Truely Concurrent Processes in the Calculus of Structures

Ross Horne

School of Computer Science and Engineering, Nanyang Technological University, Singapore

The calculus of structures [5] is sufficiently expressive to define extensions of linear logic, such as BV, featuring a self-dual non-commutative operator, *seq*. Such a self-dual operator can be used to directly model sequentiality in processes, i.e., when one event happens before another. I summarise advances in process modelling using the calculus of structures and propose some open problems.

The first problem when it comes to modelling processes is whether expressive process models can be captured in extensions of BV. Preliminary work [2] embedded a fragment of CCS with only prefix and parallel composition in BV. BV has also been used to model causality for quantum processes [1]. More recent work has considered the following extensions.

- While multiplicatives model parallelism, additive operators can be used to model various forms of non-deterministic choice. MAV [7] (BV extended with additives) has been used to model finite session types in the language Scribble [6] such that linear negation captures duality of session types. Cut-elimination, leads us to a *sub-type* system and a *multi-party compatibility* result [3], which uses provability to establish when a protocol choreography can be synthesised from knowing only local behaviours of each participant. Another fragment of MAV has been used to model *attack trees* where ordering between events are significant [8]. Linear implication is shown to respect certain quantitative properties of attack trees, including timing properties.

- The internal π-calculus can be embedded in an extension of MAV with a pair of de Morgan dual nominal quantifiers И and Э, called MAV1 [9]. Furthermore, by including first-order quantifier $\forall$ and $\exists$, the π-calculus with unrestricted inputs can be captured. Note, by using a de Morgan dual pair of nominals we avoid the problem that an embedding using a self-dual nominal quantifier [11] is unsound. Thus a larger fragment of CCS, including name restriction, can also be soundly modelled using the system MAV1, involving the pair of nominal quantifiers.

- Between the additive operators there are more controlled versions of the additive operators. By weighting with probabilities these "sub-additive" operators, $+_p$ and $\sqcup_p$ can be used to model processes with probabilistic choice. The probabilistic and non-deterministic additives coexist, hence mixed probabilistic/non-deterministic choice can be captured. Note this extension with probabilistic operators has not yet appeared in print.

$$
\begin{array}{cccc}
(a) & P \oplus Q & \qquad (b) & \exists x.P \\
 & | & & | \\
 & P \sqcup_p Q & & \text{Э}x.P \\
 & | & & | \\
 & P +_p Q & & \text{И}x.P \\
 & | & & | \\
 & P \,\&\, Q & & \forall x.P
\end{array}
$$

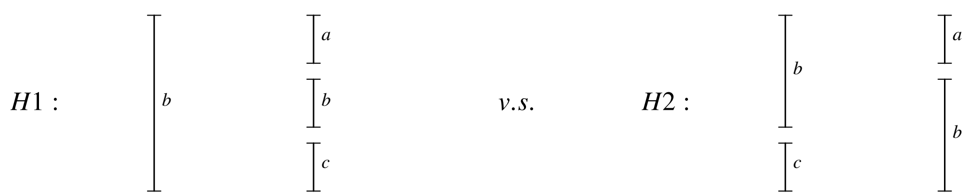
Figure 1: Relationships between operators in extensions of BV: (a) additives, (b) first-order quantifiers. Observe the nominal quantifiers and probabilistic sub-additives sit between the established operators.

The above extensions and applications demonstrate that, thanks the expressive power offered by the calculus of structures, powerful process models can be directly embedded in logical systems. An advantage of this approach is process embeddings can be directly compared using linear implication, $\multimap$, defined, as usual, in terms of linear negation and multiplicative disjunction. As a consequence of cut elimination, linear implication is a pre-order that holds in every (positive) context, hence is a notion of processes refinement. The subtle question is: which notion of refinement?

Previous work [8] has proven linear implication is sound with respect to pomset ideals [4]. Pomsets define a truly concurrent (non-interleaving) semantics for processes, meaning parallel composition cannot be reduced to interleaving. For example, processes such as $a \mathbin{;} a$ and $a \parallel a$ are distinguished. Pomset ideals are a permissive approach to pomsets where additional causal dependencies may be added, but never removed during refinement. Linear implication also define a truly concurrent semantics, for example, in BV, $a \mathbin{;} a \multimap a \parallel a$ holds but not the converse.

Linear implication is sound, but not complete with respect to pomset ideals, since linear implication exhibits branching-time properties. For example $a \mathbin{;} (b \oplus c)$ is not equivalent to $(a \mathbin{;} b) \oplus (a \mathbin{;} c)$, although the one-way implication $(a \mathbin{;} b) \oplus (a \mathbin{;} c) \multimap a \mathbin{;} (b \oplus c)$ does hold, indicating that non-determinism can be reduced during refinement. A paper, under review for a journal[1], proves that linear implication is sound with respect to weak simulation, by showing a weak simulation can be constructed from any implication. This still leaves open the question of whether a notion of refinement, with both branching-time and non-interleaving properties, coincides with linear implication.

Future perspectives. Further to the above established results, I will present some interesting problems for sharpening further the above results. ST-simulation [12], an abstraction of real-timed process semantics, appears to be the tightest match for linear implication in the literature. In ST-simulation, instead of assigning concrete timing information to events, an event is split into a start and terminate event. Care is taken in ST-simulation to avoid terminate events becoming confused. For example, the following two histories can be confused if the terminate events labelled with b are wrongly swapped.



ST-simulation is the coarsest notion of refinement in the literature exhibiting non-interleaving and branching-time properties, similarly to linear implication. Furthermore, as with linear implication, ST-simulation is preserved under action refinement (replacing an atomic event by a process) — no coarser notion of refinement satisfies this property. What is intriguing is cut-elimination, the corner stone of proof theory, leads us to this particular, rather useful, notion of refinement. I present initial findings in this direction. In particular, I propose a graphical generalisation of BV in which more general patterns of causality, such as those in the history $H2$ above can be expressed.

Note the true-concurrency of proof search in deep inference has been previously noted [10], with applications for planning problems. The line of work described in this talk is however the first to formally connect linear implication with established semantics for process calculi.

[1] See http://www.ntu.edu.sg/home/rhorne/piI.pdf supporting a journal submission under review

References

[1] Richard F Blute, Alessio Guglielmi, Ivan T Ivanov, Prakash Panangaden, and Lutz Straßburger. A logical basis for quantum evolution and entanglement. In *Categories and Types in Logic, Language, and Physics*, pages 90–107. Springer, 2014.

[2] Paola Bruscoli. A purely logical account of sequentiality in proof search. In *ICLP*, volume 2401 of *LNCS*, pages 302–316. Springer, 2002.

[3] Gabriel Ciobanu and Ross Horne. Behavioural analysis of sessions using the calculus of structures. In *PSI 2015, 25-27 August, Kazan, Russia*, volume 9609 of *LNCS*, pages 91–106, 2015.

[4] Jay L. Gischer. The equational theory of pomsets. *Theor. Comput. Sci.*, 61:199–224, 1988.

[5] Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1), 2007.

[6] Kohei Honda et al. Scribbling interactions with a formal foundation. In *ICDCIT 2011*, volume 6536 of *LNCS*, pages 55–75. Springer, 2011.

[7] Ross Horne. The consistency and complexity of multiplicative additive system virtual. *Sci. Ann. Comp. Sci.*, 25(2):245–316, 2015.

[8] Ross Horne, Sjouke Mauw, and Alwen Tiu. Semantics for specialising attack trees based on linear logic. *Fundamenta Informaticae*, 153:57–86, 2017.

[9] Ross Horne, Alwen Tiu, Bogdan Aman, and Gabriel Ciobanu. Private names in non-commutative logic. In *CONCUR 2016*, pages 31:1–31:16. LIPIcs, 2016.

[10] Ozan Kahramanoğulları. True concurrency of deep inference proofs. In Jouko Väänänen, Åsa Hirvonen, and Ruy de Queiroz, editors, *Logic, Language, Information, and Computation*, pages 249–264, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[11] Luca Roversi. A deep inference system with a self-dual binder which is complete for linear lambda calculus. *Journal of Logic and Computation*, 26(2):677, 2016.

[12] Rob J. van Glabbeek and Frits W. Vaandrager. Petri net models for algebraic theories of concurrency. In J. W. de Bakker, A. J. Nijman, and Philip C. Treleaven, editors, *PARLE, Parallel Architectures and Languages Europe, Volume II: Parallel Languages, Eindhoven, The Netherlands, June 15-19, 1987, Proceedings*, volume 259 of *Lecture Notes in Computer Science*, pages 224–242. Springer, 1987.