Up-to Techniques for Branching Bisimilarity

Rick Erkens

Abstract

A notion of bisimilarity is a behavioral equivalence on processes. From the definition of such a notion comes a proof technique that can be used to establish equality between processes. Up-to techniques are improvements of this technique. In this talk I will present a couple of results from my master's thesis [4] that I carried out under supervision of Jurriaan Rot. The main result that we discuss in this talk is that many up-to techniques that have been studied for strong and weak bisimilarity, also work for branching bisimilarity. We will also discuss some future extensions.

1 Branching bisimilarity (up-to)

Branching bisimilarity [7] is a notion of behavioral equivalence on processes that abstracts from internal transitions while preserving the branching structure of processes. We say that a relation on processes $\mathcal{R}$ is a *branching bisimulation* if for all pairs $(P, Q) \in \mathcal{R}$ the following hold:

- for all P' and for all α, if $P \xrightarrow{\alpha} P'$ then there exist Q', Q'' such that $Q \Longrightarrow Q' \xrightarrow{(\alpha)} Q''$ with $P \mathcal{R} Q'$ and $P' \mathcal{R} Q''$; and
- for all Q' and for all α, if $Q \xrightarrow{\alpha} Q'$ then there exist P', P'' such that $P \Longrightarrow P' \xrightarrow{(\alpha)} P''$ with $P' \mathcal{R} Q$ and $P'' \mathcal{R} Q'$.

where $\Longrightarrow$ is the transitive reflexive closure of $\xrightarrow{\tau}$ and $P \xrightarrow{(\alpha)} P'$ is defined as $P \xrightarrow{\alpha} P' \vee \alpha = \tau \wedge P = P'$. Processes P and Q are *branching bisimilar* (notation: $P \asymp Q$) if there exists a branching bisimulation that contains (P, Q).[1]

There are two differences with weak bisimulations (where $P \mathcal{R} Q$ and $P \xrightarrow{\alpha} P'$ entails the existence of Q' such that $Q \Longrightarrow \xrightarrow{(\alpha)} \Longrightarrow Q'$ and $P' \mathcal{R} Q'$, and the symmetric case where Q makes a transition). Firstly for branching bisimulations an additional related pair before the α-transition is required. Secondly the process P'' cannot do any more τ-transitions after the α-transition. This makes branching bisimilarity a coarser relation on processes. It has been introduced as the coarsest behavioral equivalence that abstracts from silent transitions in the linear time branching time spectrum [5], aside from rooted branching bisimilarity.

To prove that processes P and Q are *branching bisimilar* one could provide a relation $\mathcal{R}$ on processes that contains the pair (P, Q) and show that $\mathcal{R}$ is a branching bisimulation. Most textbooks would teach this proof method by starting with a singleton relation and adding more pairs until a branching bisimulation is established.

Example 1. In our examples we prove bisimilarity of CCS terms. To show that $\tau.(a.\bar{a} + \bar{a}.a + \tau)$ is branching bisimilar to $a|\bar{a}$ by starting with a singleton relation containing just this pair and expanding it as we go. Eventually the relation should contain five pairs.

[1] The presented notion is actually that of semibranching bisimulation [2]. The difference between branching and semibranching bisimulation is left outside the scope of this abstract.

This proof method can be very labour-intensive as bigger relations require that more pairs have to be checked. Up-to techniques weaken the requirement on the pairs in $\mathcal{R}$. Suppose that we have some function f that enlarges relations. In order to show that $\mathcal{R}$ is a *branching bisimulation up to some function f* the additional pairs, that is, the pairs $(P, Q'), (P', Q''), (P', Q)$ and (P'', Q') in the definition, should be in $f(\mathcal{R})$ instead of just $\mathcal{R}$.

Example 2. The relation containing $(\tau.(a.\bar{a} + \bar{a}.a + \tau), a|\bar{a})$ and $(a.\bar{a} + \bar{a}.a + \tau, a|\bar{a})$ is a branching bisimulation up to $\lambda\mathcal{R}.\mathcal{R} \cup \asymp$.

Of course if one establishes that $\mathcal{R}$ is a branching bisimulation up to f, then it is mostly not the case that $\mathcal{R}$ is a branching bisimulation. For *sound* functions f we do however have that any branching bisimulation up to f is contained in $\asymp$.

2 Applying the abstract framework for up-to techniques

Up-to techniques for strong bisimilarity [8, 11] and weak bisimilarity [12] have been studied extensively in the past. Pous and Sangiorgi describe an abstract framework that can be used to deal with coinductive definitions and up-to techniques in [10]. The same proof techniques that work for strong and weak bisimilarity have not been studied for branching bisimilarity. We use this framework to obtain a powerful proof technique and explain some of the theoretical notions that we need to obtain it.

To start off we should define a branching bisimulation in terms of a monotone function. Let $\mathbb{P}$ be a set of states and consider $\mathsf{br} : \mathcal{P}(\mathbb{P} \times \mathbb{P}) \to \mathcal{P}(\mathbb{P} \times \mathbb{P})$ defined as follows:

$$\begin{aligned}\mathsf{br}(\mathcal{R}) = \{(P, Q) \mid\ & \text{for all } P' \text{ and for all } \alpha, \text{ if } P \xrightarrow{\alpha} P' \text{ then there exist } Q', Q'' \\ & \text{such that } Q \Longrightarrow Q' \xrightarrow{(\alpha)} Q'' \text{ and } P\ \mathcal{R}\ Q' \text{ and } P'\ \mathcal{R}\ Q''; \text{ and} \\ & \text{for all } Q' \text{ and for all } \alpha, \text{ if } Q \xrightarrow{\alpha} Q' \text{ then there exist } P', P'' \\ & \text{such that } P \Longrightarrow P' \xrightarrow{(\alpha)} P'' \text{ and } P'\ \mathcal{R}\ Q \text{ and } P''\ \mathcal{R}\ Q'\}.\end{aligned}$$

This function is indeed monotone and its post-fixed points (the relations $\mathcal{R}$ that satisfy $\mathcal{R} \subseteq \mathsf{br}(\mathcal{R})$) are exactly the branching bisimulations. Moreover the greatest branching bisimulation $\asymp$ is the greatest fixed-point of br.

Given a monotone function $f : \mathcal{P}(\mathbb{P} \times \mathbb{P}) \to \mathcal{P}(\mathbb{P} \times \mathbb{P})$ a branching bisimulation up to f is a relation $\mathcal{R}$ that satisfies $\mathcal{R} \subseteq \mathsf{br}(f(\mathcal{R}))$. By proving $\mathcal{R} \subseteq \mathsf{br}(f(\mathcal{R}))$ one shows that $\mathcal{R} \subseteq \mathsf{gfp}(\mathsf{br} \circ f)$. The desired conclusion, namely $\mathcal{R} \subseteq \asymp$, only needs the missing link of soundness: we say that f is br-sound if $\mathsf{gfp}(\mathsf{br} \circ f) \subseteq \mathsf{gfp}(\mathsf{br})$.

2.1 Results

From the fact that br is monotone it immediately follows that $\lambda\mathcal{R}.\asymp$ and $\lambda\mathcal{R}.\mathcal{R} \cup \asymp$ are br-sound. Many branching bisimulation proofs are done by constructing a relation that consists of some manually defined pairs and all pairs in the greatest branching bisimulation. This ad-hoc technique corresponds to using a relation containing only the manually defined pairs and proving that it is a branching bisimulation up to $\lambda\mathcal{R}.\mathcal{R} \cup \asymp$.

The function $\lambda\mathcal{R}.{\sim}\mathcal{R}{\sim}$, where $\sim$ is strong bisimilarity, is br-sound. In contrast the function $\lambda\mathcal{R}.{\asymp}\mathcal{R}{\asymp}$ is unsound. The reader who is familiar with similar results for weak bisimilarity might not find this surprising. The unsoundness proof is similar to the one for 'weak bisimulation up to weak bisimilarity'. We can show that $(\tau.a, \mathsf{0})$ is a branching bisimulation up to $\lambda\mathcal{R}.{\asymp}\mathcal{R}{\asymp}$ whereas $\tau.a \not\asymp \mathsf{0}$.

Example 3. We consider the replication operator ! from the π-calculus: the process $!P$ behaves as infinitely many parallel copies of P. We can show that $!\tau \asymp \mathsf{0}$ by showing that the singleton relation $\{(!\tau, \mathsf{0})\}$ is a branching bisimulation up to $\lambda\mathcal{R}.{\sim}\mathcal{R}{\sim}$.

It is rather unsatisfactory to use strong bisimilarity in an up-to technique when we want to abstract from silent transitions. For weak bisimilarity there is a better solution [10] which includes using an efficiency preorder defined in [1]. A similar result can be obtained for branching bisimilarity. Without going into details we say that *branching expansion*, denoted by $\succcurlyeq$, is the efficiency preorder that is associated with branching bisimilarity. Intuitively $P \succcurlyeq Q$ holds whenever $P \asymp Q$ and in addition Q is always more efficient than P regarding the number of τ-steps that are used. The corresponding function $\lambda\mathcal{R}.{\succcurlyeq}\mathcal{R}{\preccurlyeq}$ is br-sound. One can check that the problematic case proving that $\lambda\mathcal{R}.{\asymp}\mathcal{R}{\asymp}$ is unsound does not cause problems here since $\tau.a \not\succcurlyeq a$.

Example 4. Define $P \stackrel{\mathsf{def}}{=} a.P$ and $Q \stackrel{\mathsf{def}}{=} a.\tau.\tau.\tau.Q$. We can prove that $P \asymp Q$ with a singleton branching bisimulation up to $\lambda\mathcal{R}.{\succcurlyeq}\mathcal{R}{\preccurlyeq}$.

As a last up-to technique one could use congruence properties of process algebras inside the branching bisimulation game. We show that for guarded CCS the *up-to context* technique is br-sound. A context in CCS is a term with numbered holes in it that can occur multiple times. The contextual closure of a relation can then be defined as follows:

$$\mathcal{C}(\mathcal{R}) = \{(C[P_1, \ldots, P_n], C[Q_1, \ldots, Q_n]) \mid C \text{ is a context and } P_i \;\mathcal{R}\; Q_i \text{ for all } i\}\,.$$

This contextual closure is br-sound.

2.2 Combinations of techniques

The problem with the notion of soundness is that it is not preserved under composition: for two br-sound functions f, g it is not necessarily the case that $g \circ f$ is br-sound. The abstract framework provides a way to combine all the functions that we mentioned so far. All functions that we have reported br-sound so far, are in fact *br-respectful*. We call a function br-respectful if $f \circ (\mathsf{br} \wedge \mathsf{id}) \leq (\mathsf{br} \wedge \mathsf{id}) \circ f$.[2] Results from [9, 10] show that every composition of br-respectful functions is again br-respectful. Moreover any br-respectful function is br-sound. This leads to the conclusion that a technique like up to $\lambda\mathcal{R}.{\succcurlyeq}\mathcal{C}(\mathcal{R} \cup {\asymp}){\preccurlyeq}$ is sound for branching bisimilarity.

Example 5. We can prove that $!(a + b) \asymp !\tau.a|!\tau.b$ by using a singleton branching bisimulation up to $\lambda\mathcal{R}.{\sim}\mathcal{C}(\mathcal{R}){\sim}$.

Example 6. Define the delayed replication of a as $D_a \stackrel{\mathsf{def}}{=} \tau.(\tau.D_a|\tau.a)$. We can prove that $D_a \asymp !a$ by using a singleton branching bisimulation up to $\lambda\mathcal{R}.{\succcurlyeq}\mathcal{C}(\mathcal{R} \cup {\asymp}){\preccurlyeq}$.

3 Future work

The technique of branching bisimulation up to context can be generalized from CCS to languages of which the operational rules adhere to a particular format. We can show respectfulness of the up-to context technique for Van Glabbeek's simply BB cool languages [6]. This SOS format is however very restrictive. It is still unclear under which more general conditions up-to context is a sound or respectful technique.

Delay bisimilarity and η-bisimilarity [5] are also notions of weak behavioral equivalence. Unsurprisingly the results for branching bisimilarity also apply to these equivalences. A notion of expansion can be created for both of them, their associated techniques

[2] Here $f \wedge g$ is defined as $\lambda\mathcal{R}.f(\mathcal{R}) \cap g(\mathcal{R})$ and we say that $f \leq g$ iff $f(\mathcal{R}) \subseteq g(\mathcal{R})$ for all $\mathcal{R}$.

of up to $\lambda\mathcal{R}.\succcurlyeq\mathcal{R}\preccurlyeq$ are respectful and Van Glabbeek's simple DB/HB cool languages are suitable for their respective up to context techniques.

In [3] up-to techniques for weak bisimilarity are discussed. In particular up-to context for GSOS languages is investigated in a categorical setting. We aim to take a similar approach for branching bisimilarity.

References

[1] S. Arun-Kumar and Matthew Hennessy. An efficiency preorder for processes. *Acta Inf.*, 29(8):737–760, 1992.

[2] Twan Basten. Branching bisimilarity is an equivalence indeed! *Inf. Process. Lett.*, 58(3):141–147, 1996.

[3] Filippo Bonchi, Daniela Petrisan, Damien Pous, and Jurriaan Rot. Lax bialgebras and up-to techniques for weak bisimulations. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 240–253. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[4] Rick Erkens. Proof techniques for branching bisimulation. Master's thesis, Radboud University Nijmegen, 2018.

[5] Rob van Glabbeek. The linear time - branching time spectrum II. In Eike Best, editor, *CONCUR '93, 4th International Conference on Concurrency Theory, Hildesheim, Germany, August 23-26, 1993, Proceedings*, volume 715 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 1993.

[6] Rob van Glabbeek. On cool congruence formats for weak bisimulations. *Theor. Comput. Sci.*, 412(28):3283–3302, 2011.

[7] Rob van Glabbeek and W Peter Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the ACM (JACM)*, 43(3):555–600, 1996.

[8] Robin Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989.

[9] Damien Pous. Coinduction all the way up. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 307–316. ACM, 2016.

[10] Damien Pous and Davide Sangiorgi. Enhancements of the bisimulation proof method, 2012.

[11] Davide Sangiorgi. On the proof method for bisimulation (extended abstract). In Jirí Wiedermann and Petr Hájek, editors, *Mathematical Foundations of Computer Science 1995, 20th International Symposium, MFCS'95, Prague, Czech Republic, August 28 - September 1, 1995, Proceedings*, volume 969 of *Lecture Notes in Computer Science*, pages 479–488. Springer, 1995.

[12] Davide Sangiorgi and Robin Milner. The problem of "weak bisimulation up to". In Rance Cleaveland, editor, *CONCUR '92, Third International Conference on Concurrency Theory, Stony Brook, NY, USA, August 24-27, 1992, Proceedings*, volume 630 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 1992.