5TH INTERNATIONAL CONFERENCE

Logic and Applications

LAP 2016

September 19 - 23, 2016
Dubrovnik, Croatia

Book of Abstracts

Course directors:

- Zvonimir Šikić, University of Zagreb
- Andre Scedrov, University of Pennsylvania
- Silvia Ghilezan, University of Novi Sad
- Zoran Ognjanović, Mathematical Institute of SASA, Belgrade
- Thomas Studer, University of Bern

Book of Abstracts of the 5th International Conference on Logic and Applications - LAP 2016, held at the Inter University Center Dubrovnik, Croatia, September 19 - 23, 2016.

LaTeX Typesetting and Printing:

- Jelena Ivetić, University of Novi Sad, Serbia
- Marcel Maretić, University of Zagreb, Croatia
- Dušan Gajić, University of Novi Sad, Serbia

LAP 2016 Webiste: `http://imft.ftn.uns.ac.rs/math/cms/LAP2016`
Maintained by Nenad Savić, University of Novi Sad, Serbia

Contents

A Non-Monotonic Logic for Distributed Access Control

Diego Agustín Ambrossio[1] and Marcos Cramer[1]

[1]University of Luxembourg, Luxembourg

Keywords:

non-monotonic logic, access control, autoepistemic logic

1 Motivation and Aims

Multiple logics have been proposed for distributed access control, most of which use a modality *k says* indexed by a principal k [1, 2].Previously proposed *says*-based access control logics are monotonic, i.e. adding new statements cannot lead to a previously accepted access request to get rejected. This, however, makes it impossible to straightforwardly model access denials in such logics. We propose *Distributed Access Control Logic* (*D-ACL*), a non-monotonic *says*-based access-control logic based on an extension of autoepistemic logic to the multi-agent case.

We define a query-driven decision procedure for D-ACL, which – under the assumption of a finite domain – allows to determine access rights while minimizing the information flow between principals increasing privacy.

2 D-ACL Syntax and Semantics

We assume familiarity with first-order logic.

D-ACL Syntax: D-ACL formulas are defined by the following EBNF rule, where t denotes an arbitrary term and x and arbitrary variable:

$$\varphi ::= P(t, \ldots, t) \mid t = t \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\ \varphi \mid t\ \ says\ \ \varphi$$

The intuitive reading of $t\ says\ \phi$ is "t supports ϕ". [1]

Definition 1 *A D-ACL* theory *is a set that consists of D-ACL formulas.*

[1]If the term t does not denote a principal, $t\ says\ \phi$ will be interpreted to be false.

Different principals can issue statements that become part of the access control policy. A D-ACL theory as defined above only represent statements issued by a single principal. To represent the full access control policy, we use the notion of a *distributed theory*.

Definition 2 *A* distributed theory $\mathbb{T}$ *is an indexed family* $(\mathbb{T}_A)_{A \in \mathcal{A}}$*, where each* $\mathbb{T}_A$ *is a D-ACL theory.*

D-ACL Semantics: Van Hertum et al. [3] have defined various semantics for D-ACL using Approximation Fixpoint Theory [4], but have argued for the use of the well-founded semantics in the application of D-ACL to access control. We define a decision procedure for the well-founded variant of D-ACL. We refer the reader to [3] for the definition of the well-founded semantics of D-ACL.

3 Decision Procedure

We define a query-driven decision procedure for D-ACL. It allows to determine access rights while minimizing the information flow between principals.

A query in the form of a D-ACL formula ϕ is posed to a principal A. A determines whether her theory contains enough information to verify ϕ. It can happen that A cannot verify ϕ just on the basis of her theory. For example, A's theory may contain the formula $B \; says \; p \rightarrow \phi$. In this case, A can forward a remote subquery to B concerning the status of p in B's theory. If B verifies the subquery p and informs A about this, A can complete her verification of the original query ϕ.

The decision procedureis composed of two modules. The *Query Minimization Procedure (QMP)* and the *Communication Procedure (COMM)*. QMP determines minimal sets of remote calls to other theories that could verify a query. COMM handles communication between principals, and loops that may occur. COMM works by dynamically producing a graph representing the queries and attaching (three-valued) truth values to the vertices in it. Thus, detecting and handling loops.

Query Minimization: First we translate D-ACL theories to first-order theories. We replace D-ACL formulae of the type $A \; says \; \varphi$ by new propositional variables $p^{+}_{A\_\texttt{says}\_\phi}$ or $p^{-}_{A\_\texttt{says}\_\phi}$ depending wether the formula appears in a positive or negative context.[2]

Definition 3 *Let* T *be a D-ACL theory.* $\tau(T)$ *is constructed by replacing every says-atom* $A \; says \; \varphi$ *occurring in* T*: (i) by* $p^{+}_{A\_\texttt{says}\_\varphi}$ *every* positive *occurrence says-atom; (ii) by* $p^{-}_{A\_\texttt{says}\_\varphi}$ *every* negative *occurrence says-atom.*

[2] The propositional variable $p^{+}_{A\_\texttt{says}\_\varphi}$ represents the upper bound for the truth value of $A \; says \; \varphi$ and $p^{-}_{A\_\texttt{says}\_\varphi}$ the lower bound.

We define $min\_incons(\mathcal{T}, S)$ to be the set of minimal partial structures $S' \subseteq S$ such that S' is not a partial model of $\mathcal{T}$.

Definition 4 *We define $\mathbb{S}$ to be the set containing every partial structure S such that for every says-atom $A\,says\,\varphi \in T$, either $S \models \neg p^{+}_{A\_says\_\varphi}$ or $S \models p^{-}_{A\_says\_\phi}$, but not both.*

Given a theory $\mathcal{T}$ and a query α, the QMP returns a set of sets of modal atoms necessary to be resolved by querying other theories.

h!t] Query Minimization Procedure
Input: theory T, D-ACL query α
Output: set $\mathbb{L}$ of sets of modal atoms
1: $\mathbb{L} := \emptyset$
2: $\mathcal{T} := \tau(T \cup \{\{\neg\alpha\}\})$
3: **for each** $S \in \mathbb{S}_T$ **do**
4: **if** S is not a partial model of $\mathcal{T} \cup \{\neg\alpha\}$ **then**
5: pick a partial structure S_{min} from $min\_incons(\mathcal{T} \cup \{\neg\alpha\}, S)$
6: $\mathbb{L} := \mathbb{L} \cup \{L^{S_{min}}\}$
7: **return** $\mathbb{L}$

end

4 Conclusions and Future Work

Distributed Access Control Logic (D-ACL) is a non-monotonic *says*-based access control logic. The non-monotonicity of D-ACL makes it possible to model access denials more straightforwardly than in state-of-the-art access control logics. We have defined a query-based decision procedure for D-ACL, which minimizes information flow between principals. The decision procedure corresponds with the proposed semantics.

References

[1] M. Abadi, *Logic in Access Control*, Proceedings of the Eighteenth Annual IEEE Symposium on Logic in Computer Science, 2003, pp. 228–233.

[2] D. Garg, F. Pfenning, *Stateful Authorization Logic – Proof Theory and a Case Study*, JCS, vol. 20, no. 4, pp. 353–391, 2012.

[3] P. Van Hertum, M. Cramer, B. Bogaerts, M. Denecker, *Distributed Autoepistemic Logic and its Application to Access Control*, IJCAI 2016.

[4] M. Denecker, V. Marek, M. Truszczyński, *Fixpoint 3-valued semantics for autoepistemic logic*, in AAAI'98. pp. 840–845.

Experimental Evaluation of Time-Memory Trade-Off Attack

Aleksandra Arsić[1] and Aleksandra Zdravković[1]

[1]Mathematical Institute SASA, Belgrade, Serbia

Keywords:

encryption, security evaluation, Time-Memory-Data Tradeoff attack, Trivium

This paper addresses certain issues of a generic approach for inversion of an one-way function which is of interest in different domains of mathematics including mathematical logic and its applications, and particularly regarding security evaluation of cryptographic algorithms. There are the following two straightforward approach for recovering the argument given the corresponding image generated by one-way function where the inversion is a hard problem: (i) exhaustive search over all possible arguments; (ii) employing a code-book with all possible argument-image pairs. The main problem with the both approaches is the exponential complexity of implementation. Helman [1] has proposed a technique which reduces the required time and memory complexities. Using pre-computation time of N, Hellman showed that the online time T and memory M satisfy the relation $TM^2 = N^2$, where $N = 2^n$. Consequently, the attack is called a time/memory trade-off (TMTO) algorithm. This attack on stream ciphers is a serious security threat (see [2], [3], [4], for example) and the resistance to this class of attacks is an important criterion in the design of a modern stream cipher. This paper provides certain experimental evaluation of the considered TMTO paradigm. Main goal is to provide illustrative experimental evidences on a particular quantitative feature of two TMTO design approaches. For one-way function, authors choose Trivium cipher. Trivium [5] is stream cipher designed to generate keystream from 80-bit secret key and an 80-bit initial vector (IV). Process of generating keystream bits consists of two phases. First phase aims to initialize key and IV into 288-bit initial state s. Next phase is generating keystream vector z. In this phase 15 specific bits are used for updating 3 bits of the state s and to compute 1 bit of z. The state register is then rotated and the process repeats itself until complete keysteam vector has been generated.

First phase for TMTO attack is preprocessing phase. Goal of preprocessing phase is matrix initialization.

According to computation power and technical opportunities, our matrix and the number of all possible solutions for key vector are not the same. We try to guess only 20 bits and the other 60 bits of key and 80 bits of IV are initialized by random values and stay fixed all time. In that way, number of all permutations is smaller, so time and memory space for computing matrix are smaller, too.

Preprocessing phase is described as:

Form a $m \times t$ matrix that tries to cover the whole search space which is composed of all the possible permutations with guessed 20 bits of key vector as follows:

1. Randomly generate m startpoints of the chains, each point is represented like vector of 20 bits length.
2. Make it the next point in the chain which is the output from Trivium function and update the s register with this point.
3. Iterate Step (2) t times on each startpoint respectively.
4. Store the pairs of startpoints and endpoints (SP_j, EP_j), $j = 1, ..., m$ in the matrix.

Our first experiment was to generate matrix for TMTO Attack and check if there are duplicate states. Dimensions of our matrix were $m \times t$ where $m = 2^{15}$ and $t = 2^5$. First, startpoints for each row was random initialized, keeping in the mind there are no duplicate startpoints. Next step was to fill all states in matrix. Every state in chains, except first one, is result from 20 iterations of Trivium algorithm with initialization of the register with previous state's bits from the same row in matrix. After filling the matrix, content of matrix was analyzed.

Experiments were repeated 50 times. Experiments showed that some states in matrix occur more than one time. The conclusion is the matrix does not contain all the elements of search space. Average repetition rate in all experiments was in range from 57.14 to 64.43 percents of number of all matrix states.

Our next experiment consists of constructing several tables. To construct each table, the attacker chooses t random vectors, one for each table. Described matrices have the same number of columns like first matrix, but number of rows is less than the number of rows in the single matrix. In that case, numbers of rows in all tables are equal and total number corresponds to the number of rows in the single matrix. Algorithm for matrix filling is the same, but every time when Trivium function gives us a new state vector as output, we translate it for random vector corresponding to table in which we put it. For translation we use XOR operation on its state. These tables can be generated in parallel, but statistical analysis follows the merging of

all tables in single table.

Authors did some experiments when single matrix was divided on 2, 4 and 8 smaller tables. Experiments shown that in case when we generate two tables with two random vectors, repetition rate was in range from 54.71 to 61.57 percents of size of search space. If we construct eight tables for search space and assign eight random translation vectors for each one, repetition rate will be in range from 35.19 to 41.62 percents of size of search space.

Results shown that in case of using more small tables, the total number of distinct points which cover search space is bigger. This technique reduces the number of collisions in the table, and hence allows to cover most of the points by a single table. The larger the table is, the higher is the probability that a new chain has an intersection with previous chains [6]. Each state's repetition reduces the number of distinct keys which are actually covered by a table. The efficiency of a single table rapidly decreases with its size. To obtain a high probability of success it is better to generate multiple tables using a different reduction function for each table. In our case, as reduction function XOR operator is used. Chains of different tables can have intersection. Using more reduction functions which are applied in different tables leads to smaller number of intersections.

This paper has provided certain, particular, quantitative insights regarding design of the tables which employs TMTO approach for inverting of an one-way function.

References

[1] M. Hellman, "A cryptanalytic time-memory tradeoff," *IEEE, Transactions on Information Theory*, pp. 401 – 406, 1980.

[2] O. Dunkelman and N. Keller, "Treatment of the initial value in time-memory-data tradeoff attacks on stream ciphers," *Information Processing Letters*, no. 107, pp. 133 – 137, 2008.

[3] M. Mihaljevic, S. Gangopadhyay, G. Paul, and H. Imai, "State recovery of grain-v1 employing normality order of the filter function," *IET Information Security*, vol. 6, no. 2, pp. 55 – 64, 2012.

[4] M. Mihaljevic, S. Gangopadhyay, G. Paul, and H. Imai, "Internal state recovery of keystream generator lili-128 based on a novel weakness of the employed boolean function," *Information Processing Letters*, vol. 112, no. 21, pp. 805 – 810, 2012.

[5] C. D. Canniere and B. Preneel, "Trivium specifications," *ECRYPT*.

[6] P. Oechslin, "Making a faster cryptanalytic time-memory trade-off," *Advances in Cryptology - CRYPTO 2003*, pp. 617 – 630, 2003.

Strategic reasoning in conversations under imperfect information

Nicholas Asher and Soumya Paul

Institute de Recherche en Informatique de Toulouse
118 Route de Narbonne, 31062 Toulouse, France

Keywords:

strategic conversations, infinite games, type spaces

Finite and infinite sequential games are important to theoretical computer science in studying program synthesis and verification and also to topology and set-theory. Epistemic game theory is used to study rational behaviour and equilibrium outcomes in strategic situations. This paper shows how the combination of the two can help shape linguistic intuitions, and why linguistic considerations call for important modifications in standard conceptions of sequential games. The general structure of a conversation is that the participants (players) alternate exchanging messages until the conversation is finished. The players' conversational goals constitute the winning condition of the games. If the goals of the players are opposed, the players naturally resort to strategic reasoning to determine what to say and when. Sequential finite and infinite games are hence a natural framework for the analysis of strategic conversation [3, 2].

Example 1. Consider the following excerpt from a courtroom proceedings where a prosecutor is querying the defendant.

(a) **Prosecutor**: *Do you have any bank accounts in Swiss banks, Mr. Bronston?*
(b) **Bronston**: *No, sir.*
(c) **Prosecutor**: *Have you ever?*
(d) **Bronston**: *The company had an account there for about six months, in Zurich.*
(e) **Prosecutor**: *Thank you Mr. Bronston.*

One conversational goal of the Prosecutor in (1) is to get Bronston to commit to an answer eventually (and admit to an incriminating fact) or to continue to refuse to answer (in which case he will be charged with contempt of court). Under such a situation, the response (1d) of Bronston is clearly a

clever strategic move. His aim is to avoid being in either of the above situations. Such examples abound in real-life – in political debates, courtroom proceedings, interviews, or even bargaining negotiations.

It is thus natural to model such strategic conversations as games. Attempts have been made in the literature to use the framework of signaling games [5]. However, when the preferences of the players are strictly opposed, as is the case with strategic conversations, signaling games are not suitable as they predict no communication at all in equilibrium [4]. Another crucial characteristic of such conversations is that they do not have a set end. The players while engaging in a conversation do not know when it will end, and even if it does, whether a player will be able to achieve his/her objective. [3, 2] thus proposes to model strategic conversations as infinite games over a countable 'vocabulary'. The elements of the vocabulary are taken to be SDRSs or discourse representations of SDRT [1], a well established theory for discourse interpretation.

To view conversations as infinite games, one must take into consideration characteristics which are unique to them: (i) The turn structure of the game is crucial in language games, because it is important who says what. This affects the (existence of) winning strategies for the players. (ii) A 'move' by a player in a linguistic game typically carries more semantic content than usually assumed in game theory - implicatures, ambiguity, coherence, consistency etc. (iii) Conversations may have a 'Jury' who evaluates the conversation after it has ended and determines if one or more of the players have reached their goals – determines the winner. For example, in a courtroom situation there is a physical Jury who gives the verdict whereas in a political debate, the Jury is the audience or the citizenry in general. This means that the winning conditions of the players depend on what they believe that the Jury expects them to achieve. (iv) Epistemic elements thus naturally creep into such games. In particular, the players and the Jury have 'types' and also 'beliefs' about the types of the other players and that of the Jury.

Taking into consideration the above characteristics of strategic conversations, [3] models them as infinite games of imperfect information, which they call *message-exchange games* (ME games), over a countable vocabulary V where V is the set of SDRSs [1]. The players 0 and 1 take turns in playing finite sequences of stings from V. The game may potentially go on forever. The set of plays Plays is thus $(V_0^+ \cup V_1^+)^\omega$ where for each $i \in \{0,1\}$, $V_i = V \times \{i\}$. In addition, there is a 3rd player called the Jury who determines the winning conditions (goals) $\mathsf{Win_i}$ for each player i which is a subset of $(V_0^+ \cup V_1^+)^\omega$. Player i wins a play ρ of $\mathcal{G}$ iff $\rho \in \mathsf{Win_i}$.

The modeling of strategic conversations as sequential infinite games [3, 2] is justified by the uncertainty that the players have about the Jury winning conditions $\mathsf{Win_1}$ and $\mathsf{Win_2}$. In this paper, we capture this uncertainty by using the well-established theory of type-structures [6]. We assume that each player $i \in (\{0,1\} \cup \{\mathsf{Jury}\})$ has a (possibly infinite) set of types T_i. With each

type t_i of Player i is associated a (first-order) belief function $\beta_i(t_i)$ which assigns to t_i a probability distribution over the types of the other players. The higher-order beliefs can be defined in a standard way by iterating the functions β_i. The players take turns in making their moves and after every move, all the players dynamically update their beliefs through Bayesian updates. The notions of 'optimal strategies', 'best response', 'rationality', 'common belief in rationality' etc. can then be defined in the standard way.

In Example 1, Bronston's response (1d) was aimed to 'misdirect' the Jury. He believed that the Jury was of a type that would be convinced by his ambigous response and neither incriminate him nor charge him with perjury. His move was indeed rational, *given his belief* about the Jury type.

Powerful as the above techniques are, one has to exercise caution and define the moves, states and the types of the players carefully. Having too rich a type space can lead to inexistence results as shown in [7], which says that if the space of types is not a separable set then there always exists a game with no equilibrium. In our above ME games, associating the types of a player with his/her possible strategies, we see that the space of types is a set with a large cardinality ($> \aleph_1$) and hence we lose separability.

Conversationalists are aware implicitly of the dangers of such cases and debates have exogenous means of ensuring that there are optimal strategies for the speakers to follow. For instance, in debates there is usually a 'moderator' who ensures that all the participants get a fair chance to speak. More generally, we can restore separability by limiting the set of types.

References

[1] N. Asher and A. Lascarides. *Logics of Conversation*. Cambridge University Press, 2003.

[2] N. Asher and S. Paul. Evaluating conversational success: Weighted message exchange games. In *SEMDIAL 20*, New Jersey, USA, July 2016.

[3] N. Asher, S. Paul, and A. Venant. Message exchange games in strategic conversation. *Journal of Philosophical Logic*, 2015. In press.

[4] V. Crawford and J. Sobel. Strategic information transmission. *Econometrica*, 50(6):1431–1451, 1982.

[5] Michael Franke. *Signal to act: Game theory in pragmatics*. PhD thesis, Universiteit van Amsterdam, 2009.

[6] John C Harsanyi. Games with incomplete information played by "bayesian" players, parts i-iii. *Management science*, 14:159–182, 1967.

[7] Z. Hellman and Y. Levy. Bayesian games with a continuum of states. Technical report, Bar Ilan University, 2013.

Soundness and completeness of a sequent calculus with high probabilities

Marija Boričić[1]

[1]Faculty of Organizational Sciences, University of Belgrade, Jove Ilića 154, 11000 Beograd, Serbia

August 24, 2016

Keywords:

deduction relation, sequent calculus, probability, models

We introduce sequents of the form $\Gamma \vdash^n \Delta$, a generalization of Gentzen's sequents $\Gamma \vdash \Delta$, with the intended meaning that 'the probability of the sequent $\Gamma \vdash \Delta$ belongs to the interval $[1-n\varepsilon, 1]$' for a given small real $\varepsilon > 0$ of the form $\varepsilon = \frac{1}{k}$ for some fixed $k \in \mathbf{N}$ and any $n \in \mathbf{N}$, $n \leq k$, inspired by Suppes' idea (see [7]). In order to infer conclusion of the form $\Gamma \vdash^n \Delta$, from a finite set of hypotheses $\Gamma_1 \vdash^{n_1} \Delta_1, \Gamma_2 \vdash^{n_2} \Delta_2, \ldots, \Gamma_s \vdash^{n_s} \Delta_s$, we define a sequent calculus $\mathbf{LKprob}(\varepsilon)$, which turns out to be very simple and elegant (see [1, 2]). The system $\mathbf{LKprob}(\varepsilon)$ can be considered a probabilistic extension of the classical propositional calculus of sequents $\mathbf{LK}$, analogous to the Hilbert–type classical logic probabilization (see [8, 9]).

The axioms of the system are of the form $A \vdash^0 A$ and $\Gamma \vdash^k \Delta$, for any words Γ and Δ, and any formula A. There are two kinds of inference rules — structural and logical. For instance, the rules treating conjunction are as follows:

$$\frac{\Gamma AB \vdash^n \Delta}{\Gamma A \wedge B \vdash^n \Delta}(\wedge \vdash) \quad \frac{\Gamma \vdash^n A\Delta \quad \Gamma \vdash^m B\Delta}{\Gamma \vdash^{m+n} A \wedge B\Delta}(\vdash \wedge)$$

A model for $\mathbf{LKprob}(\varepsilon)$ is a mapping $p : \mathrm{Seq} \to I \cap [0,1]$ satisfying the following conditions:

(i) $p(A \vdash A) = 1$, for any formula A;

(ii) if $p(AB \vdash) = 1$, then $p(\vdash AB) = p(\vdash A) + p(\vdash B)$, for any formulae A and B;

(iii) if sequents $\Gamma \vdash \Delta$ and $\Pi \vdash \Lambda$ are equivalent in $\mathbf{LK}$, in sense that there are proofs for both sequents $\bigwedge \Gamma \to \bigvee \Delta \vdash \bigwedge \Pi \to \bigvee \Lambda$ and $\bigwedge \Pi \to \bigvee \Lambda \vdash \bigwedge \Gamma \to \bigvee \Delta$ in $\mathbf{LK}$, then $p(\Gamma \vdash \Delta) = p(\Pi \vdash \Lambda)$.

The axioms above roughly correspond to to the Carnap's and Popper's sentence probability axioms (see [3, 4, 5, 6]).

We say that the sequent $\Gamma \vdash^n \Delta$ is satisfied in a model p, i.e. $\models_p \Gamma \vdash^n \Delta$, if and only if $p(\Gamma \vdash \Delta) \geq 1 - n\varepsilon$.

Our system is sound and complete with respect to models just described.

References

[1] M. Boričić, *Suppes–style sequent calculus for probability logic*, Journal of Logic and Computation, (to appear) doi:10.1093/logcom/exv068

[2] M. Boričić, *Suppes–style rules for probabilisty logic*, Logic Colloquium 2015.

[3] R. Carnap, *Logical Foundations of Probability*, University of Chicago Press, Chicago, 1950.

[4] H. Leblanc, B. C. van Fraassen, *On Carnap and Popper probability functions, The Journal of Symbolic Logic*, vol. 44 (1979), pp. 369–373.

[5] H. Leblanc, *Probability functions and their assumption sets — the singulary case*, Journal of Philosophical Logic, vol. 12 (1983), pp. 382–402.

[6] K. R. Popper, *Two autonomous axiom systems for the calculus of probabilities*, The British Journal for the Philosophy of Science, vol. 6 (1955), pp. 51–57, 176, 351.

[7] P. Suppes, *Probabilistic inference and the concept of total evidence*, in J. Hintikka and P. Suppes (eds.), *Aspects of Inductive Inference*, North–Holland, Amsterdam, 1966, pp. 49–55.

[8] Z. Ognjanović, M. Rašković, *A logic with higher order probabilities*, Publications de l'Institut Mathématique, vol. 60 (74) (1996), pp. 1–4.

[9] Z. Ognjanović, M. Rašković, Z. Marković, *Probability logics*, Logic in Computer Science, Zbornik radova 12 (20), Z. Ognjanović (ed.), Mathematical Institute SANU, Belgrade, 2009, pp. 35–111.

Weak cyclic Cat-operads

Pierre-Louis Curien and Jovana Obradović
IRIF, Université Paris Diderot and Inria, France

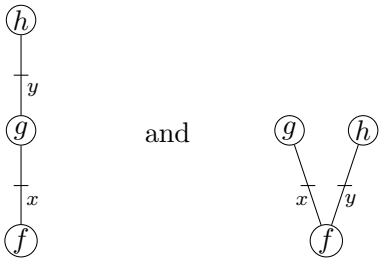
An *operad* is a collection of abstract operations of different arities, equipped with a notion of how to compose them and an action of permuting their inputs. Formally, an operad is given by a functor $\mathcal{O} : \mathbf{Bij}^{op} \to \mathbf{Set}$, where **Bij** is the category of finite sets and bijections and **Set** is the category of sets and functions, and operadic composition morphisms

$$\circ_x : \mathcal{O}(X) \times \mathcal{O}(Y) \to \mathcal{O}(X \backslash \{x\} \cup Y),$$

called *insertions*, defined for all finite sets X and Y and elements $x \in X$ such that $X \backslash \{x\} \cap Y = \emptyset$. An operation $f \in \mathcal{O}(X)$ is to be thought of as a rooted tree whose inputs are labeled by the elements of X, and the composition $f \circ_x g$ as the tree obtained by grafting of the output of g to the input x of f. For non-symmetric and non-unital operads, the axioms of operadic composition come down to the two associativity axioms,

$$(f \circ_x g) \circ_y h = f \circ_x (g \circ_y h) \qquad \text{and} \qquad (f \circ_x g) \circ_y h = (f \circ_y h) \circ_x g,$$

which, informally, say that the two ways of building (by means of grafting) the rooted trees[1]



respectively, are the same. Notice that there is no ambiguity regarding to which one of the associativity rules appplies *with respect to a chosen tree*, despite of the fact that both of them have the same left-hand side.

A *Cat-operad* is an operad that, in addition, has arrows between operadic operations of the same arity. In other words, the operadic operations of the same arity in a Cat-operad do not make just a set, but they are objects of a small category. We have an identity arrow for every operadic operation, and the arrows are closed under composition and insertions $\circ_x$. The notion of *weak Cat-operad* is to the notion of Cat-operad what the notion of bicategory is to the notion

[1]The representations of the two trees are "abbreviations" of the usual graphical notation for trees, in the sense that we did not display their input edges, as well as the output ones.

of 2-category. This means that the equations of operads are replaced by isomorphisms. In particular, the associativity of insertions is now expressed as the existence of isomorphisms

$$\beta_{f,g,h} : (f \circ_x g) \circ_y h \to f \circ_x (g \circ_y h) \qquad \text{and} \qquad \theta_{f,g,h} : (f \circ_x g) \circ_y h \to (f \circ_y h) \circ_x g.$$

By replacing the two associativity equations by isomorphisms in a category, i.e. by passing from Cat-operads to weak Cat-operads, the need to formulate conditions ensuring the coherence of these isomorphisms arises. This coherence is of the same spirit as the coherence of monoidal categories due to Mac Lane: all diagrams made of β- and θ-arrows commute. These conditions are given by Petrić and Došen in [DP15].

The goal of this work is to establish the notion of *weak cyclic Cat-operad*, i.e. a cyclic operad enriched over a category **Cat** of small categories.

Recall that a *cyclic operad* is a generalisation of an operad for which an operation, instead of having inputs and an output, now has "entries", and it can be composed with another operation along any of them. More precisely, a cyclic operad is a functor $\mathcal{C} : \mathbf{Bij}^{op} \to \mathbf{Set}$, together with composition morphisms

$$ {}_x\circ_y : \mathcal{C}(X) \times \mathcal{C}(Y) \to \mathcal{C}(X \backslash \{x\} \cup Y \backslash \{y\}),$$

satisfying certain axioms, where, intuitively, an operation $f \in \mathcal{C}(X)$ should be thought of as an *unrooted tree* whose *leaves* are labeled by the elements of X, and the composition $f_x\circ_y g$ as a tree obtained by grafting the two unrooted trees corresponding to operations f and g along their respective entries x and y. In the non-symmetric and non-unital setting, the axioms of cyclic operads are (equivalently) given as *one* of the two associativity axioms

$$\texttt{(A1)} \quad (f \, {}_x\circ_y \, g) \, {}_u\circ_z \, h = f \, {}_x\circ_y \, (g \, {}_u\circ_z \, h) \qquad \text{and} \qquad \texttt{(A2)} \quad (f \, {}_x\circ_y \, g) \, {}_u\circ_z \, h = (f \, {}_u\circ_z \, h) \, {}_x\circ_y \, g,$$

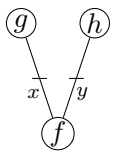
together with the commutativity axiom

$$\texttt{(CO)} \quad f \, {}_x\circ_y \, g = g \, {}_x\circ_y \, f.$$

We shall work with the axioms `(A1)` and `(CO)`, which, after replacing the equations with isomorphisms

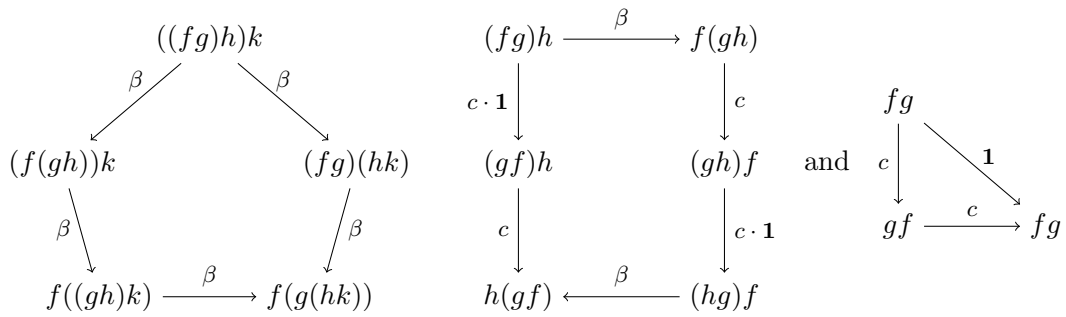
$$\beta^{x,y;u,z}_{f,g,h} : (f \, {}_x\circ_y \, g) \, {}_u\circ_z \, h \to f \, {}_x\circ_y \, (g \, {}_u\circ_z \, h) \qquad \text{and} \qquad c^{x,y}_{f,g} : f \, {}_x\circ_y \, g \to g \, {}_x\circ_y \, f$$

respectively, leads us to a setting whose coherence issues, at first glance, seem like the ones of a symmetric monoidal category. However, as opposed to a symmetric monoidal category, where *all* possible β- and c-arrows exist, in the setting of cyclic operads the *arrows are induced by the shape of the underlying tree* and an arrow $\beta^{x,y;u,z}_{f,g,h}$ does not exist if the underlying tree is



In particular, the commuting hexagon of Mac Lane is "not allowed" in this setting.

In this talk we conjecture that the coherence of weak cyclic Cat-operads is ensured by the commutations of the diagrams



Notice that the hexagon in the middle is *not* the hexagon of Mac Lane.

Our proof consists in adapting the β^{-1}-*normal form argument* of the classical term-rewriting coherence proof for symmetric monoidal categories (see [DR13, Chapter 5]), in such a way that the problem is ultimately also solved by the completeness of the standard presentation of symmetric groups.

We compare the four coherence axioms of weak Cat-operads of Došen and Petrić with our axioms. For this we use the "abbreviation"

$$\theta^{x,y;u,z}_{f,g,h} : (f\ {}_x\circ_y\ g)\ {}_u\circ_z\ h \to (f\ {}_u\circ_z\ h)\ {}_x\circ_y\ g$$

defined as

$$\theta^{x,y;u,z}_{f,g,h} = c_{g,f\ {}_u\circ_z\ h} \circ \beta^{y,x;u,z}_{g,f,h} \circ (c_{f,g} \cdot \mathbf{1}_h).$$

We also discuss the difficulties that occur if one decided to chose the associativity axiom `(A2)` instead of `(A1)` for the starting definition of cyclic operads.

References and Notes

[DR13] K. Došen, Z Petrić, Proof-Theoretical Coherence, http://www.mi.sanu.ac.rs/ kosta/coh.pdf, 2007.

[DP15] K. Došen, Z Petrić, Weak Cat-Operads, *Logical Methods in Computer Science, Volume 11, Issue 1*, 2015.

[CO16] J. Obradovic, Monoidal-like definition of cyclic operads, 2016.

Interpreting Sequent Calculi as Client–Server Games

Christian G. Fermüller

Vienna University of Technology, Austria

Keywords:

game semantics, substructural logics, sequent calculus

Resource consciousness is routinely cited as a motivation for substructural logics (see, e.g., [6]). But usually the reference to resources is kept informal, like in Girard's well-known example of being able to buy a pack of Camels and/or a pack of Marlboro [4] with a single dollar, illustrating linear implication as well as the ambiguity of conjunction between the "multiplicative" and "additive" reading. The invitation to distinguish, e.g., between a "causal", action-oriented interpretation of implication and a more traditional understanding of implication as a timeless, abstract relation between propositions is certainly inspiring and motivating. However, the specific shape and properties of proof systems for usual substructural logics owe more to a deep analysis of Gentzen's sequent system than to action-oriented models of handling scarce resources of a specific kind. Various semantics, in particular so-called game semantics for (fragments of) linear logics [1, 2] offer additional leverage points for a logical analysis of resource consciousness. But these semantics hardly support a straightforward reading of sequent derivations as actions plans devised by resource conscious agents. Moreover, the inherent level of abstraction often does not match the appeal of (e.g.) Girard's very concrete and simple picture of action-oriented inference.

We introduce a two-person game based on the idea that a proof is an *action-plan*, i.e. a *strategy* for one of the players (the "client") to establish particular *structured information*, given certain information provided the other player (the "server"). The interpretation of game states as (single conclusion) sequents leads to variations of the basic game, that match (various fragments of) intuitionistic linear logic, but also of substructural logics based on variants of Lambek's calculus [5].

To emphasize the indicated shift of perspective relative to traditional interpretations of formulas as sentences or propositions or types we introduce the notion of an *information package (ip)*. An ip is either *atomic* or else built up from given ips $F_1, F_2, \ldots, F_n$ ($n \geq 2$) using the constructors $\mathsf{any\_of}(F_1, \ldots, F_n)$,

$\mathsf{some\_of}(F_1, \ldots, F_n)$, and $(F_1 \,\mathsf{given}\, F_2)$. Among the atomic ips is the *elementary inconsistency* $\bot$.

In our **C/S**(I)-game, a *client* **C** maintains that the information packaged as H can be obtained from the information represented by the ips $G_1, \ldots, G_n$, provided by a *server* **S**, via stepwise reduction of complex ips into simpler ones. At any state of the game, the *bunch of information provided by* ***S*** is a (possibly empty) multiset of ips. The ip H which **C** currently claims to be obtainable from that information is called **C**'s *current ip*. The corresponding state is denoted by $G_1, \ldots, G_n \triangleright H$. The game proceeds in *rounds* that are always initiated by **C** and, in general, solicit some action from **S**. There are two different types of requests that **C** may submit to **S**: (1) UNPACK a non-atomic ip provided by you (i.e. the server), and (2) CHECK my (i.e. the clients) current ip.

In a request of type UNPACK **C** points to an ip G in the bunch of information provided by **S** and the game proceeds as follows:

(U^*_{any}) If $G = \mathsf{any\_of}(F_1, \ldots, F_n)$ then **C** chooses an $i \in \{1, \ldots, n\}$ and **S** has to add F_i to the bunch of provided information, accordingly.

(U^*_{some}) If $G = \mathsf{some\_of}(F_1, \ldots, F_n)$ then **S** chooses an $i \in \{1, \ldots, n\}$ and adds F_i to the bunch of provided information, accordingly.

(U^*_{given}) If $G = (F_1 \,\mathsf{given}\, F_2)$ then **S** chooses whether to add F_1 to the bunch of provided information or whether to force **C** to replace her current ip by F_2.

($U^+_\bot$) If $G = \bot$ then the game ends and **C** wins.

If the request is of type CHECK then the game proceeds according to the form of **C**'s current ip H.

(C_{any}) If $H = \mathsf{any\_of}(F_1, \ldots, F_n)$ then **S** chooses an $i \in \{1, \ldots, n\}$ and **C** has to replace the current ip by F_i, accordingly.

(C_{some}) If $H = \mathsf{some\_of}(F_1, \ldots, F_n)$ then **C** chooses an $i \in \{1, \ldots, n\}$ and replaces the current ip by F_i, accordingly.

(C_{given}) If $G = (F_1 \,\mathsf{given}\, F_2)$ then F_2 is added to the bunch of provided information and **C**'s current ip is replaced by F_1.

(C^+_{atom}) If H is atomic then the game ends and **C** wins if an occurrence of H is among the bunch of information provided by **S**.

The adequateness of the **C/S**(I)-game for intuitionistic logic I is shown via calculus **LIk**, a variant of Gentzen's sequent calculus **LI**. Instead of referring to **LIk** one may introduce 'bookkeeping rules' into the game. In particular, weakening corresponds to the rule *Dismiss* that allows **C** to eliminate an ip from the bunch of information provided by **S**, while contraction corresponds to the rule *Copy*, enabling **C** to duplicate a given ip. With these bookkeeping rules in place, one may modify the rules U^*_{any}, U^*_{some}, and U^*_{given}, by dismissing the selected ip G after unpacking, instead of adding the unpacked components to the bunch of provided information. Similarly rules $U^+_\bot$ and C^+_{atom} are modified to match the axioms of **LI**, instead of those of **LIk**.

The most important step in converting the **C/S**(I)-game into a 'resource conscious' game, is based on the following observation regarding rules that entail a

choice by and thus require **C** to be prepared to act in more than just one possible successor state to the current state. The above rules allow **C** to use *all* the information provided by **S** in *each* of the possible successor states. If, instead, we require **C** to declare which ips she intends to use for which of those options—taking care that she is using each occurrence of an ip exactly once—then we arrive at rules that match multiplicative instead of additive connectives. We illustrate this principle by the CHECK-rule for the new constructor each_of, corresponding to multiplicative conjunction in intuitionistic linear logic **ILL**.

(C_{each}) If $H = \mathsf{each\_of}(F_1, F_2)$ then **C** has to split the bunch (multiset) Π of information provided by **S** into $\Pi_1 \uplus \Pi_2 = \Pi$ and then let **S** choose whether to continue the game in state $\Pi_1 \triangleright F_1$ or in state $\Pi_2 \triangleright F_2$.

Analogously, one may define a multiplicative version U^m_{given} of U_{given}, matching linear implication. To obtain a game for full **ILL** we drop *Copy* and *Dismiss* and introduce the constructor arbitrary_many and rules for dismissing, adding another copy, or replacing $\mathsf{arbitrary\_many}(F)$ by F, respectively.

To obtain a variant of the **C**/**S**(I)-game that interprets *Full Lambek Calculus* **FL** [5] one identifies the bunch of information provided by **S** with a *list* (instead of a multiset) of ips and replaces the rule C_{given} by two variants that specify whether F_2 of $(F_1 \,\mathsf{given}\, F_2)$ is added at the beginning or at the end of the list. In this manner we actually obtain a family of games characterizing substructural logics corresponding to different types of residuated lattices (see [3]). This in turn provides the basis for interpreting lattice elements as contents of information packages.

Acknowledgement

Supported by Austrian Science Foundation, FWF grant P25417-G15 LogFraDiG.

References

[1] S. Abramsky, R. Jagadeesan: Games and Full Completeness for Multiplicative Linear Logic. *J. Symbolic Logic*, 59(2) (1994), 543–574.

[2] A. Blass: A Game Semantics for Linear Logic. *A. Pure and Appl. Logic*, 56(1992), 183–220.

[3] N. Galatos, P. Jipsen, T. Kowalski, H. Ono: *Residuated Lattices: An Algebraic Glimpse at Substructural Logics: An Algebraic Glimpse at Substructural Logics.* Elsevier, 2007.

[4] J.-Y. Girard: Linear logic – its syntax and semantics. *London Mathematical Society Lecture Note Series* (1995): 1–42.

[5] J. Lambek: The mathematics of sentence structure. *The American Mathematical Monthly*, 65/3 (1958): 154–170.

[6] F. Paoli: Substructural Logics – A Primer. Springer, 2002.

Computational Efficiency of the GF and the RMF Transforms for Quaternary Logic Functions on CPUs and GPUs

Dušan B. Gajić[1], **Radomir S. Stanković**[2]

[1] *Dept. of Computing and Control, Faculty of Technical Sciences, University of Novi Sad*
Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
[2] *Dept. of Computer Science, Faculty of Electronic Engineering, University of Niš*
Aleksandra Medvedeva 14, 18000 Niš, Serbia
E-mail: [1]dusan.b.gajic@gmail.com, [2]radomir.stankovic@gmail.com

Keywords:
Multiple-valued logic, spectral logic, Galois field (GF) transform, Reed-Muller-Fourier (RMF) transform, parallel computing, GPGPU, CUDA.

A large set of problems in science and engineering have a relatively simple solution in the spectral domain, while resolving them in the original domain requires significant effort [1, 6]. Spectral transforms redistribute the information content of a signal, allowing easier observation of some properties or simplified implementation of certain operations [6].

The application of spectral transforms in the area of digital logic is a task important enough to give rise to a particular subdiscipline called spectral logic [6]. In this area, spectral transforms over finite fields $GF(p)$ or ring of integers modulo p, as, for example, the Galois field (GF) transforms and the Reed-Muller-Fourrier (RMF) transforms, are of considerable interest. This is due to the fact that some of their properties resemble characteristics of the classical Fourier transform. Fast algorithms for the computation of these transforms are of significant importance in their practical applications [3, 9].

We present a comparison of computing times required by the GF and the RMF spectral transforms of quaternary logic functions ($p = 4$) [9]. Further, we discuss the impact of arithmetic operations performed during the computation of these transforms using Cooley-Tukey fast Fourier transform (FFT)-like algorithms on central processing units (CPUs) and graphics processing units (GPUs) [4, 5, 6]. For the implementation on the CPU and the GPU, we use the C++ and the Nvidia CUDA C programming languages, respectively [2, 7]. It

is observed that the nature of arithmetic operations required by the two transforms has a considerable impact on the resulting performance of algorithms for their computation. The efficiency of the algorithm implementation is also found to be dependent on the characteristics of the used computing platform [1].

Traditionally, CPUs with their sequential low-latency, low-thoughput von Neumann architecture were the sole target for performing general-purpose algorithms [1]. GPUs, on the other hand, have a single instruction, multiple data (SIMD) high-latency, high-throughput architecture, which rapidly evolved during the last decade [2]. From a fixed-function system designed purposely for rendering computer graphics, GPUs turned into a fully programmable and highly parallel computational platform [2]. This revolution gave rise to a new field of research in computer science which considers the implementation of general-purpose algorithms on GPUs, abbreviated as GPGPU [2], of which the presented research is an example.

The operations used in the computation of the GF transform can be implemented in different ways, depending on the order of the considered finite (Galois) field [9]. For prime values of p, the field operations are the addition and multiplication modulo p. In this case, we can use modulo p arithmetic operators or we can implement the operations using look-up tables (LUTs). When p is non-prime, the Galois field operations differ from the modulo p arithmetic operators, and, thus, must be realized using LUTs [9]. The RMF transform was introduced in [8], by changing the underlying algebraic structure into the Gibbs algebra. In the case of the RMF, the group operation is modulo p addition for all positive integer values of p, while the multiplication is a convolution-wise (Gibbs) multiplication [9]. For the mathematically rigorous definition and more details on the GF and the RMF transforms, we refer to [8, 9].

Quaternary logic functions ($p = 4$) are of special interest due to the fact that they can be easily encoded by binary values and realized with two-stable state circuits in currently dominant binary devices [9]. Since p is, in this case, a non-prime number, computation of the GF transform requires performing operations implemented as look-up tables (LUTs), while the RMF transform is calculated using modulo 4 operations.

In order to experimentally measure the effect that the required arithmetic operations have on the performance of the two considered spectral transforms, we developed implementations of the corresponding Cooley-Tukey algorithms for their computation on the CPU, using C++, and on the GPU, using CUDA. We randomly generated quaternary logic functions with $n = 8, 9, \ldots, 14$, variables and then processed them using the developed implementations on two different computer systems. The first platform is a desktop PC with an Intel i7 CPU and a Nvidia GeForce GPU, both belonging to the mid-performance level. The second system is a workstation with a high-end Intel Xeon CPU and an entry-level Nvidia Quadro work-station GPU.

We found that computing the RMF transform is, on the CPUs in our experimental systems, from 1.33 to 1.71 times faster than computing the GF transform of the same function. Computational efficiency of the RMF transform is even more evident on GPUs than on CPUs, since most of the transistors in the GPU

hardware are devoted to arithmetic logic units (ALUs) [7]. When performing the considered transforms using CUDA on GPUs, the RMF transform is computed from 1.68 to 5.22 times faster than the GF transform.

Acknowledgment

The research reported in the paper is partly supported by the Ministry of Education and Science of the Republic of Serbia, projects ON174026 (2011-2016) and III44006 (2011-2016).

References

[1] Arndt, J., *Matters Computational: Ideas, Algorithms, Source Code*, Springer, 2010.

[2] Cheng, J., Grossman, M., McKercher, T., *Professional CUDA C Programming Guide*, Wrox Press, 2014.

[3] Gajić, D. B., Stanković, R. S., "Computing spectral transforms used in digital logic on the GPU", in Astola, J., Kameyama, M., Lukac, M., Stanković, R. S. (eds.), *GPU Computing with Applications in Digital Logic*, Tampere International Center for Signal Processing - TICSP Series # 62, Tampere, Finland, 2012, ISBN 978-952-15-2920-7.

[4] Gajić, D. B., Stanković, R. S., "The impact of address arithmetic on the GPU implementation of fast algorithms for the Vilenkin-Chrestenson transform", *Proc. 43rd IEEE Int. Symp. on Multiple-Valued Logic*, Toyama, Japan, May 22-24, 2013, 296-301.

[5] Gajić, D. B., Stanković, R. S., "Computation of the Vilenkin-Chrestenson transform on a GPU", J. Multiple-Valued Logic and Soft Computing, vol. 24, no. 3-5, pp. 317-341, Old City Publishing, Philadelphia, USA, 2015.

[6] Karpovsky, M. G, Stanković, R. S., Astola, J. T., *Spectral Logic and Its Applications for the Design of Digital Devices*, Wiley-Interscience, 2008.

[7] Nvidia, *Nvidia CUDA Programming Guide*, version 7.5, Nvidia Corporation, September 2015.

[8] Stanković, R. S., "Some remarks on Fourier transforms and differential operators for digital functions", *Proc. 22nd IEEE Int. Symp. on Multiple-Valued Logic*, Sendai, Japan, 1992, DOI: 10.1109/ISMVL.1992.186818, 365-370.

[9] Stanković, R. S., Astola, J. T., Moraga, C., *Representation of Multiple-Valued Logic Functions*, Morgan & Claypool Publishers, 2012.

Probabilistic reasoning in type systems

Silvia Ghilezan[1], Jelena Ivetić[1], Zoran Ognjanović[2] and Nenad Savić[1]

[1]Faculty of Technical Sciences, University of Novi Sad, Serbia
[2]Mathematical Institute SASA, Belgrade, Serbia

We introduce a formal model $\mathsf{P}\Lambda^{\cap}$ for reasoning about probabilities of lambda terms with intersection types which is a combination of lambda calculus and probabilistic logic. We propose its syntax, Kripke-style semantics and an infinitary axiomatization. We first endow the language of typed lambda calculus with a probabilistic operator $P_{\geq s}$ and, besides the formulas of the form $M : \sigma$ and its Boolean combinations, we obtain formulas of the form

$$P_{\geq s} M : \sigma$$

to express that the probability that the lambda term M is of type σ is equal to or greater than s. More generally, formulas are of the form $P_{\geq s}\alpha$, where α is a typed lambda statement $M : \sigma$ or its Boolean combination, so the following is a formula of our formal model as well:

$$[P_{=\frac{1}{3}}(M : \sigma \to \tau) \wedge P_{=\frac{2}{3}}(N : \sigma)] \Rightarrow P_{=0}(MN : \tau).$$

Furthermore, using the similar idea as in the classical propositional calculus, we construct the canonical model in order to prove the main result, which is the soundness and strong completeness of $\mathsf{P}\Lambda^{\cap}$ with respect to the proposed model. The idea is to show that every consistent set can be extended to the maximal consistent set and then use the fact that the lambda calculus with intersection types is complete with respect to the filter lambda model ([1]).

Acknowledgements

This work was supported by the Serbian Ministry of Education and Science through projects ON174026, III44006 and ON174008.

References

[1] H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. *A filter lambda model and the completeness of type assignment.* J. Symb. Log., 48(4): 931–940, 1983.

[2] R. Cooper, S. Dobnik, S. Lappin, and S. Larsson. *A probabilistic rich type theory for semantic interpretation.* Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS), pages 72–79, 2014.

[3] M. Coppo and M. Dezani-Ciancaglini. *A new type assignment for λ-terms.* Archiv für mathematische Logik und Grundlagenforschung, 19(1): 139–156, 1978.

[4] R. Fagin, J. Y. Halpern, and N. Megiddo. *A logic for reasoning about probabilities.* Inf. Comput., 87(1/2): 78–128, 1990.

[5] M. Fattorosi-Barnaba and G. Amati. *Modal operators with probabilistic interpretations*, i. Studia Logica, 46(4):383–393, 1995.

[6] S. Ghilezan. *Strong normalization and typability with intersection types.* Notre Dame Journal of Formal Logic, 37(1): 44–52, 1996.

[7] N. J. Nilsson. *Probabilistic logic.* Artif. Intell., 28(1): 71–87, 1986.

[8] Z. Ognjanovic and M. Raskovic. *Some probability logics with new types of probability operators.* J. Log. Comput., 9(2): 181–195, 1999.

Some Notes on Finite and Hyperfinite model theory

Nebojša Ikodinović[1]

[1]Faculty of Mathematics, University of Belgrade, Serbia

Keywords:

Probabilistic logic, Hyperfinite models, The zero-one law

We shall briefly review and summarize results of Gaifman's seminal paper [2], and focus on two important directions of research inspired by [2]. The first one is closely related to logics appropriate for probability structures. The second direction is related to finite model theory originated in computer science.

1. Scott and Krauss [4] extended Gaifman's work in many important ways. They considered language systems closer in expressive power to the σ-algebras of mathematical probability and developed a corresponding model theory and proof theory. The very important advancement in this direction was made by H. Jerome Keisler in his famous paper [3] which views matters mainly from the standpoint of hyperfinite models.

2. That first-order logic has the zero-one law was proved (firstly by Glebskii at all, in 1969, and independently) by Fagin [1] who used Gaifman's extension axioms introduced in [2]. It is well-known that the zero-one law is more than an individual result – it has a similar universal applicability in finite model theory as Compactness Theorem has in infinite model theory.

It is particularly interesting that the work of Fagin [1] is similar in spirit to some results from Keisler's paper [3]. We shall outline some possible further development of probability logics with the aid of techniques of hyperfinite model theory which promises to produce new results.

Acknowledgements

The work presented here is partially supported by the Serbian Ministry of Education and Science through projects 174026 and III044006.

References

[1] R. Fagin, *Probabilities on finite models*, J. Symb. Logic **41** (1976), 50–58.

[2] H. Gaifman, *Concerning measures in first order calculi*, Israel J. Math. **2** (1964), 1–18.

[3] H. J. Keisler, *Hyperfinite model theory*, in: R. O. Gandy, J. M. E. Hyland (eds.) *Logic Colloquim 76*, North-Holland (1977), 5–110.

[4] D. Scott, P. Krauss, *Assigning probabilities to logical formulas*, in: J. Hintikka, P. Suppes (eds), *Aspects of Inductive logic*, North-Holland Publishing Company, Amsterdam (1966), 219–264.

Computable points and local computability

Zvonko Iljazović and Lucija Validžić

University of Zagreb, Croatia

Keywords:

computable metric space, semi-computable set, computable point

A compact subset of Euclidean space (or a computable metric space) is computable if it can be effectively approximated by a finite set of rational points with arbitrary precision. A compact set S is semi-computable if we can effectively enumerate all finite unions of rational balls which cover S. Each computable set is semi-computable, but a semi-computable set need not be computable. In fact, while computable points in each computable set are dense, there exists a nonempty semi-computable subset of the real line which does not contain any computable point.

We investigate conditions under which computable points in a semi-computable set are dense. Related to this, we examine the computability of a set at a point. We also extend the notion of semi-computability to non-compact sets and observe semi-computable manifolds and sets which have topological type of a polyhedron.

References

[1] V. Brattka, G. Presser, *Computability on subsets of metric spaces*, Theoretical Computer Science 305, pp. 43–76, 2003.

[2] Z. Iljazović, *Compact manifolds with computable boundaries*, Logical Methods in Computer Science 9(4:19), pp. 1–22, 2013.

[3] T. Kihara, *Incomputability of Simply Connected Planar Continua*, Computability 1(2), pp. 131–152, 2012.

[4] J.S. Miller, *Effectiveness for Embedded Spheres and Balls*, Electronic Notes in Theoretical Computer Science 66, pp. 127–138, 2002.

[5] M. Pour-El, I. Richards, *Computability in Analysis and Physics*, Springer-Verlag, Berlin-Heielberg-New York, 1989.

[6] E. Specker, *Der Satz vom Maximum in der rekursiven Analysis*, Constructivity in Mathematics (A. Heyting, ed.), North Holland Publ. Comp., Amsterdam, pp. 254–265, 1959.

[7] K. Weihrauch, *Computable Analysis*, Springer, Berlin, 2000.

Timed Multiset Rewriting and the Verification of Time-Sensitive Distributed Systems

Max Kanovich[1,5], Tajana Ban Kirigin[2], Vivek Nigam[3], Andre Scedrov[4,5] and Carolyn Talcott[6]

[1]University College London, UK
[2]University of Rijeka, HR
[3]Federal University of Paraíba, Brazil
[4]University of Pennsylvania, USA
[5]National Research University Higher School of Economics, Russian Federation
[6]SRI International, USA

Keywords:

Multiset Rewriting, Distributed Systems, Computational Complexity, Maude

Time-Sensitive Distributed Systems (TSDS), such as applications using autonomous drones, achieve goals under possible environment interference (*e.g.*, winds). Moreover, goals are often specified using explicit time constraints which must be satisfied by the system *perpetually*. For example, drones carrying out the surveillance of some area must always have *recent pictures*, *i.e.*, at most M time units old, of some strategic locations.

We propose a Multiset Rewriting language with explicit time for specifying and analysing TSDSes. We introduce two properties, *realizability* (some trace is good) and *survivability* (where, in addition, all admissible traces are good). A good trace is an infinite trace in which goals are perpetually satisfied. The transition to properties over infinite traces leads to many challenges as one can easily fall into undecidability fragments of verification problems. The main challenge is to identify the syntatical conditions on specifications so that the survivability and feasibility problems fall into a decidable fragment, and at the same time, that interesting examples can be specified. Also, the notion that a system satisfies a property perpetually implies that the desired property should be valid at all time instances independent of environment interference. Another issue is that systems should not be allowed to perform an unbounded number of actions in a single time instance, a problem similar to the Zeno paradox.

We propose a class of systems called *progressive timed systems* (PTS), where intuitively only a finite number of actions can be carried out in a bounded time period. We define a language for specifying realizability and suvivability properties which allows the specification of many interesting problems in TSDS.

We prove that for this class of systems both the realizability and the survivability problems are PSPACE-complete. Furthermore, if we impose a bound on time (as in bounded model-checking), we show that for PTS, realizability becomes NP-complete, while survivability is in the Δ_2^p class of the polynomial hierarchy.

Finally, we demonstrate that the rewriting logic system Maude can be used to automate time bounded verification of PTS.

References

[1] M. Kanovich, T. Ban Kirigin, V. Nigam, A. Scedrov, C. Talcott, *Timed Multiset Rewriting and the Verification of Time-Sensitive Distributed Systems*, FORMATS 2016 : 14th International Conference on Formal Modeling and Analysis of Timed Systems, 08/2016.

[2] M. Kanovich, T. Ban Kirigin, V. Nigam, A. Scedrov, C. Talcott, *Timed Multiset Rewriting and the Verification of Time-Sensitive Distributed Systems*, arXiv:1606.07886

Can we mitigate the attacks on Distance-Bounding Protocols by using challenge-response rounds repeatedly ?

Max Kanovich[1,5], Tajana Ban Kirigin[2], Vivek Nigam[3], Andre Scedrov[4,5] and Carolyn Talcott[6]

[1]University College London, UK
[2]University of Rijeka, HR
[3]Federal University of Paraíba, Brazil
[4]University of Pennsylvania, USA
[5]National Research University Higher School of Economics, Russian Federation
[6]SRI International, USA

Keywords:

Distance Bounding Protocols, Attack, Probability

Distance Bounding Protocols are used to infer an upper-bound on the distance between two participants by measuring the round trip time of a challenge response round launched by the Verifier, who owns the desired resource, to a Prover, who wants access to the resource.

A Verifier, who owns the desired resource, sends a challenge to the Prover, who wants the resource, remembering when the challenge was sent. The Prover then responds to the challenge (as quick as possible). From the round-trip time, Verifier can infer an upper-bound on the distance to Prover. Only if Prover is within some pre-established distance, Verifier grants him access to the resource, e.g, open a door.

In our previous work [2], we discovered a new attack on Distance Bounding Protocols, called Attack In-Between-Ticks, showing that an Intruder can gain access to a resource although he is not within the pre-established distance to Verifier. The attack exploits the differences between discrete measurements used by Verifier and the actual distance. We then speculated that the Attack in Between Ticks could be mitigated by using a large number of challenge response rounds.

This paper works out the details building the formal machinery to support this idea. We obtain some surprising (non-intuitive) results.

We show that in the case where Verifier decides to grant the access by the simple majority, the effect of the repeated challenge-response rounds can mitigate the attack but only for the specific values of the probability of the erroneous decision in one round.

Whereas in the case where Verifier decides to grant the access by the large majority (that is, with gaining a large specified level of support, for example, Prover responding in time in two thirds of the challenges) the idea of repeated challenge-response rounds works perfectly well for our protocol. In particular, having observed the "acceptance challenge-response events" in the two-thirds majority of rounds, Verifier can establish the desired upper bounds for the 'actual' challenge-response time interval but only with the high probability.

References

[1] Max Kanovich, Tajana Ban Kirigin, Vivek Nigam, Andre Scedrov, and Carolyn Talcott. *Can we mitigate the attacks on Distance-Bounding Protocols by using challenge-response rounds repeatedly?* Workshop on Foundations of Computer Security (FCS), Affiliated with IEEE CSF 2016., Lisbon, Portugal, 2016.

[2] Max Kanovich, Tajana Ban Kirigin, Vivek Nigam, Andre Scedrov, and Carolyn Talcott. *Discrete vs. dense times in the analysis of cyber-physical security protocols.* Principles of Security and Trust (POST) - 4th International Conference, pages 259–279, 2015.

Justification with Propositional Nominals

Alexander Kashev[1]

[1]Institute of Computer Science, University of Bern, Switzerland

Keywords:

justification logic, atomic models, updates, dynamic epistemic logic

The broad field of epistemic logic aims to reason about knowledge and belief. While modal logic deals with the fact of knowing or believing a certain statement, the framework of justification logic introduced by Artemov [1] provides tools to formalize explicit reasons for such belief, which makes it possible to analyze many epistemic problems and puzzles from a new perspective [2, 3, 5, 7, 9, 10].

Ordinary justification logic provides a static picture of all justified beliefs that emerge from a given set of premises. It's a natural extenison to consider a dynamically changing set of reasons for beliefs, for example as a result of communication; this is studied in the field of dynamic justification logic, started by Renne [13] and continued in [6, 8].

Kuznets and Studer introduced [12] a dynamic justification logic $\mathsf{JUP}_{\mathsf{CS}}$, providing a simple axiomatization for belief expansion and minimal change. It adds a new kind of atomic evidence term, $\mathsf{up}(A)$, representing the evidence for a formula A after updating the belief set with A. $\mathsf{JUP}_{\mathsf{CS}}$ is shown to be sound and complete with respect to a class of basic modular models [4, 11], with the basic evaluation generated inductively from an evidence basis that uses only atomic terms.

Being able to restrict the model defintion to atomic terms produces simple models to work with, but comes at a price: term application has to carry a record of the formula that was used in the application, e.g. $t \cdot_A s$, and the application axiom was modified to reflect this:

$$t : (A \to B) \land s : A \leftrightarrow t \cdot_A s : B$$

This is an uncommon addition to justification logic, first introduced by Renne [14], and it naturally raised the question whether it's possible to remove the subscript and use the more traditional application axiom for justification logic.

This talk presents a stepping stone in the study of this question. To examine its most basic form, we look at the situation after a finite set of updates with atomic propositional statements, omitting the dynamics of updates. Namely, for propositional variables in this fixed update set, we introduce *nominals*: terms that justify exactly one formula, which is enforced on axiom level.

An axiom system $\mathsf{JN}^{\mathsf{V}}_{\mathsf{CS}}$ is formulated to capture that property, and shown to be sound and complete with respect to a class of basic modular models. This semantics are shown to have the finite model property.

However, this class of models is still less natural than the original semantics for $\mathsf{JUP}_{\mathsf{CS}}$. We show that, for appropriate constant specifications, it's possible to obtain a completeness result for a more natural class of atomic models through a model reduction procedure we call *atomization*.

Those results represent the first step in adapting $\mathsf{JUP}_{\mathsf{CS}}$ to subscript-free application. The primary direction for further work is re-introducing dynamic updates and studying the epistemic properties of the resulting system. Another possible direction is extending nominals to non-atomic propositions.

References

[1] S. N. Artemov, *Explicit provability and constructive semantics*, Bulletin of Symbolic Logic, 7(1):1–36, 2001.

[2] S. N. Artemov, *Justified common knowledge*, Theoretical Computer Science, 357(1–3):4–22, 2006.

[3] S. N. Artemov, *The logic of justification*, The Review of Symbolic Logic, 1(4):477–513, 2008.

[4] S. N. Artemov, *The ontology of justifications in the logical setting*, Studia Logica, 100(1–2):17–30, 2012.

[5] A. Baltag, B. Renne, and S. Smets, *The logic of justified belief, explicit knowledge, and conclusive evidence*, Annals of Pure and Applied Logic, 165(1):49–81, 2014. Published online in August 2013.

[6] S. Bucheli, R. Kuznets, B. Renne, J. Sack, and T. Studer, *Justified belief change*, in Xabier Arrazola and María Ponte, editors, LogKCA-10, Proceedings of the Second ILCLI International Workshop on Logic and Philosophy of Knowledge, Communication and Action, pages 135–155, University of the Basque Country Press, 2010.

[7] S. Bucheli, R. Kuznets, T. Studer, *Justifications for common knowledge*, Journal of Applied Non-Classical Logics, 21(1):35–60, 2011.

[8] S. Bucheli, R. Kuznets, and T. Studer, *Partial realization in dynamic justification logic*, in Lev D. Beklemishev and Ruy de Queiroz, editors, Logic, Language, Information and Computation, 18th International Workshop, WoLLIC 2011, Philadelphia, PA, USA, May 18–20, 2011, Proceedings, volume 6642 of *Lecture Notes in Artificial Intelligence*, pages 35–51, Springer, 2011.

[9] I. Kokkinis, P. Maksimović, Z. Ognjanović, T. Studer, *First steps towards probabilistic justification logic*, Logic Journal of IGPL, 23(4), 662–687, 2015.

[10] I. Kokkinis, Z. Ognjanović, T. Studer, *Probabilistic Justification Logic*, in S. Artemov, A. Nerode, editors, Proceedings of Logical Foundations of Computer Science LFCS'16, volume 9537 of *Lecture Notes in Computer Science*, pages 174–186, Springer, 2016.

[11] R. Kuznets, T. Studer, *Justifications, ontology, and conservativity*, in T. Bolander, T. Braüner, S. Ghilardi, L. Moss, editors, Advances in Modal Logic, Volume 9, pages 437–458, College Publications, 2012.

[12] R. Kuznets, T. Studer. *Update as evidence: belief expansion*, Logical Foundations of Computer Science, volume 7734 of *Lecture Notes in Computer Science*, pages 266–279, Springer Berlin Heidelberg, 2013.

[13] B. Renne, *Dynamic Epistemic Logic with Justification*, PhD thesis, CUNY Graduate Center, 2008.

[14] B. Renne, *Evidence elimination in multi-agent justification logic*, in Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge, ACM, 2009.

Algorithmic Knowledge Discovery with Lattices of Closed Descriptions

Sergei O. Kuznetsov[1]

[1]National Research University Higher School of Economics, Moscow, Russia

Keywords:

Galois connection, closed descriptions, implicative dependencies

Galois connection between sets of objects from a subject domain and their partially ordered descriptions define a closure operator and respective closed sets of objects and closed descriptions, which make two antitone lattices. When descriptions are sets of attributes, these two lattices make a concept (Galois) lattice [1]. On the one hand, the lattice of closed descriptions gives a taxonomy of a subject domain, where each class of objects is given by its specific closed description. On the other hand, the closure operator gives a natural definition of implicative dependencies related to functional dependencies and Horn formulas. The lattice diagram gives a natural concise representation of all association rules (partial implications) that hold in the domain. We consider relationships between models of knowledge discovery naturally described in terms of lattices of closed descriptions [2, 3], present respective results on algorithmic complexity and discuss some applications.

References

[1] R.Wille, B.Ganter, *Formal Concept Analysis. Mathematical Foundations*, Springer, 1999.

[2] S.O. Kuznetsov, *Complexity of Learning in Concept Lattices from Positive and Negative Examples*, Discrete Applied Mathematics, no. 142(1-3), pp. 111-125, 2004.

[3] S.O. Kuznetsov, *Scalable Knowledge Discovery in Complex Data with Pattern Structures* In: P. Maji, A. Ghosh, M.N. Murty, K. Ghosh, S.K. Pal, Eds., Proc. 5th International Conference Pattern Recognition and Machine Intelligence (PReMI'2013), Lecture Notes in Computer Science (Springer), Vol. 8251, pp. 30-41, 2013.

The Lambek Calculus with Unary Connectives

Stepan Kuznetsov
Steklov Mathematical Institute (Moscow), Russia
(partially based on joint work with M. Kanovich, A. Scedrov, and N. Ryzhkova)

1 The Lambek Calculus

The Lambek calculus **L** was introduced by J. Lambek [7]. Formulae *(types)* of **L** are are built from a countable set of variables *(primitive types)* $\mathrm{Var} = \{p, q, r, \dots\}$ using three binary connectives: $\setminus$ *(left division),* $/$ *(right division),* and $\cdot$ *(multiplication).* The set of all types is denoted by Tp. The Lambek calculus derives *sequents* of the form $\Pi \to A$, where A is a type and Π is a sequence of types. In **L**, Π is required to be non-empty. There exists a variant of the Lambek calculus, $\mathbf{L}^*$, without this restriction.

The axioms and rules of **L** are as follows: $A \to A$

$$\frac{\Pi, A \to B}{\Pi \to B \,/\, A}\ (\to /) \qquad \frac{A, \Pi \to B}{\Pi \to A \setminus B}\ (\to \setminus)$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, (B \,/\, A), \Pi, \Delta \to C}\ (/ \to) \qquad \frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, \Pi, (A \setminus B), \Delta \to C}\ (\setminus \to)$$

$$\frac{\Gamma, A, B, \Delta \to C}{\Gamma, (A \cdot B), \Delta \to C}\ (\cdot \to) \qquad \frac{\Pi_1 \to A \quad \Pi_2 \to B}{\Pi_1, \Pi_2 \to A \cdot B}\ (\to \cdot)$$

Lambek syntactic types can be naturally interpreted as formal languages over an alphabet Σ. For the interpretation w, the following should hold:

$$w(A \cdot B) = w(A) \cdot w(B) = \{uv \mid u \in w(A), v \in w(B)\}$$
$$w(B \,/\, A) = w(B) \,/\, w(A) = \{v \mid (\forall u \in w(A))\, vu \in w(B)\}$$
$$w(A \setminus B) = w(A) \setminus w(B) = \{v \mid (\forall u \in w(A))\, uv \in w(B)\}$$

Note that this definition works differently for **L** and $\mathbf{L}^*$ (for **L**, the empty word is not included in the languages). A sequent $A_1, \dots, A_n \to B$ is true under interpretation w, if $w(A_1) \cdot \ldots \cdot w(A_n) \subseteq w(B)$. Both variants of the Lambek calculus are sound and complete w.r.t. this interpretation:

Theorem 1 (M. Pentus)**.** *A sequent is derivable in* **L** *(resp., in* $\mathbf{L}^*$*) iff it is true under any interpretation* $w\colon \mathrm{Tp} \to \mathcal{P}(\Sigma^+)$ *(resp.,* $w\colon \mathrm{Tp} \to \mathcal{P}(\Sigma^*)$*).* [10]

Lambek *categorial grammars* are finite correspondences between Lambek types and letters of an alphabet. The word $a_1 \dots a_n$ belongs to the language generated by such grammar if there exist types $A_1, \dots, A_n$ in the correspondence with letters $a_1, \dots, a_n$ resp., such that $A_1, \dots, A_n \to H$ is derivable in $\mathbf{L}$ or one of its variants. Here H is a fixed type, usually primitive.

Theorem 2 (M. Pentus)**.** *Grammars based on* $\mathbf{L}$ *(resp., on* $\mathbf{L}^*$*) generate precisely the class of context-free languages without the empty word (resp., the class of all context-free languages).* [9]

2 The Reversal Operation

The unary *reversal* operation is defined as follows: $M^{\mathrm{R}} = \{a_n \dots a_1 \mid a_1 \dots a_n \in M\}$ for any formal language M. The extension of $\mathbf{L}$ with the unary $(\cdot)^{\mathrm{R}}$ connective, $\mathbf{L}^{\mathrm{R}}$, is obtained from $\mathbf{L}$ by adding the following rules ($\Gamma^{\mathrm{R}} = A_n^{\mathrm{R}}, \dots, A_1^{\mathrm{R}}$ for $\Gamma = A_1, \dots, A_n$):

$$\frac{\Gamma \to C}{\Gamma^{\mathrm{R}} \to C^{\mathrm{R}}}\ ({}^{\mathrm{R}} \to {}^{\mathrm{R}}) \qquad \frac{\Gamma, A^{\mathrm{RR}}, \Delta \to C}{\Gamma, A, \Delta \to C}\ ({}^{\mathrm{RR}} \to)_{\mathrm{E}} \qquad \frac{\Gamma \to C^{\mathrm{RR}}}{\Gamma \to C}\ (\to {}^{\mathrm{RR}})_{\mathrm{E}}$$

The good properties of the Lambek calculus keep valid for its extension with the reversal operation ([4] for $\mathbf{L}$, [5] for $\mathbf{L}^*$):

Theorem 3. *The calculi* $\mathbf{L}^{\mathrm{R}}$ *and* $\mathbf{L}^{*\mathrm{R}}$ *are sound and complete w.r.t. interpretations of types as formal languages.*

Theorem 4. $\mathbf{L}^{\mathrm{R}}$*-grammars (resp.,* $\mathbf{L}^{*\mathrm{R}}$*-grammars) generate precisely the class of context-free languages without the empty word (resp., the class of all context-free languages).*

3 The (Sub)exponential

The *exponential* modality, ! (called "bang"), is inherited from linear logic. It is governed by the following rules. Here we consider only the $\mathbf{L}^*$ case, since the $\mathbf{L}$ one is much more subtle (see [1]).

$$\frac{\Gamma, A, \Delta \to B}{\Gamma, !A, \Delta \to B}\ (! \to) \qquad \frac{!A_1, \dots, !A_n \to B}{!A_1, \dots, !A_n \to !B}\ (\to !) \qquad \frac{\Gamma, \Delta \to B}{\Gamma, !A, \Delta \to B}\ (\text{weak})$$

$$\frac{\Gamma, !A, !A, \Delta \to B}{\Gamma, !A, \Delta \to B}\ (\text{contr}) \qquad \frac{\Gamma, !A, \Delta, \Phi \to B}{\Gamma, \Delta, !A, \Phi \to B}\ (\text{perm}_1) \qquad \frac{\Gamma, \Delta, !A, \Phi \to B}{\Gamma, !A, \Delta, \Phi \to B}\ (\text{perm}_2)$$

We also consider a less powerful modality, for which we impose contraction and permutation, but not weakening. We also denote it by ! and call a *subexponential*. This modality is motivated from the linguistic side [8].

Theorem 5. *Grammars based on the extension of* $\mathbf{L}^*$ *with* ! *(both with and without* (weak)*) generate all recursively enumerable languages.* [2]

This is obtained by encoding finite *theories* over $\mathbf{L}^*$ inside sequents using the (sub)exponential modalities.

Corollary 6. *The derivability problem for* $\mathbf{L}^*$ *extended with* ! *is undecidable.*

However, in linguistical practice ! is usually applied only to variables. For this case, the derivability problem is decidable and belongs to NP [2].

4 The Kleene Star

Yet another important operation on formal languages is the *Kleene star:* $A^* = \bigcup_{n=0}^{\infty} A^n$. For the Kleene star, we propose the following rules extending $\mathbf{L}^*$:

$$\frac{\Gamma_1 \to A \quad \dots \quad \Gamma_n \to A}{\Gamma_1, \dots, \Gamma_n \to A^*} \; (\to {}^*)$$

$$\frac{\Gamma, \Delta \to C \quad \Gamma, A, A^*, \Delta \to C}{\Gamma, A^*, \Delta \to C} \; ({}^* \to)_{\mathrm{L}} \qquad \frac{\Gamma, \Delta \to C \quad \Gamma, A^*, A, \Delta \to C}{\Gamma, A^*, \Delta \to C} \; ({}^* \to)_{\mathrm{R}}$$

In this system we allow infinite branches of proofs.

For the fragment without $\cdot$ and where * is allowed only in subformulae of the form $A^* \setminus B$ or B / A^*, this calculus enjoys completeness w.r.t. interpretations of types as formal languages [6].

There is an open question whether we could take only regular (cyclic) proofs, like in [11]. However, if we take both * and !, the answer is negative: using results from [3] for theories over Kleene algebras and then encoding the theory into the sequent using the construction from [2], one obtains Π^0_2-hardness of the system. Therefore, it is not equivalent to any system with finite proofs.

Acknowledgements

The author thanks all his colleagues who participated in discussions on the topics considered in this abstract, in particular, Lev Beklemishev, Michael Kaminski, Glyn Morrill, Fedor Pakhomov, Mati Pentus, Daniyar Shamkanov, Alexey Sorokin, Stanislav Speranski, and, of course, his co-authors, Max Kanovich, Andre Scedrov, and Nadezhda Ryzhkova.

Stepan Kuznetsov's work was supported by the Russian Foundation for Basic Research (grants 11-01-00281-a, 12-01-00888-a, 14-01-00127-a, and 15-01-09218-a), by the Presidential Council for Support of Leading Scientific Schools (grants NŠ-65648.2010.1, NŠ-5593.2012.1, NŠ-1423.2014.1, and NŠ-9091.2016.1), and by the Scientific and Technological Cooperation Programme Switzerland – Russia (project "Computational Proof Theory", 2010–2012).

References

[1] M. Kanovich, S. Kuznetsov, A. Scedrov. On Lambek's restriction in the presence of exponential modalities. Proc. LFCS 2016 (LNCS vol. 9537), 141–158.

[2] M. Kanovich, S. Kuznetsov, A. Scedrov. Undecidability of the Lambek calculus with a relevant modality. Proc. FG 2015/2016 (LNCS vol. 9804), 2016 (*to appear,* arXiv: 1601.06303).

[3] D. Kozen. On the complexity of reasoning in Kleene algebra. Information and Control **179**, 2002, 152–162.

[4] S. Kuznetsov. L-completeness of the Lambek calculus with the reversal operation. Proc. LACL 2012 (LNCS vol. 7351), 151–160.

[5] S. Kuznetsov. L-completeness of the Lambek calculus with the reversal operation allowing empty antecedents. Lambek 90 Festschrift (LNCS vol. 8222), 2014, 268–278.

[6] S. Kuznetsov, N. Ryzhkova. A fragment of the Lambek calculus with iteration (in Russian). Proc. Mal'tsev Meeting 2015, Novosibirsk.

[7] J. Lambek. The mathematics of sentence structure. Amer. Math. Monthly, 1958, 65(3), 154–170.

[8] G. Morrill, O. Valentín. Computational coverage of TLG: Nonlinearity. Proc. NLCS 2015 (EPiC Series, vol. 32), 51–63.

[9] M. Pentus. Lambek grammars are context free. Proc. 8th IEEE Symposium on Logic in Computer Science, 1993, 429–433.

[10] M. Pentus. Models for the Lambek calculus. APAL **75**, 1995, 179–213.

[11] D. S. Shamkanov. Circular proofs for the Gödel–Löb provability logic. Math. Notes **96**, 2014, 575–585.

On Subexponentials, Focusing and Modalities in Concurrent Systems

Vivek Nigam[1]

[1]Federal University of Paraíba, Brazil

Keywords:

Linear Logic, Subexponentials, Logical Frameworks, Modalities

In order to specify the behavior of distributed agents or the policies governing a distributed system, it is often necessary to reason by using different types of modalities, such as time, space, or even the epistemic state of agents. For instance, the access-control policies of a building might allow Bob to have access only in some pre-defined time, such as its opening hours. Another policy might also allow Bob to ask Alice who has higher credentials to grant him access to the building, or even specify that Bob has only access to some specific rooms of the building. Following this need, many formalisms have been proposed to specify, program and reason about such policies, *e.g.*, Ambient Calculus [1], Concurrent Constraint Programming [2], Authorization Logics [3], just to name a few.

Logic and proof theory have often inspired the design of many of these formalisms. For example, Saraswat *et al.* proposed Concurrent Constraint Programming (CCP), a model for concurrency that combines the traditional operational view of process calculi with a declarative view based on logic [2]. Agents in CCP *interact* with each other by *telling* and *asking* information represented as *constraints* to a global store. Later, Fages *et al.* in [4] proposed Linear Concurrent Constraint (`lcc`), inspired by linear logic [5], to allow the use of linear constraints, that is, tokens of information that once used by an agent are removed from the global store.

In order to capture the behavior of distributed systems which take into account spatial, temporal and/or epistemic properties, new formalisms have been proposed. For instance, Saraswat *et al.* proposed `tcc` [6], which extends CCP with time modalities. Later, Knight *et al.* [7] proposed a CCP-based language with spatial and epistemic modalities. Some of these developments have also been followed by a similar development in proof theory. For instance, Nigam proposed a framework for linear authorization logics [8], which allow the specification of access control policies that may mention the affirmations, possessions and knowledge of

principals and demonstrated that a wide range of linear authorization policies can be specified in linear logic with subexponentials (SELL) [9, 10].

In this talk, we show that time, spatial, and epistemic modalities can be *uniformly* specified in a single logical framework called SELLF$^{\Cap}$ which extends intuitionistic SELL with universal ($\Cap$) and existential ($\Cup$) quantifiers over subexponentials. SELL$^{\Cap}$ has good proof-theoretic properties: it admits cut-elimination and it has a complete focusing discipline [11], giving rise to the focused system SELLF$^{\Cap}$.

Then we will show that subexponentials can be interpreted as spatial, epistemic and temporal modalities, thus providing a framework for specifying concurrent systems with these modalities. This is accomplished by encoding in SELLF$^{\Cap}$ different CCP languages, for which the proposed quantifiers play an important role. For instance, they enable the use of an *arbitrary number of subexponentials*, required to model the unbounded nesting of modalities, which is a common feature in epistemic and spatial systems. This does not seem possible in existing logical frameworks such as [12] which do not contain subexponentials nor its quantifiers. Finally, the focusing discipline enforces that the obtained encodings are *faithful* w.r.t. CCP's operational semantics in a strong sense: one operational step matches exactly one logical phase. This is the strongest level of adequacy called adequacy on the level of derivations [13]. Such level of adequacy is not possible for similar encodings of linear CCP systems, such as [4].

Another important feature of subexponentials is that they can be organized into a pre-order, which specifies the provability relation among them. By coupling subexponential quantifiers with a suitable pre-order, it is possible to specify *declaratively* the rules in which agents can manipulate information. For example, an agent cannot see the information contained in a space that she does not have access to. The boundaries are naturally implied by the pre-order of subexponentials.

Finally, if time permits, we will talk about SELLS$^{\Cap}$ which extends SELL$^{\Cap}$ by allowing subexponentials to be specified as algebraic structures called $\times$-poset. This extension allows for the specification of preferences, formally specified as soft-constraints. We then show that besides being able to capture existing CCP-languages, we propose a novel CCP language called Subexponential CPP, which includes features such as 1) computational spaces where agents can tell and ask preferences (soft-constraints); 2) systems where spatial and temporal modalities can be combined; 3) shared spaces for communication that can be dynamically established; and 4) systems that can dynamically create nested spaces.

This talk is based on the publications [14] and [15].

References

[1] L. Cardelli, A. D. Gordon, Mobile ambients, Theor. Comput. Sci. 240 (1) (2000) 177–213.

[2] V. A. Saraswat, Concurrent Constraint Programming, MIT Press, 1993.

[3] M. Abadi, M. Burrows, B. W. Lampson, G. D. Plotkin, A calculus for access control in distributed systems, ACM Trans. Program. Lang. Syst. 15 (4) (1993) 706–734.

[4] F. Fages, P. Ruet, S. Soliman, Linear concurrent constraint programming: Operational and phase semantics, Inf. Comput. 165 (1) (2001) 14–41.

[5] J.-Y. Girard, Linear logic, Theor. Comput. Sci. 50 (1987) 1–102.

[6] V. A. Saraswat, R. Jagadeesan, V. Gupta, Timed default concurrent constraint programming, J. Symb. Comput. 22 (5/6) (1996) 475–520.

[7] S. Knight, C. Palamidessi, P. Panangaden, F. D. Valencia, Spatial and epistemic modalities in constraint-based process calculi, in: M. Koutny, I. Ulidowski (Eds.), CONCUR, Vol. 7454 of Lecture Notes in Computer Science, Springer, 2012, pp. 317–332.

[8] V. Nigam, On the complexity of linear authorization logics, in: LICS, IEEE, 2012, pp. 511–520.

[9] V. Danos, J.-B. Joinet, H. Schellinx, The structure of exponentials: Uncovering the dynamics of linear logic proofs, in: G. Gottlob, A. Leitsch, D. Mundici (Eds.), Kurt Gödel Colloquium, Vol. 713 of Lecture Notes in Computer Science, Springer, 1993, pp. 159–171.

[10] V. Nigam, D. Miller, Algorithmic specifications in linear logic with subexponentials, in: A. Porto, F. J. López-Fraguas (Eds.), PPDP, ACM, 2009, pp. 129–140.

[11] J.-M. Andreoli, Logic programming with focusing proofs in linear logic, J. Log. Comput. 2 (3) (1992) 297–347.

[12] K. Watkins, I. Cervesato, F. Pfenning, D. Walker, A concurrent logical framework I: Judgments and properties, Tech. Rep. CMU-CS-02-101, Carnegie Mellon University, revised, May 2003 (2003).

[13] V. Nigam, D. Miller, A framework for proof systems, J. Autom. Reasoning 45 (2) (2010) 157–188.

[14] V. Nigam, C. Olarte, E. Pimentel, A general proof system for modalities in concurrent constraint programming, in: CONCUR, Vol. 8052 of LNCS, Springer, 2013, pp. 410–424.

[15] C. Olarte, E. Pimentel, V. Nigam, Subexponential concurrent constraint programming, Theoretical Computer Science 606 (1) (2015) 98–120.

Logical Games for Minimal Logic

Alexandra Pavlova[1]

[1]Saint Petersburg State University, Russia

Keywords:

Dialogue logic, Sequent calculi, Game-Theoretical Semantics

By logical games here we understand two types of games: Dialogue logic of Paul Lorenzen and Kuno Lorenz and Game-Theoretical Semantics (GTS) proposed by Jaakko Hintikka and developed by Gabriel Sandu. Dialogue logic and Game-Theoretical Semantics (GTS) are believed to define different types of truth: the former establishing validity and the later handling truth in a model [9][8]. However, there has been shown a correspondence between those two types of games stating the existence of an algorithm permitting us to transform a winning strategy for Eloise in Game-Theoretical semantics into the corresponding one for the Proponent in a dialogue with hypotheses [9] and visa versa. However, aimed at achieving this result some changes were proposed for the intuitionistic and classical dialogues as defined in [6][7] adjusting them to a model.

Apart from those results, a major work has been done to prove correspondence between dialogue games and sequent calculi. Several authors proposed their proves for the intuitionistic dialogues and the corresponding intuitionistic validity, such as Fermüller [3], Felscher [2], Sørensen and Urzyczyn [10]. Recently there has been proposed an elegant version of proof for both intuitionistic and classical logic by Alama, Knoks and Uckelman [1]. They used a variant of the sequent calculus system $GKcp$ [12] for classical propositional logic.

In this paper we define a class of dialogue games for minimal logic and a corresponding sequent calculus. Minimal propositional logic can be obtained by rejecting not only the classical law of excluded middle (as intuitionistic logic does), but also the principle of explosion (ex falso quodlibet) $A, \neg A \vdash B$, where B is arbitrary. Thus, we define a sequent calculus for minimal logic as an intuitionist calculus (like LJ of Genzen) but without the right weakening (WR) of the form:

$$\frac{\Gamma \to \varnothing}{\Gamma \to \mathfrak{D}} \ (WR\varnothing)$$

where $\mathfrak{D}$ is an arbitrary formula. It is easy to see that this rule corresponds to the Genzen NJ rule of the form: $\frac{\bot}{\mathfrak{D}}$, as $\Gamma \to \varnothing$ correspond to the $\bot$. We can provide a simple exemple of the theorem $A \supset (\neg A \supset B)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \to A}{\neg A, A \to}\,(\neg L)}{\neg A, A \to B}\,(WR\varnothing)}{A \to \neg A \supset B}\,(\to R)}{\to A \supset (\neg A \supset B)}\,(\to R)}$$

Then we define a minimal dialogue game as an intuitionistic game (by the intuitionistic dialogue game we understand the one where there is a rule restricting defenses of the players as follows: "***D11*** *If it is X's turn and there are more than one attack by Y that X has not yet defended, only the most recent one may be defended*") where the Proponent cannot leave any attack of the Opponent without a defense. There is only one exception represented by the attack on the negation because there is no way to perform a defense against this attack. We provide an exemple of the same formula as in the sequent calculus $A \supset (\neg A \supset B)$:

Round	**Opponent**	**Proponent**
0		(1) $A \supset (\neg A \supset B)$
I	(2) A	(3) $\neg A \supset B$
II	(4) $\neg A$	
III		(5) A

Then we come up with a proof of the correspondence between the winning strategies for the Proponent in that class of games and the validity in minimal propositional logic. In our proof we use a modified version of Kleene intuitionistic system G_3 [5] without structural rules.The axiom now has the following form: $A, \Gamma \to \Theta, A$. Furthermore, the inference rules are modified in such a way that we keep the main formulae, for instance:

$$\dfrac{\dfrac{\dfrac{\dfrac{A, A \wedge B \to \neg(A \wedge B), A}{\neg A, A, A \wedge B \to \neg(A \wedge B)}\,(\neg L)}{\neg A, A \wedge B \to \neg(A \wedge B)}\,(\wedge L_1)}{\neg A \to \neg(A \wedge B)}\,(\neg R)}{\to \neg A \supset \neg(A \wedge B)}\,(\to R)$$

Finally, as there has been established a correspondence between game-theoretical semantics and dialogue games with hypothesis, then various sequent calculi (with non-empty antecedent as a set of initial hypothesis specifying a domaine used in the game-theoretical semantics) may encode winning strategies for Eloise in different types of games within the Game theoretical semantics.

Acknowledgements

The research is supported by the Russian Foundation for Humanities, project 15-23-01002. We would also like to thank Iouri Netchitailov and Elena Lisanyuk for fruitful discussions and help.

References

[1] Alama J., Knoks A., and Uckelman S. L.: *Dialogue games for classical logic*, in TABLEAUX 2011: Workshops, Tutorials, and Short Papers, Technical Report IAM-11-002 (Universität Bern): pp. 82–86

[2] Felscher W.: *Dialogues, strategies, and intuitionistic provability.* Annals of Pure and Applied Logic 28, pp. 217-254 (1985)

[3] Fermüller C. G.: *Parallel dialogue games and hypersequents for intermediate logics.* In: Mayer, M.C., Pirri, F. (eds.) TABLEAUX 2003. pp. 48-64 (2003)

[4] Hintikka J.: *The Principles of Mathematics Revisited*, Cambridge University Press, Cambridge (1996)

[5] Kleene S. C.: *Introduction to Metamathematics*, the Netherlands (1952)

[6] Krabbe E. C. W.: *Dialogue logic. Handbook of the History of Logic*, Volume 7, pp. 665-704 (2006)

[7] Lorenzen P. and Lorenz K.: *Dialogische Logik.* Wissenschaftlische Buchgesellschaft, Darmstadt (1978)

[8] Pavlova A. M.: *Truth in Dialogue Logic and Game-Theoretical Semantics (GTS) // Logical Investigations.* vol. 21, no 2, pp. 107–133 (2015)

[9] Rahman Sh., Tulenheimo T.: *From Games to Dialogues and Back: Towards a General Frame for Validity.* In *Games: Unifying logic, Language and Philosophy.* Springer (2009)

[10] Sørensen, M.H., Urzyczyn, P.: *Sequent calculus, dialogues, and cut elimination.* In: Reflections on Type Theory, λ-Calculus, and the Mind, pp. 253-261. Universiteit Nijmegen (2007)

[11] Stegmüller W.: *Remarks on the completeness of logical systems relative to the validity-concepts of P. Lorenzen and K. Lorenz.* Notre Dame J. Formal Logic Volume 5, Number 2, pp. 81-112 (1964)

[12] Troelstra, A.S., Schwichtenberg, H.: *Basic Proof Theory.* Cambridge University Press, 2nd edn. (2000)

Logics of deceit and outsmarting (with pictures of computers)

Duško Pavlović

University of Hawaii at Manoa, USA

Just like the logic of science is driven by the fact that "we can never be sure when we are right, only when we are wrong" (as Richard Feynman put it), the logic of security is driven by the fact that we can never be sure when we are secure, only when we are under attack. Science and security therefore evolve through similar logical processes, which I shall discuss in the first part of the talk.

The main difference between science and security is that science is a game that people play against nature, whereas security is a game of people against people. While nature does not change its behaviors whenever a scientific theory makes them predictable, the adversaries keep changing their behaviors in order to *outsmart* each other: they try not only to predict each other's behaviors, but also to deceive and mislead each other's predictions. They learn to adapt their strategies, and to disturb the opponents' strategies.

Although there is a vast literature about strategic learning, such outsmarting games have largely remained beyond reach, as the game theoretic analyses had to limit their scope to adaptive learning of opponent's non-adaptive strategies. The reason for that limitation is that outsmarting, as adaptive learning of adaptive strategies, involves algorithms that are capable to learn the behaviors of other algorithms. The standard low-level models of computation are not suitable for such applications: a Turing machine that learns the behaviors of other Turing machines, or a lambda-term that captures other lambda-terms, are important theoretical constructs, but they are clumsy as programming tools. In the second part of the talk, I will describe the *monoidal computer*, a high-level graphic model of computation, that supports at least some forms of reasoning about outsmarting. If the time permits, I will present a computational explanation of an apparent algorithmic paradox at the heart of arguably the best adaptive strategy evolved so far.

A simple method of proving logical constancy by consequence extraction

Tin Perkov

Polytechnic of Zagreb, Croatia

Keywords:

logical constants, consequence extraction

A generalization of a method for showing that symbols such as boolean connectives, (generalized) quantifiers or modal operators are logical constants is attempted in this work. The idea of consequence extraction as presented by Bonay and Westerståhl in [2] is used for this purpose.

There are several approaches to the following foundational question: which symbols of a formal language are logical? One notion of logical constant is that, given a language and a consequence relation, a symbol is a constant if replacing it with another symbol of the same type destroys at least one valid inference of that consequence relation (see [2]).

One technical problem with this approach is that there is often only one symbol of a given type, so we have nothing to replace it with in order to test its constancy. This can be solved by introducing a new symbol which does not essentially change the language. In an example given in [2], the negation, which is the only unary connective in many familiar languages, is replaced with another unary connective defined as "equal to false" that is added to the language to prove that $\neg$ is a constant.

Introducing new symbols depends, however, on the nature of a particular language. This is probably easy to do from case to case, but it is attempted here to give a general method that would work in any language in essentially the same way. First consider few motivating examples.

- Consider the basic propositional modal language with the usual (local) logical consequence relation $\Vdash_{ML}$, as defined in [1]. We expect modal operator $\Diamond$ to be a logical constant, but in the definition of the basic modal language it is the only symbol of its kind. However, a symbol $\Box$ is used as an abbreviation for dual $\Box\varphi \equiv \neg\Diamond\neg\varphi$, so we can include it in the language as another modal operator. Now, from the duality itself we have, for instance, that $\Box p \Vdash_{ML} \neg\Diamond\neg p$ is a valid inference, but if

we replace $\Box$ with $\Diamond$ we get $\Diamond p \Vdash_{ML} \neg\Diamond\neg p$, which is easily verified not to be valid.

- The disjunction $\vee$ is proved in [2] to be a logical constant of the propositional logic using a classic example that $p \models_{PL} p \vee q$ but $p \not\models_{PL} p \wedge q$. Note that here the constancy of $\vee$ is also proved by using its dual. The conjunction $\wedge$ is dual to $\vee$ in the same way $\Diamond$ is to $\Box$: we have $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$.

- The universal quantifier for the first-order logic is dual to the existential quantifier. Similarly as in the first example, we have a valid inference $\forall x A \models_{FO} \neg\exists x \neg A$, but $\exists x A \not\models_{FO} \neg\exists x \neg A$. Therefore, $\forall$ is a logical constant.

These examples lead to the following very simple, but fairly general idea: for any symbol S such as logical connective, quantifier or operator, we define the dual S' in a similar way. Given a consequence relation $\Rightarrow$, duality means that we have valid inferences of the form $S'(\varphi_1, \varphi_2, \dots) \Rightarrow \neg S(\neg\varphi_1, \neg\varphi_2, \dots)$ and $\neg S(\neg\varphi_1, \neg\varphi_2, \dots) \Rightarrow S'(\varphi_1, \varphi_2, \dots)$. If S is not self-dual, then at least one of these inferences fails if we replace S with S'. Therefore, S is a logical constant.

It is rather obvious that any missing dual can always be included in a language by defining it as an abbreviation and then adding it to the list of symbols, and that this way the language stays essentially the same. So, for any symbol that is used inductively in building formulas in a way that the truth of these formulas depends on the truth of one or more (depending on arity) formulas in the scope of that symbol, we have the dual symbol. By the reasoning presented above, it is immediately verified that any such symbol, if it is not self-dual, is a logical constant for a given language and consequence relation. This includes seemingly vast majority of symbols we have in mind when trying to generalize the notion of logical constants, like logical connectives $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$ etc., quantifiers $\forall$, $\exists$ (also as second-order quantifiers, even polyadic) and many more, modal operators $\Diamond$, $\Box$ and so on.

However, there are some examples of self-dual symbols which are also considered to be logical constants, notably the negation itself ($\neg p \Leftrightarrow \neg\neg\neg p$). In such cases we have to use some other symbol of the same type to prove the constancy, as Bonay and Westerståhl did in the case of $\neg$. An example of a self-dual generalized quantifier is "more then a half of", if interpreted on a finite set of odd cardinality. As a general method, in such case we can use any symbol of the same type that is not self-dual, thus destroying at least one of the inferences which express self-duality. So for example, replacing a self-dual quantifier with $\exists$ proves that it is a constant. If a self-dual symbol is unique of its type, we add some symbol that is not self-dual as an abbreviation.

Finally, consider 0-ary symbols. For example, predicates or relational symbols in first-order logic, propositional variables in propositional logic or modal logic and so on, are not generally considered to be logical constants.[1] But truth values like $\top$ and $\bot$, if included in a language, are considered logical constants. And rightly so, because we have $\top \Rightarrow \neg\bot$, but $\bot \not\Rightarrow \neg\bot$. We can consider $\top$ and $\bot$ dual to each other (there just isn't anything in their scope to negate, so dual is simply the negation).

References

[1] P. Blackburn, M. de Rijke, Y. Venema, *Modal Logic*, Cambridge University Press, 2001.

[2] D. Bonay, D. Westerståhl, *Consequence mining*, Journal of Philosophical Logic 41, pp. 671–709, 2012.

[1] However, some of them, like equality, may be considered constants if we fix a (normal) interpretation of the symbol.

The sure thing principle and Simpson paradox

Zvonimir Šikić

University of Zagreb, Croatia

We present Blyth argument (Blyth 1972) that Simpson reversals (Yule 1903; Simpson 1951) prove that the sure thing principle is not valid.

We show, contra (Pearl 2015), that Simpson reversals do not necessarily involve causality and that there is a purely probabilistic and true version of the sure thing principle with no causalities involved.

We offer a general argument, contra (Bandyopadhyay 2011), that Simpson reversals are not surprising (paradoxical) and apply it to the resolution of the concrete example from a Mathematical Olympiad.

We argue that people generalize from disjunctive syllogism (which is valid) to the sure thing principle (which is not valid), because they generalize from conditional $\rightarrow$ to support $\uparrow$.

Namely, the basic properties of supports $A \uparrow B$, defined as $pr(B|A) > pr(B)$, are exactly the opposite to those of conditionals $A \rightarrow B$, as proved in (Šikić 2016).

References

[1] P. S. Bandyoapdhyay , D. Nelson, M. Greenwood, G. Brittan, J. Berwald (2011), The logic of Simpson's paradox, Synthese, 181/2, 185-208.

[2] Blyth, C.(1972), On Simpson's paradox and the sure-thing principle. Journal of the American Statistical Association 67,364-366.

[3] Yule, G. (1903), Notes on the theory of association of attributes in statistics. Biometrika 2, 121-134.

[4] Pearl, Judea (2015),The sure-thing principle, UCLA Cognitive Systems Laboratory, Technical Report R-466.

[5] Pearl, J. (2009), Causality: Models, Reasoning, and Inference. 2nd ed. Cambridge University Press, New York.

[6] Savage, L. J. (1954), The foundations of statistics. John Wiley & Sons Inc., New York.

[7] Simpson, E. (1951), The interpretation of interaction in contingency tables. Journal of the Royal Statistical Society, Series B 13, 238-241.

[8] Šikić, Z.(2016), On probable conditionals, European Journal of Analitic Philosophy, to appear.

Remarks on Reversible Circuit Synthesis from Decision Diagrams

Suzana Stojković[1], Milena Stanković[1], Claudio Moraga[2] and Radomir S. Stanković[1]

[1]Dept. of Computer Science, Faculty of Electronics, Niš, Serbia
[2]TU Dortmund University, 44221 Dortmund, Germany

Decision diagrams are a data structure enabling compact representations of large Boolean functions. This feature is used to design reversible circuits for functions of a large number of variables, the term scalability is often used in this context.

A decision diagram consists of non-terminal nodes and constant nodes connected by edges. To each non-terminal node a variable in the function is assigned and called the decision variables. A function is assigned to a diagram by decomposition rules, which are defined in terms of variables and subfunctions, i.e., co-factors, determined with respect to the variables.

In classical circuit synthesis from decision diagrams, the circuit is produced by replacing each non-terminal node with a module realizing the decomposition rule assigned to the node, and providing interconnections corresponding to the edges connecting the nodes. In this way, multi-level circuits are produced with each level in the circuit corresponding to a level in the diagram and preserving the hierarchical structure of the diagram. The main difference in reversible circuit synthesis compared to the classical synthesis, is that the hierarchical tree-like structure of a decision diagram is mapped into the linear structure of a cascade. This mapping is preformed by the so-called post-order traversal of the diagram, meaning that the left node is visited first, then the right node, and after that their root node. Each non-terminal node starting from the bottom left corresponds to a level in the cascade. Mapping the tree structure into the linear structure requires that all the information in a level of the cascade has to be preserved for other levels. This requires to introduce additional (ancilla) lines, which can be considered as a drawback of this synthesis method.

To overcome it, research was done towards reducing the ancilla lines, but the problem is that the resulting reversible circuits usually have a large quantum cost. Ancilla-free synthesis based on BDDs is scalable, however, the price is high quantum cost [1], [7], [8], [11], [17].

Majority of decision diagram based design methods concern with reducing the number of nodes, i.e., the size of a decision diagrams, as a way to simplify the circuits. This can be achieved by using conventional methods for reduction of the size of diagrams, including variable ordering, exploiting negated edges, considering diagrams defined in terms of various decomposition rules, the usage of Free BDDs, linearization of BDDs, and other related optimization methods [2], [4], [5], [18].

In this respect, Kronecker binary decision diagrams (KBDDs) are proposed as a most efficient data structure among various decision diagrams for reversible circuit synthesis, since usually have the minimum number of non-terminal nodes, however, improvements are possible regarding the cost of the produced reversible circuits by taking into account the cost of modules realizing non-terminal nodes [3], [6], [16].

KBDDs use either the Shannon, positive, and negative Davio rules, under the restriction that the same rule is assigned to all the nodes at a level in the diagram. By using a suitable combination of nodes for a given function f, the number of non-terminal nodes in KBDD is reduced comparing with other decision diagrams for f. Since the cost of the Shannon nodes is larger than that of Davio nodes, it might happen that the total cost of the resulting circuit is larger than that of circuits produced from Functional decision diagrams (FDDs).

Further improvement can be achieved when the number of nodes in FDDs is reduced by selecting between the positive and the negative Davio nodes, i.e., by using FPFDDs [9], [15].

Recall that finding an exact minimum FPFDD requires 2^n checks, compared to 3^n checks in KFDDs. In general, FDD are suitable for reversible circuit synthesis by Toffoli gates for two reasons

1. The analytical description of Davio nodes corresponds to the expression defining the Toffoli gates.

2. Reduction rules adapted to the Davio nodes often reduce the amount of information that has to be transferred between levels in a linear circuit, which means reduction of lines.

These considerations for FDDs and Toffoli gates, motivated to discuss synthesis of reversible circuits with Hadamard gates based on Walsh decision diagrams. We show by experiments that, as in the case of FDDs, by pairing the functional description of reversible circuits with decomposition rules in used decision diagrams leads to reversible circuits with good performances.

References

[1] Al-Rabadi, A.N., *Reversible Logic Synthesis: From Fundamentals to Quantum Computing*, Springer Science & Business Media, 2012.

[2] Chattopadhyay, A., Littarru, A., Amarú, L., Gaillardon, P.-E., De Micheli, G., "Reversible logic synthesis via Biconditional binary decision diagrams", *Proc. 45th IEEE Int. Symp. on Multiple-valued Logic*, 2015, 2-7.

[3] Drechsler, R., Wille, R., "Synthesis of reversible circuits using decision diagrams" (invited paper), *Int. Symp. on Electronic System Design (ISED 2012)*, Kolkata, India, December 19-22, 2012, 1-5.

[4] Feinstein, D. Y., Thornton, M.A., "Reversible logic synthesis based on decision diagram variable ordering", *Journal of Multiple-Valued Logic & Soft Computing*, Vol. 19, No. 4, 2012, 325-339.

[5] Kerntopf, P., Perkowski, M., *Function-driven Linearly Independent Expansions of Boolean Functions and Their Application to Synthesis of Reversible Circuits*, Portland State University PDXScholar Electrical and Computer Engineering Faculty Publications and Presentations, No. 5-2003.

[6] Lin, C.C., Jha, N.K., "RMDDS: Reed-Muller decision diagram synthesis of reversible logic circuits", *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, Vol. 10, No. 2, 2014, Article No. 14.

[7] Miller, D. M., Thornton, M. A., "QMDD: A decision diagram structure for reversible and quantum circuits", *36th Int. Symp. on Multiple-Valued Logic*, May 17-20, 2006, 30.

[8] Nath, J., Nath, A., "Application of decision diagrams to desing quantum logic circuits", *Journal of Global Research in Computer Science*, Vol. 3, No. 3, 2012,

[9] Pang, Y., Yan, Y., Lin, J., Huang, H., Wu, W., "An efficient method to synthesize reversible logic by using Positive Davio decision diagrams", *Circuits, Systems, and Signal Processing*, Vol. 33, No. 10, 2014, 3107-3121.

[10] Schönborn, E., Datta, K., Wille, R., Sengupta, I., Rahaman, H., Drechsler, R., "Optimizing DD-based synthesis of reversible circuits using negative control lines", *Proc. 17th Int. Symp. on Design and Diagnostics of Electronic Circuits & Systems*, Warsaw, Poland, April 23-25, 2014, 129-134.

[11] Soeken, M., Tague, L., Dueck, G.W., Drechsler, R., "Ancilla-free synthesis of large reversible functions using binary decision diagrams", *Journal of Symbolic Computation*, 73, 2016, 1-26.

[12] Soeken, M., Wille, R., Drechsler, R., "Hierarchical synthesis of reversible circuits using positive and negative Davio decomposition", *Proc. 5th Int. Design and Test Workshop (IDT)*, Abu Dhabi, December 14-15, 2010, 143-148.

[13] Soeken, M., Wille, R., Minato, S., Drechsler, R., "Using πDDs in the design of reversible circuits (work-in-progress)", in R. Gluck, Yokoyama, T., (Eds.), *Reversible Computation 4th International Workshop*, July 2-3, 2012 (RC 2012), Copenhagen, Denmark, RC 2012, LNCS 7581, Springer, 2013, 197-203.

[14] Soeken, M., Wille, R., Hilken, C., Przigoda, N., Drechsler, R., "Synthesis of reversible circuits with minimal lines for large functions", *Asia and South Pacific Design Automation Conference*, 2012, 85-92.

[15] Stojković, S., Stanković, M., Moraga, C., "Complexity reducton of Toffoli networks besed on FDD", *Facta Universitatis, Ser. Electronics and Energetics*, Vol. 28, No. 2, 2015, 251-262.

[16] Wang, Y.-R., Shen, X.-K., Zhou, Y.-H., "Synthesis design method of reversible logic circuit based on Kronecker functional decision diagram", *Chinese Journal of Electronics*, Vol. 42, No. 5, 2014, 1025-1029.

[17] Wille, R., Drechsler, R., "BDD-based synthesis of reversible logic for large functions", *Proc. 46th Design Automation Conference*, 2009, 270-275.

[18] Wille, R., Drechsler, R., "Effect of BDD optimization on synthesis of reversible and quantum logic", *Electronic Notes in Theoretical Computer Science*, Vol. 253, No. 6, 2010, 57-70.

A Logic for Temporal Beliefs and Intentions– Completeness and Belief revision

Marc van Zee
University of Luxembourg
Rue Richard Coudenhove-Kalergi 6, L-1359 Luxembourg
marc.vanzee@uni.lu
Dragan Doder
Univerzitet u Beogradu
Mašinski fakultet
Kraljice Marije 16, 11000 Beograd, Srbija
ddoder@mas.bg.ac.rss

August 24, 2016

We introduce a logic for temporal beliefs and intentions based on Shoham's database perspective [1]. For the logic, we develop strongly complete axiomatization. We formalize Shoham's coherence conditions on beliefs and intentions. In order to do this we separate strong beliefs from weak beliefs. Strong beliefs are independent from intentions, while weak beliefs are obtained by adding intentions to strong beliefs and everything that follows from that. We provide AGM-style postulates for the revision of strong beliefs and intentions: strong belief revision may trigger intention revision, but intention revision may only trigger revision of weak beliefs. After revision, the strong beliefs are coherent with the intentions. We show in a representation theorem that a revision operator satisfying our postulates can be represented by a pre-order on interpretations of the beliefs, together with a selection function for the intentions.

Acknowledgements

This work was supported by the Serbian Ministry of Education and Science through project ON174026, and by the National Research Fund (FNR) of Luxembourg through project RationalArchitecture and PRIMAT.

References

[1] Yoav Shoham. Logical theories of intention and the database perspective. Journal of Philosophical Logic, 38(6), 633–647 (2009).

Generalized Veltman models

Mladen Vuković

University of Zagreb, Croatia

Keywords:

interpretability logics, generalized Veltman semantics, bisimulation

The language of the interpretability logic IL contains propositional letters p_0, $p_1, \ldots$, the logical connectives $\wedge$, $\vee, \rightarrow$, $\leftrightarrow$ and $\neg$, the unary modal operator $\Box$ and the binary modal operator $\triangleright$. The paper [3] provides the necessary definitions and detailed explanation on IL.

There are several kinds of semantics for the system IL. The basic semantics is Veltman models. Generalized Veltman models were defined by D. de Jongh. We use generalized Veltman models in [5] to prove independences between principles of interpretability. E. Goris and J. Joosten [1] also define and use a kind of generalized Veltman models. We define a notion of bisimulation between two generalized Veltman models in [6], and prove Hennessy–Milner theorem for generalized Veltman semantics. We study various kinds of bisimulations of generalized Veltman models in [4]. We prove in [7] that there is a bisimulation between Veltman model and generalized Veltman model. The existence of a bisimulation in general setting is an open problem. T. Perkov and M. Vuković in [2] use generalized Veltman models in proofs of finite model properties for different interpretability logics. One can naturally pose the question on connection between different kinds of models for interpretability logics. We compare different kinds of generalized Veltman models.

References

[1] E. Goris, J. Joosten, *A new principle in the interpretability logic of all reasonable arithmetical theories*, Logic Journal of the IGPL 19 (2011) 1–17

[2] T. Perkov, M. Vuković, *Filtrations of Generalized Veltman models*, Mathematical Logic Quarterly, to appear

[3] A. Visser, *An overview of interpretability logic*, In: K. Marcus (ed.) et al., Advances in modal logic. Vol. 1. Selected papers from the 1st international workshop (AiML'96), Berlin, Germany, October 1996, Stanford, CA: CSLI Publications, CSLI Lect. Notes. 87(1998), 307–359

[4] D. Vrgoč, M. Vuković, *Bisimulations and bisimulation quotients of generalized Veltman models*, Logic Journal of the IGPL, 18(2010), 870–880

[5] M. Vuković, *The principles of interpretability*, Notre Dame Journal of Formal Logic, 40(1999), 227–235

[6] M. Vuković, *Hennessy–Milner theorem for interpretability logic*, Bulletin of the Section of Logic, 34(2005), 195–201

[7] M. Vuković, *Bisimulations between generalized Veltman models and Veltman models*, Mathematical Logic Quarterly, 54(2008), 368–373

Asterix calculus - classical computation in detail

Dragiša Žunić[1] and Pierre Lescanne[2]

[1]Carnegie Mellon University, Doha, Qatar
[2]Ecole Normale Supérieure de Lyon, France

Keywords:

Classical logic, classical computation, Asterix calculus, ${}^*\mathcal{X}$ calculus, structural rules, sequent calculus

We present Asterix calculus (also denoted ${}^*\mathcal{X}$ calculus), built from names instead of variables. Asterix is designed to stand in computational correspondence with classical logic represented in the sequent calculus. More precisely, in the sequent system $G1$ [1], featuring explicit structural rules weakening and contraction.

It is possible to define many variants of Gentzen sequent systems. The basic Genzen systems for classical and intuitionistic logic denoted as $G1, G2$ and $G3$ are formalized in [2] and later revisited in [1]. In brief, the essential difference between $G1$ and $G3$ is the presence or absence of explicit structural rules. The distinguishing point of $G2$ is the use of the so-called mix instead of a cut rule.

In the context of the Curry-Howard paradigm, we have the following correspondence between classical logic's system $G1$ and ${}^*\mathcal{X}$-terms:

$$\begin{array}{rcl} \textit{Proofs} & \Leftrightarrow & \textit{Terms} \\ \textit{Propositions} & \Leftrightarrow & \textit{Types} \\ \textit{Cut ellimination} & \Leftrightarrow & \textit{Reduction} \end{array}$$

Having explicit terms for weakening and contraction at hand is an advantage strategically speaking. On the one hand we reveal the computational role of these constructors (erasure and duplication, respectively).

On the other hand, having these terms explicit, and thus a very fine grained calculus, we can identify which syntactically different terms (proofs) should be considered the same; by providing equations identifying terms up-to trivial rules-permutation.

Of course the calculus retains the desirable properties of its predecessors: type preservation, linearity preservation, strong normalisation of typed terms.

Besides Asterix ($^*\mathcal{X}$) [3, 4] there is also Obelix ($\mathcal{X}$ calculus) [5, 6]. Informally speaking, these calculi are classical analogues of intuitionistic λlxr, featuring explicit substitution, weakening and contraction [7], and λx, featuring explicit substitution [8], respectively.

Acknowledgements

This publication was made possible by NPRP grant NPRP 7-988-1-178 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] A. S. Troelstra, H. Schwichtenberg, *Basic Proof Theory*, Cambridge University Press, 1996.

[2] S. Kleene, *Introduction to Metamathematics*, North Holland, 1952.

[3] S. Ghilezan, P. Lescanne and D. Zunic, *Computational interpretation of classical logic with explicit structural rules*, draft, 2012.

[4] D. Žunić, *Computing With Sequent and Diagrams in Classical Logic - Calculi $^*\mathcal{X}$, $©\mathcal{X}$ and $^d\mathcal{X}$*, ENS Lyon (PhD thesis), 2007.

[5] C. Urban, *Classical Logic and Computation*, University of Cambridge (PhD thesis), 2000.

[6] S. van Bakel, S. Lengrand, P. Lescanne, *The language $\mathcal{X}$: circuits, computation and Classical Logic*, Proc. 9th Italian Conf. on Theoretical Computer Science, vol. 3701, pp. 81-96, 2005.

[7] D. Kesner, S. Lengrand, *Ressource operators for lambda-calculus*, Information and Computation, vol. 205, pp. 419-473, 2007.

[8] F. Barbanera, S. Berardi, *A Symmetric Lambda Calculus for "Classical" Program Extraction*, TACS, pp. 495-515, 1994.