

Divergence-Preserving Branching Bisimilarity

Bas Luttik

Eindhoven University of Technology
The Netherlands

s.p.luttik@tue.nl

This note considers the notion of divergence-preserving branching bisimilarity. It briefly surveys results pertaining to the notion that have been obtained in the past one-and-a-half decade, discusses its role in the study of expressiveness of process calculi, and concludes with some suggestions for future work.

1 Introduction

Branching bisimilarity was proposed by van Glabbeek and Weijland as an upgrade of (strong) bisimilarity that facilitates abstraction from internal activity [16]. It preserves the branching structure of processes more strictly than Milner’s *observation equivalence* [21], which, according to van Glabbeek and Weijland, makes it, e.g., better suited for verification purposes. A case in point is the argument by Graf and Sifakis that there is no temporal logic with an *eventually* operator that is adequate for observation equivalence in the sense that two processes satisfy the same formulas if, and only if, they are observationally equivalent [17]. The crux is that observation equivalence insufficiently takes into account the intermediate states of an internal computation. Indeed, branching bisimilarity requires a stronger correspondence between the intermediate states of an internal computation.

Branching bisimilarity is also not compatible with a temporal logic that includes an eventually operator, because it abstracts to some extent from *divergence* (i.e., infinite internal computations). Thus, a further upgrade is necessary, removing the abstraction from divergence. De Nicola and Vaandrager show that *divergence-sensitive* branching bisimilarity coincides with the equivalence induced by satisfaction of formulas of the temporal logic CTL^*_{-X} [6]. (CTL^* [8] is an expressive state-based logic that includes both linear time and branching time modalities; CTL^*_{-X} refers to the variant of CTL^* obtained by omitting the next-state modality, which is incompatible with abstraction from internal activity.)

Divergence-sensitive branching bisimilarity still has one drawback when it comes to verification: it identifies deadlock and livelock and, as an immediate consequence, is not compatible with parallel composition. It turns out that the notion of *divergence-preserving* branching bisimilarity¹, which is the topic of this note, has all the right properties: it is the coarsest equivalence that is compatible with parallel composition, preserves CTL^*_{-X} formulas, and distinguishes deadlock and livelock [15]. Moreover, on finite processes divergence-preserving branching bisimilarity can be decided efficiently [18].

In [16], a coloured-trace characterisation of divergence-preserving branching bisimilarity is provided. In [13], relational and modal characterisations of the notion are given. For some time it was simply assumed that these three characterisations of the notion coincide, but this was only proved in [14]. To establish that the relational characterisation coincides with the coloured-trace and modal characterisations, it needs to be proved that the relational characterisation yields an equivalence relation that satisfies

¹For stylistic reasons we prefer the term “divergence-preserving branching bisimilarity” over “branching bisimilarity with explicit divergence”, which is used in earlier articles on the topic.

the so-called stuttering property, and this is surprisingly involved. A similar phenomenon is observed in the proof that a rooted version of divergence-preserving branching bisimilarity is compatible with the recursion construct $\mu X. \dots$ [12]. Due to the divergence condition, Milner's ingenious argument in [23] that strong bisimilarity is compatible with recursion required several novel twists.

In this note we shall give a survey of results pertaining to divergence-preserving branching bisimilarity that were obtained in the past one-and-a-half decade. In Section 2 we shall present and discuss a relational characterisation of the notion. In Section 3 we comment on modal characterisations of the notion, and discuss the relationship with the temporal logic CTL^*_X . In Section 4 we briefly discuss to what extent the notion is compatible with familiar process algebraic operators. In Section 5, we explain how it plays a role in expressiveness results. In Section 6 we arrive at some conclusions and mention some ideas for future work.

2 Relational characterisation

We presuppose a set \mathcal{A}_τ of *actions* including a special element τ , and we presuppose a *labelled transition system* (S, \longrightarrow) with labels from \mathcal{A}_τ , i.e., S is a set of *states* and $\longrightarrow \subseteq S \times \mathcal{A}_\tau \times S$ is a *transition relation* on S . Let $s, s' \in S$ and $\alpha \in \mathcal{A}_\tau$; we write $s \xrightarrow{\alpha} s'$ for $(s, \alpha, s') \in \longrightarrow$ and we abbreviate the statement ' $s \xrightarrow{\alpha} s'$ or $(\alpha = \tau \text{ and } s = s')$ ' by $s \xrightarrow{(\alpha)} s'$. We denote by \rightarrow^+ the transitive closure of the binary relation $\xrightarrow{\tau}$, and by \twoheadrightarrow its reflexive-transitive closure. A *process* is given by a state s in a labelled transition system, and encompasses all the states and transitions reachable from s .

Definition 1. A symmetric binary relation \mathcal{R} on S is a *branching bisimulation* if it satisfies the following condition for all $s, t \in S$ and $\alpha \in \mathcal{A}_\tau$:

- (T) if $s \mathcal{R} t$ and $s \xrightarrow{\alpha} s'$ for some state s' , then there exist states t' and t'' such that $t \twoheadrightarrow t'' \xrightarrow{(\alpha)} t'$, $s \mathcal{R} t''$ and $s' \mathcal{R} t'$.

We say that a branching bisimulation \mathcal{R} *preserves (internal) divergence* if it satisfies the following condition for all $s, t \in S$:

- (D) if $s \mathcal{R} t$ and there is an infinite sequence of states $(s_k)_{k \in \omega}$ such that $s = s_0$, $s_k \xrightarrow{\tau} s_{k+1}$ and $s_k \mathcal{R} t$ for all $k \in \omega$, then there is a state t' such that $t \twoheadrightarrow^+ t'$, and $s_k \mathcal{R} t'$ for some $k \in \omega$.

States s and t are *divergence-preserving branching bisimilar* (notation: $s \stackrel{\Delta}{\sim}_b t$) if there is a divergence-preserving branching bisimulation \mathcal{R} such that $s \mathcal{R} t$.

The divergence condition (D) in the definition above is slightly weaker than the divergence condition used in the relation characterisation of divergence-preserving branching bisimilarity presented in [13], which actually requires that t admits an infinite sequence of τ -transitions and every state on this sequence is related to some state on the infinite sequence of τ -transitions from s . Nevertheless, as is established in [14], the notion of divergence-preserving branching bisimilarity defined here is equivalent to the one defined in [13]. In [14] it is also proved that $\stackrel{\Delta}{\sim}_b$ is an equivalence, that the relation $\stackrel{\Delta}{\sim}_b$ is itself a divergence-preserving branching bisimulation, and that it satisfies the so-called *stuttering property*: if $t_0 \xrightarrow{\tau} \dots t_n$, $s \stackrel{\Delta}{\sim}_b t_0$ and $s \stackrel{\Delta}{\sim}_b t_n$, then $s \stackrel{\Delta}{\sim}_b t_i$ for all $0 \leq i \leq n$.

Let us say that a state s is *divergent* if there exists an infinite sequence of states $(s_k)_{k \in \omega}$ such that $s = s_0$ and $s_k \xrightarrow{\tau} s_{k+1}$ for all $k \in \omega$. It is a straightforward consequence of the definition that divergence-preserving branching bisimilarity relates divergent states to divergent states only, i.e., that we have the following proposition.

Proposition 2. *If $s \stackrel{\Delta}{\sim}_b t$, then s is divergent only if t is divergent.*

Proof. Suppose that $s \not\leftrightarrow_b^\Delta t$ and s is divergent. Then there exists an infinite sequence of states $(s_k)_{k \in \omega}$ such that $s = s_0$ and $s_k \xrightarrow{\tau} s_{k+1}$ for all $k \in \omega$. We inductively construct an infinite sequence of states $(t_\ell)_{\ell \in \omega}$ such that $t = t_0$, $t_\ell \xrightarrow{\tau} t_{\ell+1}$, together with a mapping $\sigma : \omega \rightarrow \omega$ such that $s_{\sigma(\ell)} \not\leftrightarrow_b^\Delta t_\ell$ for all $\ell \in \omega$;

- We define $t_0 = t$ and $\sigma(0) = 0$; note that $s_{\sigma(0)} = s \not\leftrightarrow_b^\Delta t = t_0$.
- Suppose that the sequence $(t_\ell)_{\ell \in \omega}$ and the mapping σ have been defined up to ℓ . Then, in particular, $s_{\sigma(\ell)} \not\leftrightarrow_b^\Delta t_\ell$. We distinguish two cases:

If $s_{\sigma(\ell)+k} \not\leftrightarrow_b^\Delta t_\ell$ for all $k \in \omega$, then by (D) there exists $t_{\ell+1}$ such that $t_\ell \xrightarrow{\tau} t_{\ell+1}$ and $s_{\sigma(\ell)+k} \not\leftrightarrow_b^\Delta t_{\ell+1}$ for some $k \in \omega$; we can then define $\sigma(\ell+1) = k$.

Otherwise, there exists some $k \in \omega$ such that $s_{\sigma(\ell)+k} \not\leftrightarrow_b^\Delta t_\ell$ and $s_{\sigma(\ell)+k+1} \not\leftrightarrow_b^\Delta t_\ell$. Since $s_{\sigma(\ell)+k} \xrightarrow{\tau} s_{\sigma(\ell)+k+1}$ it follows by (T) that there exist t_ℓ'' and $t_{\ell+1}$ such that $t_\ell \xrightarrow{\tau} t_\ell'' \xrightarrow{\tau} t_{\ell+1}$, $s_{\sigma(\ell)+k} \not\leftrightarrow_b^\Delta t_\ell''$ and $s_{\sigma(\ell)+k+1} \not\leftrightarrow_b^\Delta t_{\ell+1}$. Clearly, we have that $t_\ell \neq t_{\ell+1}$, so $t_\ell \xrightarrow{+} t_{\ell+1}$ and we can define $\sigma(\ell+1) = \sigma(\ell) + k + 1$.

From the existence of an infinite sequence of states $(t_\ell)_{\ell \in \omega}$ such that $t = t_0$ and $t_\ell \xrightarrow{\tau} t_{\ell+1}$ it follows that t is divergent, as was to be shown. \square

As the following example illustrates, however, a symmetric binary relation on S relating states that satisfies (T) of Definition 1 and relates divergent states to divergent states only is *not necessarily* included in a divergence-preserving branching bisimulation relation. In other words, a symmetric binary relation on S that satisfies (T) and only relates divergent states to divergent states may relate states that are not divergence-preserving branching bisimilar.

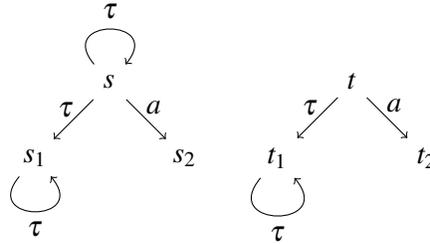


Figure 1: An example transition system illustrating that (D) cannot be replaced by the requirement that \mathcal{R} relates divergent states to divergent states.

Example 3. Consider the transition system depicted in Figure 1. The symmetric closure of the relation $\mathcal{R} = \{(s, t), (s_1, t_2), (s_2, t_2)\}$ satisfies (T) and it relates divergent states to divergent states only. It does not, however, satisfy (D), for $s \mathcal{R} t$ and defining $s_k = s$ for all $k \in \omega$ we get an infinite sequence of states $(s_k)_{k \in \omega}$ such that $s_k \xrightarrow{\tau} s_{k+1}$ and $s_k \mathcal{R} t$ for all $k \in \omega$, while there does not exist a t' such that $t \xrightarrow{+} t'$ and $s_k \mathcal{R} t'$ for some $k \in \omega$. Note that s admits a complete path at which a is continuously (weakly) enabled, whereas t does not admit such a complete path.

3 Modal characterisations

As shown in [13], to get an (action-based) modal logic that is adequate for branching bisimilarity one could take an adaptation of standard Hennessy-Milner logic replacing, for all actions $\alpha \in \mathcal{A}_\tau$ in the

usual unary may and must modalities $\langle a \rangle$ and $[a]$ by a binary *just-before* modality a . A state s satisfies the formula $\varphi a \psi$ if, and only if, there exist states s'' and s' such that $s \longrightarrow s'' \xrightarrow{(\alpha)} s'$, φ holds in s'' and ψ holds in s' . To get an adequate logic for divergence-preserving branching bisimilarity, it suffices to add a unary *divergence modality* Δ such that s satisfies $\Delta\varphi$ if, and only if, there exists an infinite sequence of states $(s_k)_{k \in \omega}$ such that $s \longrightarrow s_0$, $s_k \xrightarrow{\tau} s_{k+1}$ and φ holds in s_k for all $k \in \omega$.

Let Φ be the class of formulas generated by the following grammar:

$$\varphi ::= \neg\varphi \mid \bigwedge \Phi' \mid \varphi \alpha \varphi \mid \Delta\varphi \quad (\alpha \in \mathcal{A} \setminus \tau, \varphi \in \Phi, \Phi' \subseteq \Phi) .$$

We then have that states s and t are divergence-proving branching bisimilar if, and only if, s and t satisfy exactly the same formula in Φ [14]. We may restrict the cardinality of Φ' in conjunctions to the cardinality of the set of states S .

Example 4. Consider again the transition system depicted in Figure 1. States s and t are not divergence-preserving branching bisimilar. The formula $\Delta(\top a \top)$ (in which \top abbreviates $\bigwedge \emptyset$) expresses the existence of a divergence on which the action a is continuously enabled. It is satisfied by state s , but not by t .

There is also an intuitive correspondence between branching bisimilarity and the state-based temporal logic CTL_{-X}^* (CTL^* without the next-state modality) [7]. The standard semantics of CTL_{-X}^* is, however, with respect to Kripke structures, in which states rather than transitions have labels and the transition relation is assumed to be total. To formalise the correspondence, De Nicola and Vaandrager devised a framework of translations between labelled transition systems and Kripke structures [6]. The main idea of the translation from labelled transition systems to Kripke structures is that

1. every transition $s \xrightarrow{a} t$ ($a \neq \tau$) is replaced by two transitions $s \longrightarrow t_a$ and $t_a \longrightarrow t$, where t_a is a fresh state that is labelled with $\{a\}$;
2. every transition $s \xrightarrow{\tau} t$ gives rise to a transition $s \longrightarrow t$; and
3. for every state s without outgoing transitions (i.e., every deadlock state of the labelled transition system) a transition $s \longrightarrow s$ is added to satisfy the totality requirement of Kripke structures.

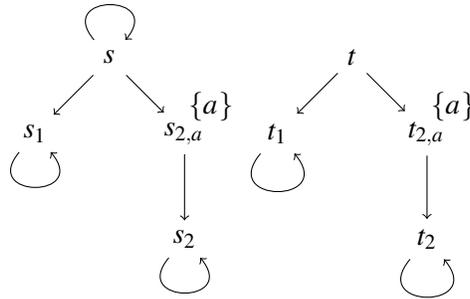


Figure 2: Result of apply De Nicola and Vaandrager's translation to the labelled transition system in Figure 1.

Example 5. If we apply the translation sketched above to the labelled transition system depicted in Figure 1, then we get the Kripke structure depicted in Figure 2. Note that by clause 3 of the translation state s_2 gets a transition to itself, whereas it is a deadlock state in the original transition system. Clearly, there is no CTL_{-X} formula that distinguishes, e.g., between s_1 and s_2 , although in the labelled transition system depicted in Figure 1 these states are not divergence-preserving branching bisimilar.

De Nicola and Vaandrager propose a notion of *divergence-sensitive branching bisimilarity* on finite LTSs and establish that two states in an LTS are divergence-sensitive branching bisimilar if, and only if, in the Kripke resulting from the translation sketched above they satisfy the same CTL_{-X}^* formulas. Divergence-sensitive branching bisimilarity coincides with divergence-preserving branching bisimilarity on deadlock-free LTSs. In fact, the only difference between divergence-sensitive branching bisimilarity and divergence-preserving branching bisimilarity is that the latter distinguishes between deadlock and livelock states, whereas the former does not.

To preserve the distinction between deadlock and livelock, a modified translation is proposed in [15], obtained from the translation sketched above by replacing clause 3 by

- 3'. add a fresh state d labelled with $\{\delta\}$, and for every state s without outgoing transitions a transition $s \longrightarrow d$.

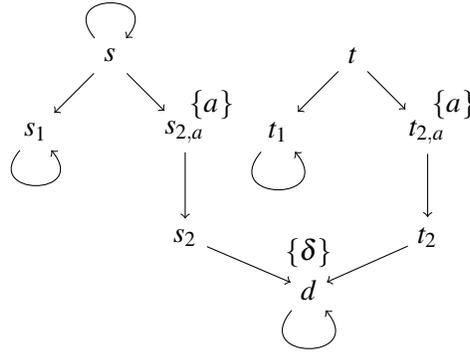


Figure 3: Result of apply the deadlock preserving translation to the labelled transition system in Figure 1.

Example 6. Applying the modified translation on the labelled transition in Figure 1, we get the Kripke structure in Figure 3. Note that s_1 does not satisfy the CTL_{-X}^* formula $\text{EF } \delta$, while s_2 does.

Two states in a labelled transition system are divergence-preserving branching bisimilar if they satisfy the same CTL_{-X}^* formulas in the Kripke structure that results from the modified transition [15].

4 Congruence

An important reason to prefer divergence-preserving branching bisimilarity over divergence-sensitive branching bisimilarity is that the former is compatible with parallel composition, whereas the latter is not.

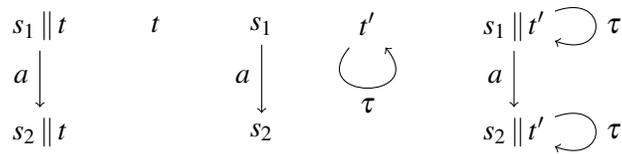


Figure 4: Divergence-sensitive branching bisimilarity is not compatible with parallel composition.

Example 7. Consider the transition system in Figure 4. States $s_1 \parallel t$ and $s_2 \parallel t$ represent the parallel compositions of states s_1 and t , and of states s_2 and t , respectively. Similarly, states $s_1 \parallel t'$ and $s_2 \parallel t'$ represent the parallel compositions of states s_1 and t' , and of states s_2 and t' , respectively. Recall that divergence-sensitive branching bisimilarity does not distinguish deadlock (state t) and livelock (state t'), so we have that t and t' are divergence-sensitive branching bisimilar. States $s_1 \parallel t$ and $s_1 \parallel t'$ are, however, not divergence-sensitive branching bisimilar. Note that $s_1 \parallel t'$ has a complete path on which a is continuously enabled, whereas $s_1 \parallel t$ does not have such a complete path, and so these two states do not satisfy the same CTL_{-X}^* formulas.

Divergence-preserving branching bisimilarity is the coarsest equivalence included in divergence-sensitive branching bisimilarity that is compatible with parallel composition [15]. Hence, it is also the coarsest congruence for parallel composition relating only processes that satisfy the same CTL_{-X}^* formulas.

It is well-known that branching bisimilarity is not compatible with non-deterministic choice, and that the coarsest behavioural equivalence that is included in branching bisimilarity and that is compatible with non-deterministic choice, is obtained by adding a so-called *root condition*. The same holds for divergence-preserving branching bisimilarity.

Definition 8. Let \mathcal{R} be a divergence-preserving branching bisimulation. We say that \mathcal{R} satisfies the *root condition* for s and t if, whenever

- (R1) if $s \xrightarrow{\alpha} s'$ for some state s' , then there exists a state t' such that $t \xrightarrow{\alpha} t'$ and $s' \mathcal{R} t'$.
- (R2) if $t \xrightarrow{\alpha} t'$ for some state t' , then there exists a state s' such that $s \xrightarrow{\alpha} s'$ and $s' \mathcal{R} t'$.

States s and t are *rooted divergence-preserving branching bisimilar* if there is a divergence-preserving branching bisimulation relation \mathcal{R} satisfying the root condition for s and t such that that $s \mathcal{R} t$.

In [11], formats for transition system specifications are presented that guarantee that divergence-preserving branching bisimilarity and its rooted variant are compatible with the operators defined by the transition system specification. These formats relax the requirements of the branching bisimulation and rooted branching bisimulation formats of [10]. The relaxation of the formats is meaningful: the process-algebraic operations for *priority* [1] and *sequencing* [5, 4, 3], with which (rooted) branching bisimilarity is *not* compatible, are in the rooted divergence-preserving branching bisimulation format. So, in contrast to its divergence-insensitive variant, rooted divergence-preserving branching bisimilarity is compatible with priority and sequencing.

The structural operational rule for the recursion operator $\mu X. \dots$, which was considered in the context of observation equivalence by Milner [22] and in the context of divergence-sensitive variants of observation equivalence by Lohrey, D'Argenio and Hermanns [19], is not in the format for rooted divergence-preserving branching bisimilarity. Nevertheless, rooted divergence preserving branching bisimilarity is compatible also with this operator [12]. The proof of this fact requires an adaption of the up-to technique used by Milner in his argument that (strong) bisimilarity is compatible with recursion [23].

5 Expressiveness of process calculi

Phillips showed that abstraction from divergence can be exploited to prove that every recursively enumerable transition system is branching bisimilar to a boundedly branching computable transition system [25]². In contrast, there exist recursively enumerable transition systems that are not divergence-

²Phillips actually claimed the correspondence modulo observation equivalence, but it is easy to see that his proof also works modulo branching bisimilarity.

preserving branching bisimilar to a computable transition system (cf., e.g., Example 3.6 in [2]). Hence, in a theory that aims to integrate computability and concurrency, divergence preservation is important.

In [2], interactivity is added to Turing machines by associating an action with every computation step. This so-called *reactive* Turing machine has a transition system semantics and can be studied from a concurrency-theoretic perspective. A transition system is called *executable* if it is behaviourally equivalent to the transition system associated with a reactive Turing machine. The notion of executability provides a way to characterise the absolute expressiveness of a process calculus. If every transition system that can be specified in the calculus is executable, then the calculus is said to be executable. Conversely, if every executable transition system can be specified in the calculus, then the calculus is said to be behaviourally complete.

A calculus with constants for deadlock and successful termination, unary action prefixes, binary operations for non-deterministic choice, sequencing and ACP-style parallel composition, iteration and nesting is both executable and behaviourally complete up to divergence-preserving branching bisimilarity [3]. The π -calculus is also behaviourally complete up to divergence-preserving branching bisimilarity. Since it allows the specification of transition systems with unbounded branching, it is, however, not executable up to divergence-preserving branching bisimilarity; it is nominally orbit-finitely executable up to the divergence-insensitive variant of branching bisimilarity [20].

The aforementioned results illustrate the role of divergence in the consideration of the absolute expressiveness of process calculi. Preservation of divergence is also widely accepted as an important criterion when comparing the relative expressiveness of process calculi [24].

6 Conclusions

We have discussed the relational and modal characterisations of divergence-preserving branching bisimilarity, commented on its compatibility with respect to process algebraic operations and on its role in the study of the absolute expressiveness. We conclude by briefly mentioning some directions for future work.

Sound and complete axiomatisations for the divergence-sensitive spectrum of observation congruence for basic CCS with recursion are provided in [19]. The congruence result in [12] can serve as a stepping stone for providing similar sound and complete axiomatisations for divergence-preserving branching bisimilarity. Then, it would also be interesting to consider the axiomatisation of divergence-preserving branching bisimilarity for full CCS with recursion, although that would first require a non-trivial extension of the congruence result.

Ad hoc up-to techniques for divergence-preserving branching bisimilarity have already been used, e.g., in the congruence proof in [12] and in proof that the π -calculus is behaviourally complete [20]. Recently, several more generic up-to techniques for branching bisimilarity were proved sound [9]. An interesting direction for future work would be to consider extending those up-to techniques for divergence-preserving branching bisimilarity too.

References

- [1] J. C. M. Baeten, J. A. Bergstra & J. W. Klop (1986): *Syntax and defining equations for an interrupt mechanism in process algebra*. *Fundamenta Informaticae* 9(2), pp. 127–167.
- [2] J. C. M. Baeten, B. Luttik & P. van Tilburg (2013): *Reactive Turing machines*. *Inf. Comput.* 231, pp. 143–166, doi:10.1016/j.ic.2013.08.010.

- [3] J. C. M. Baeten, B. Luttik & F. Yang (2017): *Sequential Composition in the Presence of Intermediate Termination (Extended Abstract)*. In K. Peters & S. Tini, editors: Proceedings Combined 24th International Workshop on Expressiveness in Concurrency and 14th Workshop on Structural Operational Semantics, EXPRESS/SOS 2017, Berlin, Germany, 4th September 2017, EPTCS 255, pp. 1–17, doi:10.4204/EPTCS.255.1.
- [4] A. Belder, B. Luttik & J. C. M. Baeten (2019): *Sequencing and Intermediate Acceptance: Axiomatisation and Decidability of Bisimilarity*. In M. Roggenbach & A. Sokolova, editors: 8th Conference on Algebra and Coalgebra in Computer Science, CALCO 2019, June 3-6, 2019, London, United Kingdom, LIPIcs 139, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 11:1–11:22, doi:10.4230/LIPIcs.CALCO.2019.11.
- [5] B. Bloom (1994): *When is Partial Trace Equivalence Adequate?* *Formal Asp. Comput.* 6(3), pp. 317–338, doi:10.1007/BF01215409.
- [6] R. De Nicola & F. W. Vaandrager (1995): *Three Logics for Branching Bisimulation*. *Journal of the ACM* 42(2), pp. 458–487, doi:10.1145/201019.201032.
- [7] E. A. Emerson & E. M. Clarke (1982): *Using Branching Time Temporal Logic to Synthesize Synchronization Skeletons*. *Science of Computer Programming* 2(3), pp. 241–266, doi:10.1016/0167-6423(83)90017-5.
- [8] E. A. Emerson & J.Y. Halpern (1986): *‘Sometimes’ and ‘Not Never’ revisited: on branching time versus linear time temporal logic*. *Journal of the ACM* 33(1), pp. 151–178, doi:10.1145/4904.4999.
- [9] R. Erkens, J. Rot & B. Luttik (2020): *Up-to Techniques for Branching Bisimilarity*. In A. Chatzigeorgiou, R. Dondi, H. Herodotou, C. A. Kapoutsis, Y. Manolopoulos, G. A. Papadopoulos & F. Sikora, editors: SOFSEM 2020: Theory and Practice of Computer Science - 46th International Conference on Current Trends in Theory and Practice of Informatics, SOFSEM 2020, Limassol, Cyprus, January 20-24, 2020, Proceedings, Lecture Notes in Computer Science 12011, Springer, pp. 285–297, doi:10.1007/978-3-030-38919-2_24.
- [10] W. J. Fokkink, R. J. van Glabbeek & P. de Wind (2012): *Divide and congruence: From decomposition of modal formulas to preservation of branching and η -bisimilarity*. *Inf. Comput.* 214, pp. 59–85, doi:10.1016/j.ic.2011.10.011.
- [11] W. J. Fokkink, Glabbeek R. J. van & B. Luttik (2019): *Divide and congruence III: From decomposition of modal formulas to preservation of stability and divergence*. *Inf. Comput.* 268, doi:10.1016/j.ic.2019.104435.
- [12] R. J. van Glabbeek, B. Luttik & L. Spaninks (2020): *Rooted Divergence-Preserving Branching Bisimilarity is a Congruence*. CoRR abs/1801.01180. Available at <http://arxiv.org/abs/1801.01180>. Submitted.
- [13] R. J. van Glabbeek (1993): *The Linear Time – Branching Time Spectrum II; The semantics of sequential systems with silent moves (extended abstract)*. In E. Best, editor: Proceedings 4th International Conference on Concurrency Theory, CONCUR’93, Hildesheim, Germany, August 1993, LNCS 715, Springer, pp. 66–81, doi:10.1007/3-540-57208-2_6.
- [14] R. J. van Glabbeek, B. Luttik & N. Trčka (2009): *Branching Bisimilarity with Explicit Divergence*. *Fundamenta Informaticae* 93(4), pp. 371–392, doi:10.3233/FI-2009-109.
- [15] R. J. van Glabbeek, B. Luttik & N. Trčka (2009): *Computation Tree Logic with Deadlock Detection*. *Logical Methods in Computer Science* 5(4), doi:10.2168/LMCS-5(4:5)2009.
- [16] R. J. van Glabbeek & W. P. Weijland (1996): *Branching time and abstraction in bisimulation semantics*. *Journal of the ACM* 43(3), pp. 555–600, doi:10.1145/233551.233556.
- [17] S. Graf & J. Sifakis (1987): *Readiness Semantics for Regular Processes with Silent Actions*. In T. Ottmann, editor: Automata, Languages and Programming, 14th International Colloquium, ICALP87, Karlsruhe, Germany, July 13-17, 1987, Proceedings, Lecture Notes in Computer Science 267, Springer, pp. 115–125, doi:10.1007/3-540-18088-5_10.
- [18] J. F. Groote, D. N. Jansen, J. J. A. Keiren & A. J. Wijs (2017): *An $O(m \log n)$ Algorithm for Computing Stuttering Equivalence and Branching Bisimulation*. *ACM Trans. Comput. Logic* 18(2), doi:10.1145/3060140.

- [19] M. Lohrey, P. R. D’Argenio & H. Hermanns (2005): *Axiomatising divergence*. *Inf. Comput.* 203(2), pp. 115–144, doi:10.1016/j.ic.2005.05.007.
- [20] B. Luttik & F. Yang (2020): *The π -Calculus is Behaviourally Complete and Orbit-Finitely Executable*. CoRR abs/1410.4512v8. Available at <http://arxiv.org/abs/1410.4512>.
- [21] R. Milner (1980): *A Calculus of Communicating Systems*. *Lecture Notes in Computer Science* 92, Springer, doi:10.1007/3-540-10235-3.
- [22] R. Milner (1989): *A Complete Axiomatisation for Observational Congruence of Finite-State Behaviors*. *Inf. Comput.* 81(2), pp. 227–247, doi:10.1016/0890-5401(89)90070-9.
- [23] R. Milner (1990): *Operational and Algebraic Semantics of Concurrent Processes*. In Jan van Leeuwen, editor: *Handbook of Theoretical Computer Science (Vol. B)*, MIT Press, Cambridge, MA, USA, pp. 1201–1242. Available at <http://dl.acm.org/citation.cfm?id=114891.114910>.
- [24] K. Peters (2019): *Comparing Process Calculi Using Encodings*. In J. A. Pérez & J. Rot, editors: *Proceedings Combined 26th International Workshop on Expressiveness in Concurrency and 16th Workshop on Structural Operational Semantics, EXPRESS/SOS 2019, Amsterdam, The Netherlands, 26th August 2019, EPTCS 300*, pp. 19–38, doi:10.4204/EPTCS.300.2.
- [25] I. Phillips (1993): *A Note on Expressiveness of Process Algebra*. In G. L. Burn, S. J. Gay & M. Ryan, editors: *Proceedings of the First Imperial College Department of Computing Workshop on Theory and Formal Methods*, Isle of Thorns Conference Centre, Chelwood Gate, Sussex, UK, 29-31 March 1993, Workshops in Computing, Springer, pp. 260–264.