The Proof Technique of Unique Solutions of Contractions

Davide Sangiorgi

The proof technique of unique solutions of contractions*

Davide Sangiorgi[1]

Università di Bologna & INRIA

This extended abstract summarises work conducted with Adrien Durier and Daniel Hirschkoff (ENS Lyon), initially reported in [38].

Bisimilarity is employed to define behavioural equivalences and reason about them. Originated in concurrency theory, bisimilarity is now widely used also in other areas, as well as outside Computer Science. In this work, behavioural equivalences, hence also bisimilarity, are meant to be *weak* because they abstract from internal moves of terms, as opposed to the *strong* ones, which make no distinctions between the internal moves and the external ones (i.e., the interactions with the environment). Weak equivalences are, practically, the most relevant ones: e.g., two equal programs may produce the same result with different numbers of evaluation steps.

In proofs of bisimilarity results, the bisimulation proof method has become predominant, particularly with the enhancements of the method provided by the so called 'up-to techniques' [29]. Among these, one of the most powerful ones is 'up-to expansion and context', whereby the derivatives of two terms can be rewritten using expansion and bisimilarity and then a common context can be erased. Forms of 'bisimulations up-to context' have been shown to be effective in various fields, including process calculi [29, 39, 27], λ-calculi [19, 18, 16, 40], and automata [7, 34].

The landmark document for bisimilarity is Milner's CCS book [21]. In the book, Milner carefully explains that the bisimulation proof method is not supposed to be the only method for reasoning about bisimilarity. Indeed, various interesting examples in the book are handled using other techniques, notably *unique solution of equations*, whereby two tuples of processes are componentwise bisimilar if they are solutions of the same system of equations. This method is important in verification techniques and tools based on algebraic reasoning [32, 33, 2].

Milner's theorem that guarantees unique solutions [21] has however limitations: the equations must be 'guarded and sequential', that is, the variables of the equations may only be used underneath a visible prefix and preceded, in the syntax tree, only by the sum and prefix operators. This limits the expressiveness of the technique (since occurrences of other operators above the variables, such as parallel composition and restriction, in general cannot be removed), and its transport onto other languages (e.g., languages for distributed systems or higher-order languages, which usually do not include the sum operator).

* The authors are partially supported by the ANR project 12IS02001 PACE.

We propose a refinement of Milner's technique in which equations are replaced by special inequations called *contractions*. Intuitively, for a behavioural equivalence $\asymp$, its contraction $\succeq_{\asymp}$ is a preorder in which $P \succeq_{\asymp} Q$ holds if $P \asymp Q$ and, in addition, Q has the *possibility* of being as efficient as P. That is, Q is capable of simulating P by performing less internal work. It is sufficient that Q has one 'efficient' path; Q could also have other paths, that are slower than any path in P. Uniqueness of the solution of a system of contractions is defined as with systems of equations: any two solutions must be equivalent with respect to $\asymp$. The difference with equations is in the meaning of solution: in the case of contractions the solution is evaluated with respect to the preorder $\succeq_{\asymp}$, rather than the equivalence $\asymp$.

If a system of equations has a unique solution, then the corresponding system of contractions, obtained by replacing the equation symbol with the contraction symbol, has a unique solution too. The converse however is false: it may be that only the system of contractions has a unique solution. More important, the condition that guarantees a unique solution in Milner's theorem about equations can be relaxed: 'sequentiality' is not required, and 'guardedness' can be replaced by 'weak guardedness', that is, the variables of the contractions can be underneath *any* prefix, including a prefix representing internal work. (This is the same constraint in Milner's 'unique solution of equations' theorem for *strong* bisimilarity; the constraint is unsound for equations on weak bisimilarity.)

Milner's theorem is not complete for *pure* equations (equations in which recursion is only expressible through the variables of the equations, without using the recursion construct of the process language): there are bisimilar processes that cannot be solutions to the same system of guarded and sequential pure equations. In contrast, completeness holds for weakly-guarded pure contractions. The contraction technique is also *computationally* complete: any bisimulation $\mathcal{R}$ can be transformed into an equivalent system of weakly-guarded contractions that has the same size of $\mathcal{R}$ (where the size of a relation is the number of its pairs, and the size of a system of contractions is the number of its contractions). An analogous result also holds with respect to bisimulation enhancements such as 'bisimulation up-to expansion and context'. The contraction technique is in fact computationally equivalent to the 'bisimulation up-to contraction and context' technique — a refinement of 'bisimulation up-to expansion and context'.

The contraction technique can be generalised to languages whose syntax is the term algebra derived from some signature, and whose semantics is given as an LTS. In this generalisation the weak-guardedness condition for contractions becomes a requirement of *autonomy*, essentially saying that the processes that replace the variables of a contraction do not contribute to the initial action of the resulting expression. The technique can also be transported onto other equivalences, including contextually-defined equivalences such as barbed congruence, and non-coinductive equivalences such as contextual equivalence (i.e., may testing) and trace equivalence [24, 9, 10]. For each equivalence, one defines its contraction preorder by controlling the amount of internal work performed.

Further, a contraction preorder can be injected into the bisimulation game. That is, given an equivalence $\asymp$ and its contraction preorder $\succeq_{\asymp}$, one can define the technique of 'bisimulation up-to $\succeq_{\asymp}$ and context' whereby, in the bisimulation game, the derivatives of the two processes can be manipulated with $\succeq_{\asymp}$ and $\asymp$ (similarly to the manipulations that are possible in the standard 'bisimulation up-to expansion and context' using the expansion relation and bisimilarity) and a common context can then be erased. The resulting 'bisimulation up-to $\succeq_{\asymp}$ and context' is sound for $\asymp$. This technique allows us to derive results for $\asymp$ using the (enhanced) bisimulation proof method, thus transferring 'up-to context' forms of reasoning, originally proposed for labeled bisimilarities and their proof method, onto equivalences that are contextual or non-coinductive.

The contraction technique cannot however be transported onto all (weak) behavioural equivalences. For instance, it does not work in the setting of infinitary trace equivalence (whereby two processes are equal if they have the same finite and infinite traces) [11, 10] and must testing [9]. A discussion on this point is deferred to the concluding section.

An example of application of contractions to a higher-order language, which exploits the autonomy condition, is also reported in [38]

Milner's theorem about unique solution of equations stems from an axiomatisation of bisimulation on finite-state processes [23]. Indeed, in axiomatisations of behavioural equivalences [21, 2], the corresponding rule plays a key role and is called *fixed-point rule*, or *recursive specification principle*; see also [30], for trace equivalence. The possible shapes of the solutions of systems of equations, in connection with conditions on the guardedness of the equations, is studied by Baeten and Luttik [4].

Unique solution of equations has been considered in various settings, including languages, algebraic power series and pushdown automata (see the surveys [17, 26]), as well as in coalgebras (e.g., [20]). These models, however, do not have the analogous of 'internal step', around which all the theory of contractions is built. In functional languages, unique solution of equations is sometimes called 'unique fixed-point induction principle'. See for instance [35], in which the conditions resembles Milner's conditions for CCS, and [15], which studies equations on streams advocating a condition based on the notion of 'contractive function' (the word 'contraction' here is unrelated to its use in our paper).

A tutorial on bisimulation enhancements is [29]. 'Up-to context' techniques have been formalised in a coalgebraic setting, and adapted to languages whose LTS semantics adheres to the GSOS format [5]; see for instance [6], which uses lambda-bialgebras, a generalisation of GSOS to the categorical framework.

Our transporting of the bisimulation proof method and some of its enhancements onto non-coinductive equivalences reminds us of techniques for reducing non-coinductive equivalences to bisimilarity. For instance, trace equivalence on nondeterministic processes can be reduced to bisimilarity on deterministic processes, following the powerset construction for automata [14]; a similar reduction can be made for testing equivalence [8]. These results rely on transformations

of transitions systems, which modify the nondeterminism and the set of states, in such a way that a given equivalence on the original systems corresponds to bisimilarity on the altered systems. In contrast, in the techniques based on contractions the transformation of processes is performed dynamically, alongside the bisimulation game: two processes are manipulated only when necessary, i.e., when their immediate transitions would break the bisimulation game.

In CSP [12], some beautiful results have been obtained in which systems of equations have unique solutions provided their least fixed point (intuitively obtained by infinite unfolding of the equations) does not contain divergent states; see [32, 33]. In CSP the semantics has usually a denotational flavour and, most important, the reference behavioural equivalence, failure equivalence, is divergent sensitive. We are currently trying to compare this kind of techniques, based on divergence, with those based on contractions. We just note here that unique solution of contractions holds in cases where the infinite unfolding of the contractions would introduce divergence.

As for the technique based on equations, so the technique based on contractions is meant to be used in combination with algebraic reasoning, on terms whose behaviour is not finite or finite-state: the recursion on the contraction variables captures the infinite behaviour of terms, and the proof that certain processes are solutions is carried out with pure algebraic reasoning. In comparison with equations, a drawback of unique solution of contractions for an equivalence $\asymp$ is that the solutions are not $\asymp$-interchangeable: it may be that P is solution and Q is not, even though $P \asymp Q$.

The proof of completeness of the 'unique solution of contractions' method with respect to the bisimulation proof method uses the sum operator to express the possible initial actions of a process. We are currently exploring how completeness could be recovered in languages in which the sum operator is missing.

We also plan to explore more in depth the contraction techniques in higher-order languagesss. Such study may shed light on the applicability of up-to context techniques to higher-order languages. In a higher-order language, while there are well-developed techniques for proving that a bisimulation is a congruence [28], up-to context is still poorly understood [19, 18, 16, 40, 27]. For instance, for pure λ-calculi and applicative bisimilarity, the soundness of the full up-to context technique (allowing one to remove any context, possibly binding variables of the enclosed terms) still represents an open problem.

Another setting in which up-to context techniques have been recently applied is that of language equivalence for automata, see e.g., [7, 34]. Our techniques are however for languages with internal moves. In the case of automata, a τ-action could correspond to the empty word, which is absorbed in concatenations of words, in the same way as τ-actions are absorbed in concatenation of traces. Even taking into account the way the empty word (or the empty language) and τ-steps are used, the analogy seems light. It is unclear whether contractions could be useful on automata.

Our original motivation for studying contractions was to better understand 'up-to context' enhancements of the bisimulation proof method and their sound-

ness. More broadly, the goal of the line of work reported is to improve our understanding of bisimilarity and the proof techniques for it, including the possibility of exporting the techniques onto other equivalences.

References

1. S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. *Acta Informatica*, 29:737–760, 1992.
2. J.C.M. Baeten, T. Basten, and M.A. Reniers. *Process Algebra: Equational Theories of Communicating Processes*. Cambridge University Press, 2010.
3. Jos C. M. Baeten, Jan A. Bergstra, and Jan Willem Klop. Ready-trace semantics for concrete process algebra with the priority operator. *Comput. J.*, 30(6):498–506, 1987.
4. Jos C. M. Baeten and Bas Luttik. Unguardedness mostly means many solutions. *Theor. Comput. Sci.*, 412(28):3090–3100, 2011.
5. B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, 1995.
6. F. Bonchi, D. Petrisan, D. Pous, and J. Rot. Coinduction up to in a fibrational setting. Proc. CSL-LICS'14, ACM, 2014.
7. Filippo Bonchi and Damien Pous. Checking nfa equivalence with bisimulations up to congruence. In *Proc. POPL'13*, pages 457–468. ACM, 2013.
8. Rance Cleaveland and Matthew Hennessy. Testing equivalence as a bisimulation equivalence. *Formal Asp. Comput.*, 5(1):1–20, 1993.
9. R. De Nicola and R. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.
10. R.J. van Glabbeek. The linear time—branching time spectrum II. In *Proc. CONCUR '93*, volume 715. Springer, 1993.
11. R.J. van Glabbeek. The linear time—branching time spectrum I. In *Handbook of Process Algebra*, pages 3–99. Elsevier, 2001.
12. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
13. K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 152(2):437–486, 1995.
14. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Boston, MA, USA, 2006.
15. G. Hutton and M. Jaskelioff. Representing Contractive Functions on Streams. Submitted, 2011.
16. V. Koutavas and M. Wand. Small bisimulations for reasoning about higher-order imperative programs. In *Proc. POPL'06*, pages 141–152. ACM, 2006.
17. Michal Kunc. Simple language equations. *Bulletin of the EATCS*, 85:81–102, 2005.
18. S.B. Lassen. Relational reasoning about contexts. In *Higher-order operational techniques in semantics*, pages 91–135. Cambridge University Press, 1998.
19. S.B. Lassen. Bisimulation in untyped lambda calculus: Böhm trees and bisimulation up to context. *Electr. Notes Theor. Comput. Sci.*, 20:346–374, 1999.
20. S. Milius, L. S. Moss, and D. Schwencke. Abstract GSOS rules and a modular treatment of recursive definitions. *Logical Methods in Computer Science*, 9(3), 2013.
21. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

22. R. Milner and D. Sangiorgi. Barbed bisimulation. In *Proc. 19th ICALP*, volume 623 of *LNCS*, Springer Verlag, 1992.
23. Robin Milner. A complete axiomatisation for observational congruence of finite-state behaviors. *Inf. Comput.*, 81(2):227–247, 1989.
24. James H. Morris. *Lambda-Calculus Models of Programming Languages*. Phd thesis MAC-TR-57, M.I.T., project MAC, Dec. 1968.
25. V. Natarajan and Rance Cleaveland. Divergence and fair testing. In *Proceedings of ICALP'95*, volume 944 of *LNCS*, pages 648–659. Springer, 1995.
26. Ion Petre and Arto Salomaa. Algebraic systems and pushdown automata. In *Handbook of Weighted Automata*, EATCS Series, pages 257–289. Springer, 2009.
27. Adrien Piérard and Eijiro Sumii. Sound bisimulations for higher-order distributed process calculus. In *Proc. FOSSACS*, volume 6604 of *LNCS*, pages 123–137. Springer, 2011.
28. Andrew Pitts. Howe's method. In *Advanced Topics in Bisimulation and Coinduction*. Cambridge University Press, 2012.
29. Damien Pous and Davide Sangiorgi. Enhancements of the bisimulation proof method. In *Advanced Topics in Bisimulation and Coinduction*. Cambridge University Press, 2012.
30. Alexander Moshe Rabinovich. A complete axiomatisation for trace congruence of finite state behaviors. In *Proc. 9th MFPS*, volume 802 of *LNCS*, pages 530–543. Springer, 1993.
31. Arend Rensink and Walter Volger. Fair testing. *Information and Computation*, 205:125–198, 2007.
32. A. W. Roscoe. *The theory and practice of concurrency*. Prentice Hall, 1998.
33. A. W. Roscoe. *Understanding Concurrent Systems*. Springer, 2010.
34. Jurriaan Rot, Marcello M. Bonsangue, and Jan J. M. M. Rutten. Coinductive proof techniques for language equivalence. In *Proc. LATA*, volume 7810 of *LNCS*, pages 480–492. Springer, 2013.
35. David Sands. Computing with Contexts: A simple approach. ENTCS, volume 10, Elsevier, 1998.
36. Sangiorgi, D. and Milner, R. The problem of "Weak Bisimulation up to". In *Proc. CONCUR '92*, volume 630 of *LNCS* pages 32–46. Springer, 1992.
37. D. Sangiorgi. Locality and True-concurrency in Calculi for Mobile Processes. In *Proc. TACS '94*, volume 789 of *LNCS*, pages 405–424. Springer, 1994.
38. D. Sangiorgi. Equations, contractions, and unique solutions In *Proc. POPL 15*, pages 421–432. ACM, 2015.
39. D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.
40. Davide Sangiorgi, Naoki Kobayashi, and Eijiro Sumii. Environmental bisimulations for higher-order languages. *ACM Trans. Program. Lang. Syst.*, 33(1):5, 2011.