 2235 – 2245.
4. Katti, S., Rahul, H., Hu, W., Katabi, D., Médard, M., and Crowcroft, J. Xors in the air: practical wireless network coding. *SIGCOMM Comput. Commun. Rev. 36* (August 2006), 243–254.

Multi-Channel Key Distribution Protocols Using Visible Light Communications in Body Sensor Networks

Xin Huang, Bangdao Chen, A. W. Roscoe

Department of Computer Science
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

Body sensor networks (BSNs) typically are wearable wireless sensor networks that consist of one sink node and several sensor nodes. BSNs have the potential to revolutionize many fields including patient monitoring and sports training. Given the examples considered, the data transmitted is sensitive and will represent a serious threat to privacy. Generally speaking, security protection of BSNs relies on key distribution protocols. In the past decade, most key distribution protocols in BSNs are based on pre-deployed secrets, which are difficult to be updated during the time, for example, when they are compromised after deployment. Recent research on visible light communications and asymmetric cryptography brings new possibilities.

Visible light communication (VLC) channels are data communication channels using visible light roughly between 400 nm and 700 nm. The senders, i.e. light sources, include displays and LEDs; and the receivers include light sensors and cameras. Two example VLC channels are listed as follows: (1) VLC channel is established using LEDs and a camera. Data is firstly encoded into a message frame using Manchester coding. Secondly, the frame is modulated to LED light signals using on-off keying method, i.e. presence and absence of light. Finally, the light signal is automatically recognized by a fixed camera. The bandwidth is always limited due to the maximum sample rate of camera. (2) VLC channel is established using LEDs and a light sensor (only sensitive to visible light). The procedure is similar with the first example, but the transmission speed can be high.

Human controlled visible light communication (HVLC) channels are established based on VLC channels. Human users should prevent any physical interference to visible light signals or communication devices. In this case, attackers would find it difficult to block, modify, delay, replay, or fake messages transmitted over HVLC channels. Thus, HVLC channels are no-spoofing and no-blocking (NSB) channels.

Using HVLC channels, we can improve on the Elliptic-Curve Diffie-Hellman (ECDH) protocol in body sensor networks [1]. ECDH works as follows. Assume that the key pair of sink node $Sink$ is (sk_{Sink}, pk_{Sink}), and the key pair of the ith sensor node is (sk_i, pk_i). The two parties exchange the public keys, and calculate the shared key $K = sk_{Sink}[pk_i] = sk_{Sink}[sk_i[G]] = sk_i[pk_{Sink}]$, where G is the public base point [1]. However, it is well known that ECDH suffers from man-in-the-middle attack. We have designed two improved protocols: a hash-based two-channel ECDH (HT-ECDH) protocol and an ECDH version of parallel hash commitment before knowledge (ECDH-PHCBK) protocol. HT-ECDH uses high speed HVLC channels. The protocol is listed as follows.

1. $Sink \longrightarrow \forall Sensor_i : Sink, pk_{Sink}$. Sink node $Sink$ sends each sensor node $Sensor_i$ its ID $Sink$ and public key pk_{Sink} over wireless channels.
   Actions: User inputs group size $group\_size$ into $Sink$.
2. $\forall Sensor_i \longrightarrow Sink : Sensor_i, pk_i$. Each $Sensor_i$ sends its ID $Sensor_i$ and public key pk_i to $Sink$ over wireless channels.
   Actions: (1) $Sink$ verifies the group size; if verification fails, $Sink$ aborts and informs user. (2) Each $Sensor_i$ computes $hash_i = \mathcal{H}(Sink, pk_{Sink}, Sensor_i, pk_i)$, where $\mathcal{H}()$ is a hash function.

3. $\forall Sensor_i \Longrightarrow_{HVLC} Sink : Sensor_i, hash_i$. Each $Sensor_i$ sends ID $Sensor_i$ and $hash_i$ to $Sink$ via the HVLC channel.
   Actions: $Sink$ verifies $hash_i = \mathcal{H}(Sink, pk_{Sink}, Sensor_i, pk_i)$, and the group size is $group\_size$. If any verification fails, $Sink$ aborts and informs user.

ECDH-PHCBK uses low bandwidth HVLC channels. There are two differences. (1) $hash_i$ in step 3 of HT-ECDH is too long to be transmitted over these channels, thus we use a digest function in ECDH-PHCBK instead. A digest function $\mathcal{D}()$ takes as input a message msg of arbitrary length and produces as output a short message digest $digest$ of fixed length ($digest$ is usually 16 bits) [2]. (2) $\mathcal{D}()$ is vulnerable to combinatorial attack: given one input msg_1, attackers can find another input msg_2 such that $\mathcal{D}(msg_1) = \mathcal{D}(msg_2)$. In order to eliminate this attack, a hash commitment scheme is used. The hash commitment scheme consists of two phases: (a) Commit phase: sender chooses a hidden long random value $nonce$ and sends out commitment $hash\_commit = \mathcal{H}(nonce, \cdots)$. (b) Reveal phase: sender sends an opening $nonce$. The ECDH-PHCBK protocol is listed as follows.

Commit phase:

1. $Sink \longrightarrow \forall Sensor_i : Sink, pk_{Sink}, hash\_commit$. $Sink$ sends $Sensor_i$ its ID $Sink$, public key pk_{Sink} and the commitment $hash\_commit$ over wireless channels.
   Actions: $Sink$ generates $nonce$ and $hash\_commit = \mathcal{H}(nonce, Sink)$.
2. $\forall Sensor_i \longrightarrow Sink : Sensor_i, pk_i$. Each $Sensor_i$ sends its ID $Sensor_i$ and public key pk_i to $Sink$ over wireless channels.
   Actions: (1) Each $Sensor_i$ finished step 1 and 2 flashes LEDs. (2) User verifies that each $Sensor_i$ has flashed and group size is $group\_size$. If verification is successful, user informs $Sink$ to release opening message in step 3.

Reveal phase:

3. $Sink \longrightarrow \forall Sensor_i : nonce$. $Sink$ sends the opening message $nonce$ to each $Sensor_i$ over wireless channels.
   Actions: (1) $Sensor_i$ verifies that $hash\_commit = \mathcal{H}(nonce, Sink)$. If the verification is fails, $Sensor_i$ aborts and informs user. (2) $Sensor_i$ computes $digest_i$, which is $\mathcal{D}(nonce, Sink, pk_{Sink}, Sensor_i, pk_i)$.
4. $\forall Sensor_i \Longrightarrow_{HVLC} Sink : Sensor_i, digest_i$. $Sensor_i$ transfers its ID $Sensor_i$ and the digest value $digest_i$ to $Sink$ via the HVLC channel.
   Actions: $Sink$ verifies the equality: $digest_i = \mathcal{D}(nonce, Sink, pk_{Sink}, Sensor_i, pk_i)$ for each $Sensor_i$. If verification fails, $Sink$ aborts and informs user.

In conclusion, we have designed HVLC channels for BSNs. They reduce the human burden. In addition, the two protocols can eliminate man-in-the-middle attack against ECDH. Besides, the digests in ECDH-PHCBK are not the same for different sensor nodes, which is different from HCBK [2]. It works better when connections among sensor nodes are not necessary.

References

1. Liu, A., and Ning, P. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Proceedings of the 7th international conference on Information processing in sensor networks* (2008), IEEE Computer Society, pp. 245–256.
2. Nguyen, L. H., and Roscoe, A. W. Authenticating ad hoc networks by comparison of short digests. *Information and Computation 206*, 2-4 (2008), 250–271.

Experimentally-calibrated population of models predicts and explains inter-subject variability in cardiac cellular electrophysiology

Oliver Britton

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

Biological variability is exhibited at all levels in all organs of living organisms. It manifests itself as differences in physiological function between individuals of the same species, and often more drastically by significant differences in the outcome of their exposure to pathological conditions. Thus, healthy cardiac cells of the same species and location exhibit a qualitatively similar response to a stimulus, i.e. the action potential (AP), which is the pattern of electrical activity generated by a cardiac cell due to the flow of ionic currents through ion channels in the cell membrane, following an initial electrical excitation of the cell. However, between individuals significant quantitative differences exist in AP morphology and duration, which may explain the different individual response of each of the cells (and patients) to disease or drug action. The variability underlying the physiological and pathological response of different individuals has often been ignored in experimental and theoretical research, ultimately hampering the extrapolation of results to a population level. Experimentalists often average the results obtained in different preparations to reduce experimental error, therefore also averaging out the effects of inter-individual variation and resulting in an important loss of information. This averaging of experimental data is inherited by theoretical research, and consequently, models are often developed for a 'typical' behaviour within a particular population [2]. Therefore, whereas all experimentally-measured APs are different even within a homogeneous population, a single AP model is obtained from the data, again losing all information regarding inter-subject variability.

We tightly couple experimental measurements and mathematical modelling to construct and calibrate a population of cardiac electrophysiology cell models representative of physiological variability, which we then use to investigate the causes of experimentally-measured variability in physiological conditions and following drug response. Our research builds on previous studies by us and others [4–6] showing the importance of mathematical methods such as model populations and sensitivity analysis in investigating the ionic determinants of inter-individual variability in biological properties.

We have generated a population of cell models that is able to represent the variability exhibited in specific experimental recordings under physiological conditions and to predict inter-subject variability in the response to potassium channel block. We base our investigations on rabbit Purkinje electrophysiology, due to the importance of Purkinje fibres in lethal arrhythmias and in drug testing in preclinical safety pharmacology [1]. We hypothesize that inter-subject variability in experimentally-measured APs is primarily caused by quantitative differences in the properties of ionic currents (e.g. how many channels of a particular type are present in the membrane), rather than by qualitative differences in the biophysical processes underlying the currents (e.g. which types of channel are present at all in the cell). The equations proposed in the rabbit Purkinje AP Corrias-Giles-Rodriguez (CGR) model [3] are considered as the model structure to generate over 10,000 candidate models, all sharing the same equations (i.e. the same ionic biophysical processes) as in the original CGR model, but with different parameter values for important ionic current conductances and time constants, randomly selected within a wide range. The cell model population is then calibrated

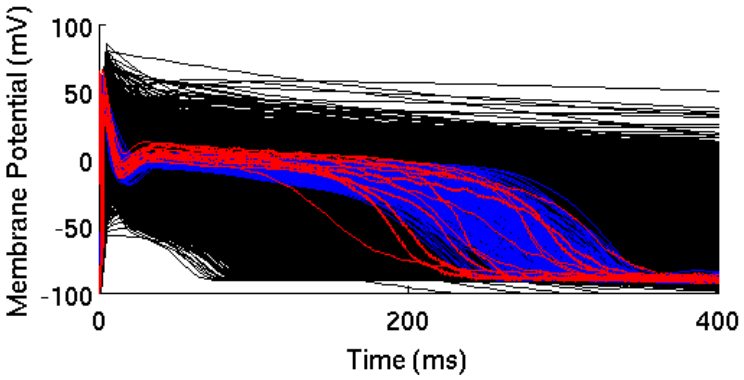


Fig. 1. Simulated output of calibrated population of models (blue traces) compared to experimental AP traces (red) and models rejected from the population due to being out of range on one or more measures of AP behaviour at one or more pacing frequencies (black traces). Results shown for pacing at 1 Hz.

using a set of cellular biomarkers extracted from experimental AP recordings at three pacing frequencies in order to discard models that are inconsistent with experimental data. This process discarded all but 213 of the candidate models. We then used the calibrated model population to identify the ionic mechanisms determining intra-species variability in each biomarker using regression analysis, yielding information on the relative importance of ionic currents in the generation of the AP at each pacing frequency. Following this we showed that the calibrated model population quantitatively predicts the prolongation of AP duration (APD) caused by exposure to four doses of dofetilide, a blocker of the rapid component of the delayed rectifier potassium current (I_{Kr}). We chose I_{Kr} block as the intervention to evaluate the predictive power of our population of models as I_{Kr} block is the main assay required in safety pharmacology assessment. These results indicate that our method can be used to predict the range of behaviours of an independent data set and link that behaviour to underlying mechanisms. The flexibility of our methodology to construct and calibrate populations of models means it can be easily applied to other areas of biology.

References

1. The European Medicines Agency Safety Pharmacology Studies for Human Pharmaceuticals, 2005.
2. Carusi, A., Burrage, K., and Rodrguez, B. Bridging experiments, models and simulations: an integrative approach to validation in computational cardiac electrophysiology. *American Journal of Physiology - Heart and Circulatory Physiology 303*, 2 (2012), H144–H155.
3. Corrias, A., Giles, W., and Rodriguez, B. Ionic mechanisms of electrophysiological properties and repolarization abnormalities in rabbit Purkinje fibers. *Am J Physiol Heart Circ Physiol 300*, 5 (2011), H1806–13.
4. Marder, E., and Taylor, A. L. Multiple models to capture the variability in biological neurons and networks. *Nat Neurosci 14*, 2 (2011), 133–8.
5. Romero, L., Pueyo, E., Fink, M., and Rodriguez, B. Impact of ionic current variability on human ventricular cellular electrophysiology. *Am J Physiol Heart Circ Physiol 297*, 4 (2009), H1436–45.
6. Sarkar, A. X., Christini, D. J., and Sobie, E. A. Exploiting mathematical models to illuminate electrophysiological variability between individuals. *J Physiol 590*, Pt 11 (2012), 2555–67.

Mathematical Models of Tissue Engineering

The influence of cell seeding strategy and culture conditions on the growth of cell aggregates on a permeable membrane

Lloyd Chapman
Professor Helen Byrne, Dr Rebecca Shipley, Dr Jon Whiteley, Dr Sarah Waters

Mathematical Institue and Department of Computer Science, University of Oxford

In vitro tissue engineering is the process of growing cells and tissues in the laboratory for implantation into patients to repair or replace damaged or lost tissues. One way tissue engineers do this is to put cells into or onto a porous scaffold (a process known as cell seeding), and then incubate the cells in a device known as a bioreactor. The scaffold provides a structure for the cells to grow on and the bioreactor enables the nutrient and mechanical environment of the cells to be controlled. In dynamic culture systems, like the one we consider, nutrient delivery to the cells is enhanced by perfusing the scaffold with a nutrient-rich culture medium.

Some simple relationships between cell seeding distribution, seeding density, flow conditions and tissue growth have been established for specific bioreactors and cell types via experiments [2]. However, to be able to analyse the effect of the seeding strategy and operating conditions on the tissue growth in more detail, we need to develop theoretical models of the tissue growth that consider the different length and time scales and interactions of the physical and chemical processes involved.

To this end, we have developed and analysed a simple 2D continuum model of the growth of cell aggregates along the surface of a permeable membrane, past and through which a nutrient-rich culture medium flows. The problem is motivated by the set-up of a single-fibre hollow fibre membrane bioreactor (HFMB) [1] (Fig. 1).

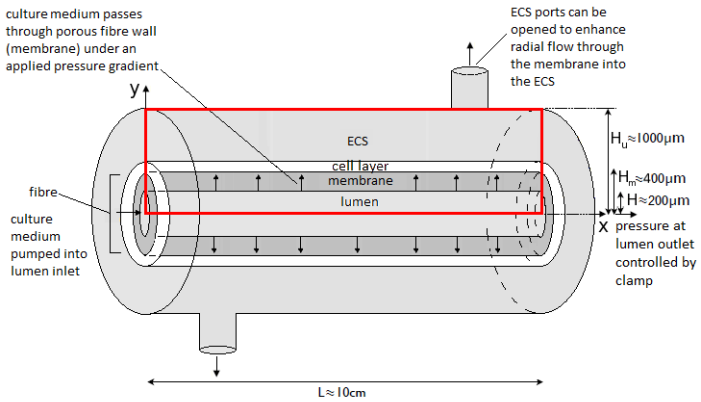


Fig. 1: **Schematic of the single-fibre hollow fibre membrane bioreactor used in the laboratory.** Red rectangle shows the 2D model region. Arrows show direction of fluid flow.

Cells seeded on the surface of the fibre in a HFMB often grow unevenly over the surface when cultured and tend to form aggregates of varying cell densities. This may be due to a number of factors, including the efficiency of the cell seeding process and the suitability of the subsequent culturing conditions. In our model we describe the fluid flow past and through the membrane by Stokes flow and Darcy flow respectively, use this to find the oxygen distribution via an advection-diffusion equation, and then determine the cell aggregate growth from the oxygen distribution. We assume that the cell aggregates are infinitesimally thin, provide

resistance to the flow and act as line sinks in the oxygen field. Cell growth and proliferation are assumed to occur only at the ends of the aggregates at a rate proportional to the oxygen concentration there. When two aggregates come into contact with each other they are assumed to merge to form one larger aggregate. We use the model to assess how the flow conditions, specifically the flow rate of the culture medium and the pressure imposed at the outlet of the lower region, and the initial aggregate distribution affect the aggregate growth, in terms of the time taken for the aggregates to reach confluence (grow across the entire membrane).

Numerical simulations of the model reveal that the time to confluence is inversely proportional to both the flow rate and lower region outlet pressure. This is because increasing these parameters leads to a greater flux of fluid through the membrane, improving the delivery of oxygen to the aggregates (Fig. 2). The rate of aggregate growth eventually plateaus above a certain flow rate and outlet pressure due to oxygen transport limitations (the oxygen concentration is highest at the inlet, so the aggregates cannot grow faster than this concentration would allow). The maximum possible growth rate is also limited by certain mechanical effects that we have neglected, such as shear stress on the cells due to the flow.

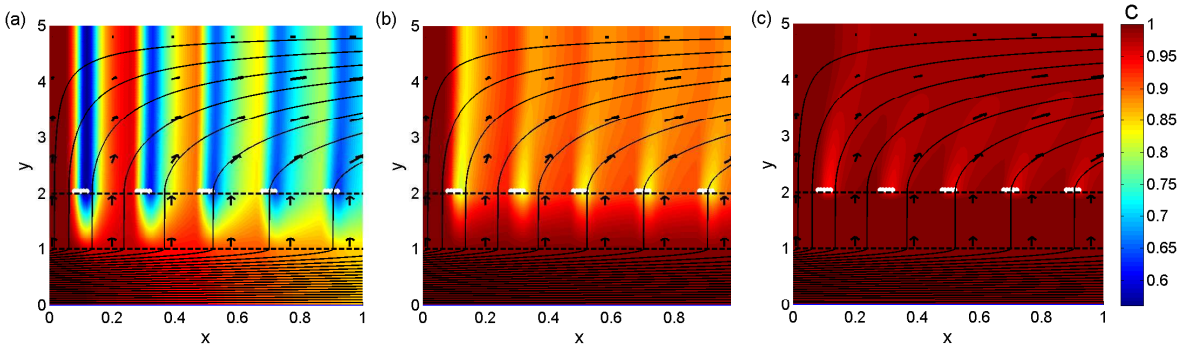


Fig. 2: **Nutrient distribution following seeding for different flow regimes.** (a) Diffusion-dominated, (b) Balanced diffusion and advection, (c) Advection-dominated.

The same number of aggregates of the same initial length (covering 25% of the membrane) grow to confluence more quickly when they are initially spaced evenly along the membrane, i.e. when the spaces between the aggregates are equal and as a large as they can be, than when they are clumped together. This is because the aggregates can grow for longer before merging with their neighbouring aggregates.

Our findings suggest possible ways of improving tissue growth rates in HFMBs and similar bioreactor systems. However, to improve the biological realism of the model, further investigation of the impact of the aggregate growth law on the rate of growth, e.g. considering laws based on the cells proliferating all along the aggregate, and the effect of waste product levels on the growth are necessary. We also need to assess the effect that modelling a 2D slice of the bioreactor instead of the full cylindrical geometry has on the results. Beyond this, we need to consider the circumstances under which it is more appropriate to use a discrete cell-based model of the tissue rather than a continuum model, e.g. for small cell numbers and/or low cell densities, or a hybrid model that combines a discrete cell-based model with a continuum model, e.g. for highly heterogeneous tissue.

References

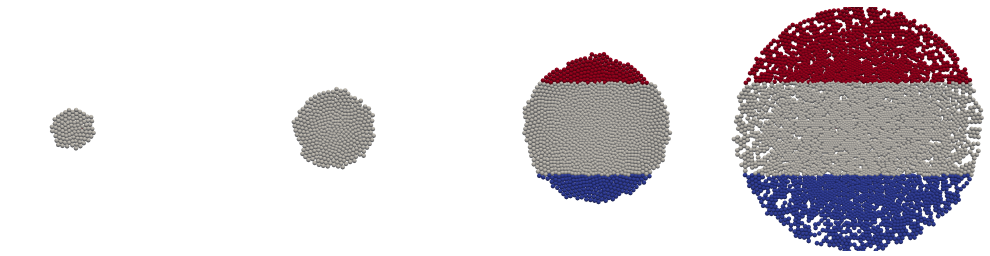
1. Ellis, M., and Chaudhuri, J. Poly (lactic-co-glycolic acid) hollow fibre membranes for use as a tissue engineering scaffold. *Biotechnology and Bioengineering 96*, 1 (2007), 177–187.
2. Martin, I., Wendt, D., and Heberer, M. The role of bioreactors in tissue engineering. *Trends in Biotechnology 22*, 2 (2004), 80 – 86.

A distributed algorithm for simulating an off-lattice model of a population of cells

Daniel Harvey

Computational Biology Group
Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

This talk will present a novel approach for simulating a model of a population of mammalian cells on distributed parallel computers. There are many approaches to modelling populations of cells, from relatively simple sets of Ordinary Differential Equations (ODEs) which we can solve analytically [1], to computational models that treat each cell as a discrete entity and are consequently costly to simulate even for small cell numbers [2]. The benefit of such 'discrete' approaches is that we can study the population-level effects of perturbations at the single-cell level. For example, we can better model the growth of a cancer from a single aberrant cell to a solid tumour. However, such simulations are computationally costly. Serial algorithms currently limit simulations to approximately 10^5 cells, while a solid tumour is not visible on an X-ray until it contains around 10^8 cells. The development of effective therapeutic strategies using this model clearly depends on being able to extend the scale at which we can simulate these models. By harnessing the power of parallel computers we will be able to extend the reach of these model beyond current limitations.



(a) $t = 25h$ (b) $t = 50h$ (c) $t = 75h$ (d) $t = 100h$

Fig. 1. A time-course visualisation of a simulation of a growing population of cells, computed on three processor nodes. Each cell is coloured according to its processor.

The model we consider represents each individual cell as a sphere around a single point in space. Model cells interact with near neighbours through a 'linear-spring' type force along the line joining their centres, which models the adhesive/repulsive interaction of real cells. Models of subcellular processes (such as the cell cycle) can be included for each cell by the simulation user. Our parallel algorithm and implementation is a general approach for this type of cell-level model that can adapt to different user-defined sub-cellular models, as well as different models of cell-cell interaction.

To develop a distributed algorithm that enables our simulation to be executed on distributed parallel computers we divide the simulation domain into segments for each processor to simulate, and use the Message Passing Interface (MPI) model to communicate data between processors owning neighbouring space segments. The segments are chosen by splitting the domain into strips along one of the co-ordinate axes. Figure 1 shows the spatial

decomposition for a simulation on three processors of a growing population in 2D, with each cell coloured according to which processor it lies on. We have validated our algorithm by extensive comparison with results of simulations using the serial algorithm.

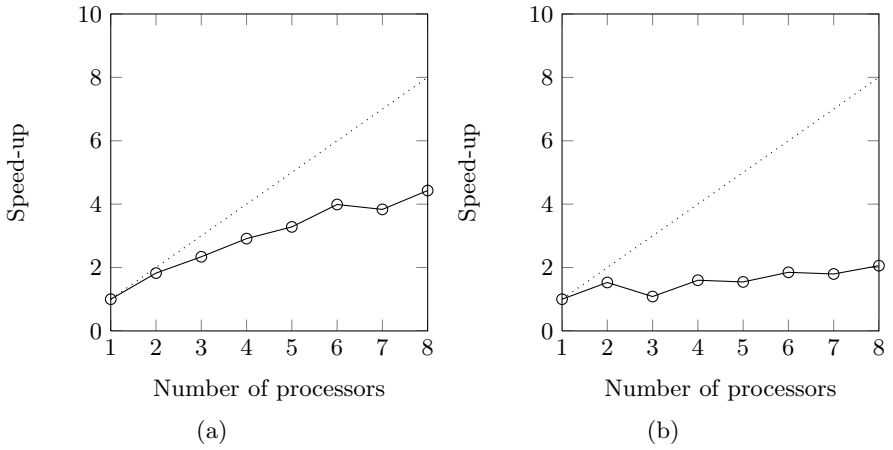


Fig. 2. Representative speed-up results from different simulations on one node of the OSC HAL cluster: (a) An evenly distributed population of cells in a cubic domain. (b) A 2D monolayer of cells growing inside a square domain.

One of the problems faced by this approach is that the level of detail associated with each cell makes the scaling of even load-balanced simulations relatively poor, due to high communication over-heads. Figure 2(a) shows the speed-up achieved on a single node of a cluster for an evenly distributed population compared to the approximate theoretical limit of speed-up (dotted line). We see that for 8 processors approximately half of the possible speed-up is achieved. Furthermore, for populations with irregular shapes a static decomposition leads to poor load balance, which further decreases parallel efficiency. Figure 2(b) shows this effect for a growing 2D monolayer of cells similar to the simulation shown in Figure 1. In this case the distribution of computational work is not evenly spread across the computing resources which leads to very limited actual speed-up. Future development will address these problems by developing a dynamic approach to load-balancing, to take account of the dynamic nature of cells.

References

1. Sachs, R., Hlatky, L., and Hahnfeldt, P. Simple ODE models of tumor growth and anti-angiogenic or radiation treatment. *Math. Comput. Model. 33* (2001), 1297–1305.
2. Schaller, G., and Meyer-Hermann, M. Multicellular tumor spheroid in an off-lattice Voronoi-Delaunay cell model. *Phy. Rev. E 71* (2005), 051910+.

The author would like to acknowledge the use of the Oxford Supercomputing Centre (OSC) (www.osc.ox.ac.uk) in carrying out this work.

Spatial Stochastic Modelling of Gene Regulatory Mechanisms: Why Space Matters

Anna Jones
Supervised by Prof. Kevin Burrage and Prof. Helen Byrne

Oxford University Department of Computer Science
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

Computational simulation of biological systems can provide us with important insights into the dynamics of a cell. In particular, stochastic simulation has proven to be an invaluable tool for modelling cellular environments containing low numbers of molecules such as proteins and messenger RNA (mRNA) [2]. However, there is one aspect of cellular dynamics that is often overlooked in stochastic models, and that is the effect of spatial constraints on the behaviour of a cell; The behaviour of a cell depends not only on the number of molecules present, but also on the distribution of these molecules and how they move about the cell space and interact with each other. For this reason, we hypothesise that spatial stochastic models of cells may be better at simulating the experimentally observed behaviour of a cell than spatially averaged models. To test this hypothesis, we created and compared spatially dependent models against spatially averaged models for a number of biological processes associated with gene expression.

Gene expression is the process by which genes are transcribed into mRNA molecules which are in turn translated into proteins [3]. Gene regulatory networks consist of groups of genes whose function can be positively or negatively affected through various signals and/or molecular interactions, which promote or inhibit the production of proteins. Gene expression is inherently stochastic: low numbers of genes, proteins and mRNAs are found within many cells, and many genes are controlled by probabilistic interactions between these molecules [4]. Therefore, stochastic modelling is an appropriate technique for studying this biological process such as gene regulation [1].

One type of gene regulation that we study is that of post-transcriptional regulation. Post-transcriptional regulation is a process that occurs between the transcription and translation of a gene. In particular, we focus on the post-transcriptional control of the gene *PTEN*, which acts as a tumour suppressor in cells. As shown in Figure 1(a), *PTEN* acts as a tumour suppressor by inhibiting the production of Akt, a protein kinase that promotes cell proliferation and survival, by the enzyme PI3K. In doing so, *PTEN* promotes programmed cell death (or apoptosis) which can signal for the destruction of mutated or cancerous cells before they can proliferate to form a tumour. Therefore, normal *PTEN* expression is extremely important for preventing tumour development.

PTEN is regulated by small RNAs known as microRNAs which bind to the *PTEN* mRNA and repress its ability to produce protein, thereby limiting its tumour suppressive capabilities. [6]. An assortment of mechanisms can affect regulation of *PTEN* by microRNAs. One type of mechanism is known as "sponge modulation", in which RNAs that share microRNA binding sites with *PTEN* may compete for a shared pool of microRNAs. This group of "competing endogenous RNAs" (ceRNAs), act by sponging up microRNAs, leaving fewer microRNAs to bind to and repress *PTEN* mRNA. This causes an up-regulation of *PTEN* as it can now resume protein production. Another type of mechanism that regulates *PTEN*-microRNA interactions is known as"non-sponge modulation"; Decoy microRNAs ("decoys") block microRNA binding sites on mRNAs. If these decoys block the binding sites of *PTEN* mRNAs, then *PTEN* will be up-regulated as there will be fewer opportuni-

ties for microRNAs to bind to and repress *PTEN*. If the decoys block the binding sites of the ceRNAs, then *PTEN* will be down-regulated as there are fewer ceRNAs to soak up the microRNAs and also fewer decoys to block the *PTEN* mRNA binding sites.

Many models of gene regulatory networks do not reflect spatial effects and make the

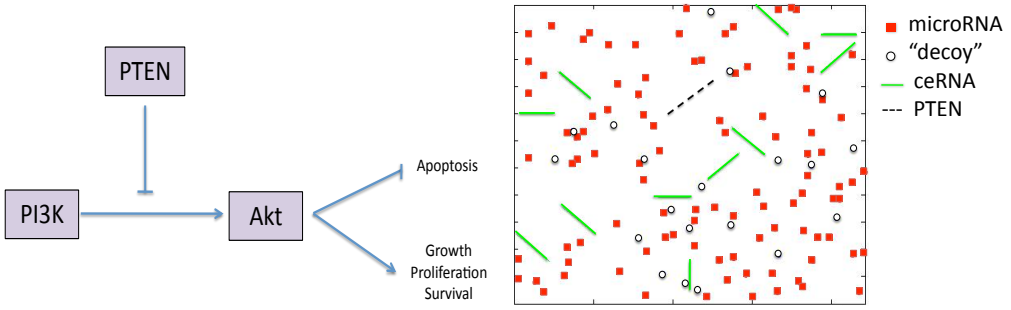


Fig. 1. Figure (a) shows a schematic of the influence of *PTEN* on the PI3K-Akt pathway. When *PTEN* levels are low, Akt production can occur via this pathway, leading to an increase in cell growth and proliferation and a greater likelihood of tumour formation. Figure (b) shows a screenshot of the cellular automaton model we created.

assumption that the system is well mixed, i.e. that the system reactions occur much more slowly than the time it takes for a molecule to diffuse across a cell [5]. In practice, however, processes such as target binding by a microRNA may be spatially dependent; a microRNA may find its target site faster, or slower, depending on the initial distance from the target and the mechanism by which it moves through the cell. We therefore create a spatial stochastic model of the system to examine the effect of spatial dynamics on the behaviour of a cell. We use a Monte Carlo model as a basis for our model and created a two-dimensional cellular automaton (CA) model (Figure 1(b)) to incorporate spatial effects. Our initial analysis suggests that spatial modelling of these gene regulatory processes does produce different results to those obtained from the spatially-averaged models.

References

1. Andrews, S., et al. Stochastic models of biological processes. *Encyclopaedia of Complexity and System Science 9* (2009), 8730–8749.
2. Burrage, K., et al. Stochastic simulation for spatial modelling of dynamic process in a living cell. In *Design and Analysis of Biomolecular Circuits*, Koeppl, Setti, di Bernardo, and Densmore, Eds. Springer New York, New York, NY, 2011, pp. 43–62.
3. El Samad, H., et al. Stochastic modelling of gene regulatory networks. *International Journal of Robust and Nonlinear Control 15*, 15 (Oct. 2005), 691–711.
4. Paulsson, J. Models of stochastic gene expression. *Physics of life reviews 2*, 2 (2005), 157–175.
5. Raser, J. M., and E. K., O. Noise in gene expression: origins, consequences, and control. *Science (New York, N.Y.) 309*, 5743 (Sept. 2005), 2010–3.
6. Sumazin, P., et al. An Extensive MicroRNA-Mediated Network of RNA-RNA Interactions Regulates Established Oncogenic Pathways in Glioblastoma. *Cell 147*, 2 (Oct. 2011), 370–381.

Extending Logic Programming for Life Sciences Applications

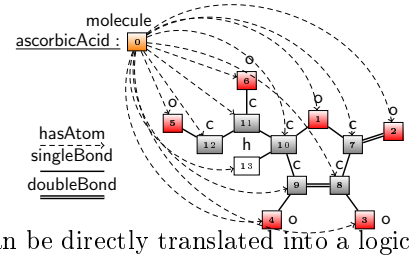
Despoina Magka

Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

The volume of bioinformatics data produced by research laboratories worldwide is increasing at an astonishing rate turning the need to adequately catalogue, represent and index the vast amounts of produced data sources into a pressing challenge. Semantic Web technologies have achieved significant progress towards the federation of biochemical information via the definition and use of domain vocabularies with formal semantics, also known as *ontologies*. OWL [4], a family of logic-based knowledge representation formalisms, has played a pivotal role in the advent of Semantic Web due to its significant ability to reason over ontologies by means of logical inference. A core reasoning task in biochemical domains is classification; with the help of performant OWL reasoners life scientists can employ OWL to drive fast, automatic and repeatable classification processes. As a consequence, OWL bio- and chemo-ontologies are widely used for the modelling of life sciences knowledge.

A prominent Semantic Web ontology is ChEBI, an open-access dictionary that provides taxonomical information for molecular entities with numerous applications such as drug discovery and study of disease pathways. Inspite of ChEBI being available in OWL format, classifying ChEBI with an OWL reasoner produces only few of the desired subsumptions because OWL is not expressive enough to capture the structure and the properties of chemical entities. For instance, the tree-model property of OWL prevents one from faithfully representing *cyclic* molecules: OWL axioms can state that butane molecules have four carbon atoms, but they cannot state that the four atoms in a cyclobutane molecule are arranged in a *ring*. Additionally, due to the first-order logic semantics of OWL it is difficult to describe classes based on the *absence* of certain characteristics: if an inorganic molecule is defined as 'a molecule *not* containing a carbon atom', then in order to derive that water is inorganic one needs to also specify that water contains at most three atoms and that both hydrogens and oxygens are disjoint from carbons. As a consequence of these inadequacies, ChEBI is manually curated by human experts who determine the chemical classes of new molecular entries. Currently, ChEBI describes nearly 30,000 entities and grows at a rate of 3,500 entities per year. Given the size of other publicly available chemical databases with millions of molecules, there is clearly a strong potential for ChEBI to expand by speeding up the curating tasks through automation of chemical classification.

Towards that direction, we developed a theoretical and practical framework for the representation of graph-shaped objects, with a particular application to chemistry. Namely, we introduced Description Graph Logic Programs (DGLP) [6] a logic-based language that addresses the previously outlined expressivity limitations of OWL and admits a sound, complete and terminating reasoning procedure for the automatic classification of structured objects. The main modelling artefact of our formalism is the *description graph*, a directed labeled graph that abstracts out the structure of objects and can be directly translated into a logic program. E.g., the description graph of the ascorbic acid molecule is depicted above.



From a technical point of view, DGLP is a language that is based on logic programming rules with function symbols in the head and adopts the stable model semantics. Thus, DGLP is directly related to extensions of datalog with existential rules, a formalism that has been extensively studied in various areas such as theory of databases, knowledge representation and answer set programming [1]. Existential rules may incur non-termination of the reasoning procedure due to the creation of fresh terms that can be infinitely many; so, the main focus of research is to guarantee termination by formulating suitable restrictions, the so-called *acyclicity* conditions. In order to ensure decidability of our formalism we proposed a number of novel acyclicity criteria, explored their computational properties and proved that they are strictly more general than previously suggested analogous conditions; reasoning over acyclic DGLPs was shown to be 2EXPTIME-complete [2].

In terms of applicability, we implemented LoPStER, a prototype that performs logic-based chemical classification and draws upon DLV [5], a state of the art logic programming reasoner. In order to assess the feasibility of our approach, we empirically evaluated our implementation using data extracted from the ChEBI ontology. Our software classified 500 molecules under 51 chemical classes in 40 secs, which exhibits a significant improvement in comparison with our previous (450 seconds to classify 70 molecules [6]) and related (Hastings et al. report a total of 4 hours to compute the superclasses of 140 molecules [3]) work.

Furthermore, while conducting the experiments we discovered a number of missing and inconsistent axioms from the manually curated ChEBI ontology. As one can infer from the molecular structure of ascorbic acid, ascorbic acid is a carboxylic ester (i.e. a molecule containing (C=O)O) as well as a polyatomic cyclic entity. Inspite of the fact that these superclasses were exposed by our classification methodology, we were not able to identify them in the ChEBI hierarchy. Moreover, ascorbic acid is asserted as a carboxylic acid (i.e. a molecule with a carboxy group, which has formula C(=O)OH) which is not the case as it can be deduced by the lack of a carboxy group in its molecular graph. We interpret the revealing of these discrepancies as an indication of the practical relevance of our contribution.

Concerning future benefits, our prototype could form the basis of an application to assist biocurators towards a more rapid development of the ChEBI ontology. From a modelling point of view, our approach could stimulate the adoption of a different and expressive reasoning paradigm based on logic programming for which highly optimised reasoners are available. For the future, we plan to design a surface syntax that will enable life scientists to represent knowledge without the need to script intricate logic programs; from a theoretical point of view, it would be interesting to investigate extensions of DGLP with numerical values that would allow for more expressive modelling such as molecular weights.

References

1. Calì, A., Gottlob, G., Lukasiewicz, T., Marnette, B., and Pieris, A. Datalog+/-: A family of logical knowledge representation and query languages for new applications. In *LICS'10*.
2. Cuenca Grau, B., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., and Wang, Z. Acyclicity Conditions and their Application to Query Answering in Description Logics. In *KR 2012* (2012), AAAI Press.
3. Hastings, J., Dumontier, M., Hull, D., Horridge, M., Steinbeck, C., Stevens, R., Sattler, U., Hörne, T., and Britz, K. Representing Chemicals Using OWL, Description Graphs and Rules. In *OWLED* (2010), vol. 614.
4. Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F. From SHIQ and RDF to OWL: the making of a Web Ontology Language. *J. Web Sem. 1*, 1 (2003), 7–26.
5. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., and Scarcello, F. The DLV system for knowledge representation and reasoning. *ACM TOCL 7*, 3 (2006).
6. Magka, D., Motik, B., and Horrocks, I. Modelling Structured Domains Using Description Graphs and Logic Programming. In *ESWC* (2012), Springer, pp. 330–344.

A Hierarchical Word Alignment Model based on Pitman-Yor Processes

Alex Wilson

Department of Computer Science, University of Oxford

When presented with a pair of sentences in different languages, but with a shared meaning, the *word alignment* of the sentences indicates which words correspond in the two sentences. For example, the alignment between the English sentence "The library is closed" and the Spanish sentence "La biblioteca está cerrada" indicates that 'The' is aligned to 'La', 'library' to 'biblioteca', and so on. These word alignments have a large number of applications; being able to automatically extract the alignment is therefore of great interest. However, this is not an easy problem; alignments are often not as simple as the one-to-one example given above.

The most commonly used models of word alignment are the IBM Models (1-5) [1], and the Hidden Markov Model (HMM) [6]. These are usually trained using the EM algorithm, although recently a Bayesian prior was added to IBM Model 1 (with Gibbs sampling used for inference) [3]; this showed an improvement over the EM implementation of Model 1.

When aligning, the simple IBM Models (1 and 2) and the HMM only take into account the type of the word and the ordering of the words. The higher numbered IBM Models add the concept of word fertility (how many target words each source word aligns with). Given the difficulty of the word alignment problem, it is desirable to modify the models to utilise morphological information about the words, such as part-of-speech (POS) tags, or word stems. The standard models, for example, would not know that nouns are more likely to align with other nouns than they are with verbs, or even punctuation.

To utilise morphological information for the purpose of improving alignments, hierarchical versions of the simple IBM alignment models (IBM Models 1 and 2) and the Hidden Markov Model (HMM) were created, based on hierarchical Pitman-Yor processes (PYP). PYPs are nonparametric Bayesian models, and are a generalisation of the Dirichlet process, with parameters a, b and a base distribution ϕ. The lexical parameters, T, of each model (the probabilities of a particular source word aligning to a particular target word) had PYP priors placed upon them (as described in [3]); the hierarchical model was then constructed by placing PYP priors on the base distributions. It should be noted that Pitman-Yor processes have a power-law distribution similar to the distribution of words in natural language, so they are a good fit for language-based models. The hierarchy can be extended indefinitely by adding further PYP priors to each base distribution.

The model was tested by training it on two large parallel corpora: the French-English Hansards corpus [2] and the Japanese-English Kyoto Free Translation Task corpus [4]. Hand-aligned evaluation sets were obtained for both corpora; after a set number of iterations of the Gibbs sampling, the output was compared to the evaluation alignment. The results were evaluated using the same metrics suggested in [5]: recall, precision and the alignment error rate (AER).

Three sources of hierarchical data were used in separate tests: part-of-speech tags, word stems, and WordNet synsets (sets of synonyms for each word), in two-level hierarchies (where each word is paired with some hierarchical tag, e.g. word+stem). In addition, part-of-speech tags and word stems were used in a three level hierarchy. As a baseline, the same corpora and training schemes were used to train the original non-hierarchical models (using the Bayesian formulation with Gibbs sampling).

With the exception of the synsets, the addition of hierarchical data improved upon the baseline. It is thought that the reason behind the poor results of the synset data is due to the simplistic sense disambiguation performed to generate the data. The best results were gained by the hierarchical HMM using both part-of-speech tags and stems (due to a lack of a Japanese stemmer, this was only performed on the Hansards data); the results for this experiment appear in Table 1.

	Baseline			Hierarchical		
Training Scheme	Recall	Precision	AER	Recall	Precision	AER
IBM Model 1	80.09	66.11	29.03	82.39	67.90	27.08
IBM Model 2	89.30	79.88	16.78	89.33	80.43	16.42
HMM	92.89	81.96	14.26	93.51	82.86	13.46

Table 1. Results for the Hansards data with a POS tags and stems

In future work, hierarchical versions of the higher numbered IBM alignment models could be constructed. Hierarchical PYPs could also be added to the other parameters in the models. In addition, further attempts to utilise synsets or other semantic information can be performed using these hierarchical models.

References

1. Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist. 19*, 2 (June 1993), 263–311.
2. Germann, U. Aligned hansards of the 36th parliament of canada. `http://www.isi.edu/natural-_language/download/hansard/index.html`, 2011.
3. Mermer, C., and Saraçlar, M. Bayesian word alignment for statistical machine translation. *Proceedings of ACL HLT* (2011).
4. Neubig, G. The Kyoto free translation task. http://www.phontron.com/kftt, 2011.
5. Och, F., and Ney, H. A systematic comparison of various statistical alignment models. *Computational linguistics 29*, 1 (2003), 19–51.
6. Vogel, S., Ney, H., and Tillmann, C. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics - Volume 2* (Stroudsburg, PA, USA, 1996), COLING '96, Association for Computational Linguistics, pp. 836–841.

Orderly Algorithm for Fast Enumeration of String Graphs

for use in Graphical Conjecture Synthesis

David Quick

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

The setting for this paper is given by the categorical quantum mechanics developed in [1] and the various graphical languages based on commutative frobenius algebras developed following that paper. The ZX-calculus [2] and GHZ/W-calculus [3] are particularly popular examples in quantum computation and have been used in proofs in a variety of areas.

The program Quantomatic [6] was designed to work with structures such as the ZX-calculus and was then extended to allow much more general graphical theories to be input. QuantoCosy [5] was then built on top of Quantomatic as a graphical conjecture synthesis program. This program takes a set of generators for a graphical language (each with a fixed numbers of inputs and outputs) along with a valuation on them. It then looks for equalities between graphs built from the generators and returns a list of graphical identities. Note that our graphs are directed and are allowed to have edges which are free at one end (these represent inputs and outputs). We also allow a typing on the vertices, usually represented by the colour of the vertex.

Part of this procedure is to be able to create a list of all graphs up to a certain size (number of vertices). This paper develops an efficient algorithm for production of this catalogue of graphs without duplicates. We do so by choosing a clever representation of graphs which leads to easy listing of graphs (possibly with duplicates). We then determine a way to decide whether a representation of a graph is the 'canonical' one (defined below), in which case we add it to our catalogue. This is a generalisation of the method from [4] in which untyped, undirected, simple graphs are catalogued.

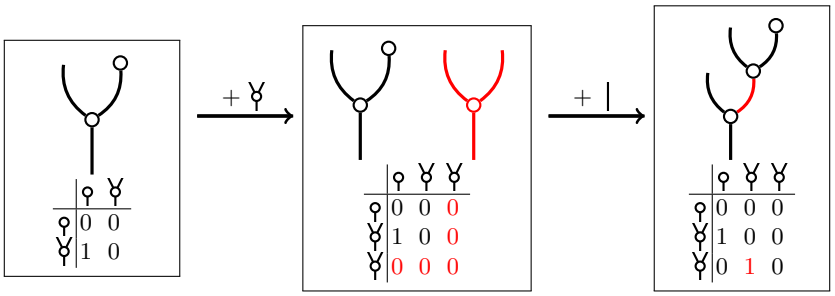


Fig. 1: Example of Vertex and Edge Addition

We build graphs from the empty graph by adding vertices and edges one at a time. Figure 1 demonstrates addition of a vertex and an edge to a graph (red parts are the additions). The adjancency matrices below the graphs store the number of edges from the ith vertex to the jth vertex as the (i, j)th entry. We represent our graphs by having a fixed order in which to write the entries of the adjacency matrix as a vector. The order we choose is to

take (for a_{ij} the (i,j)th entry of the adjacency matrix) the entry a_{11} first (edges on first vertex) then a_{12}, a_{22}, a_{21} (edges on first two vertices) and continue writing them out in this fashion ($a_{13}, a_{23}, a_{33}, a_{32}, a_{31}, a_{14}$...). For example the matrices in the diagram above can be represented as:

$(\{\circ, \circ\!\!-\!\!\circ\}, (0,0,0,1))$, $(\{\circ, \circ\!\!-\!\!\circ, \circ\!\!-\!\!\circ\!\!-\!\!\circ\}, (0,0,0,1,0,0,0,0,0))$, $(\{\circ, \circ\!\!-\!\!\circ, \circ\!\!-\!\!\circ\!\!-\!\!\circ\}, (0,0,0,1,0,0,0,1,0))$

We only need to consider adding new vertices to the end of our vertex list and adding new edges on or to the right of the last non-zero entry of the edge vector. This will still produce every possible graph by correctly adding vertices and edges in order.

Representing graphs like this leads to a simple way to make sure never to add two isomorphic graphs to the catalogue. We say a representation of a graph in the form above is canonical if it's vector of edges is maximal (lexicographically). Then every time we produce a new representation of a graph we can check for canonicity. Those which are found to be canonical are added to our catalogue and we can continue to build larger graphs from them. Those which are not canonical do not need to be catalogued and no graph built up from them needs to be added to the catalogue either so we can forget about them.

The full paper gives a proof that every non-empty canonical graph representation is found either by an edge or vertex augmentation of another canonical graph representation. Hence we can build up a catalogue of canonical representations of graphs starting from the empty graph by iteratively performing every possible vertex augmentation and then any number of edge augmentations.

An added advantage of cataloguing graphs from smaller ones in the area of Conjecture Synthesis is that we do not care about cataloguing graphs which are built up from reducible graphs. Hence every time a new graph is produced we can check if it can be reduced and if so add the reduction to our ruleset and then drop the graph from our process.

References

1. Abramsky, S., and Coecke, B. A categorical semantics of quantum protocols. *Proceedings of the 19th IEEE conference on Logic in Computer Science (LiCS'04)* (February 2004).
2. Coecke, B., and Duncan, R. Interacting quantum observables: Categorical algebra and diagrammatics. *New J. Phys. 13* (June 2009), 043016.
3. Coecke, B., and Kissinger, A. The compositional structure of multipartite quantum entanglement.
4. Colbourn, C., and Read, D. Orderly algorithms for generating restricted classes of graphs. *Journal of Graph Theory, Vol. 3 (1979) 187-195* (December 1979).
5. Kissinger, A. Synthesising graphical theories.
6. Kissinger, A., Merry, A., Dixon, L., and Duncan, R. Quantomatic, 2009.