The complexity of deciding characteristic formulae *

Luca Aceto, Antonis Achilleos, Aggeliki Chalki, and Anna Ingólfsdóttir

Department of Computer Science, Reykjavik University, Reykjavik, Iceland
luca@ru.is, antonios@ru.is, angelikic@ru.is. annai@ru.is

1 Introduction

In concurrency theory, *characteristic formulae* serve as a bridge between model checking and preorder or equivalence checking. At an intuitive level, a characteristic formula provides a complete logical characterization of the behaviour of a process with respect to some notion of behavioural equivalence or preorder. For example, consider the widely used bisimulation equivalence relation [1]; Hennessy and Milner have shown in [2] that, under a mild finiteness condition, two processes are bisimilar if and only if they satisfy the same Hennessy-Milner logic (**HML**) formulae. Apart from its intrinsic theoretical interest, this seminal logical characterization of bisimilarity means that, when two processes are *not* bisimilar, there is always an **HML** formula that distinguishes between them. However, using the Hennessy-Milner theorem to show that two processes are bisimilar would involve verifying that they satisfy the same **HML** formulae and there are infinitely many of those. This is where characteristic formulae come into play. An **HML** formula φ is characteristic for process p, if every process q satisfies φ iff p and q are bisimilar. As a consequence, one can decide bisimulation equivalence between p and q by finding the characteristic formula $\chi(p)$ for p and checking whether $q \models \chi(p)$, that is a model-checking problem. Thus characteristic formulae allow one to reduce bisimilarity checking to model checking.

Conversely, Boudol and Larsen studied in [3] the problem of characterizing the collection of modal formulae for which model checking can be reduced to equivalence checking. See [4, 5, 6] for other contributions in that line of research. The aforementioned articles showed that characteristic formulae coincide with those that are consistent and prime. (A formula is prime if whenever it entails a disjunction $\varphi_1 \vee \varphi_2$, then it must entail φ_1 or φ_2.) Moreover, characteristic formulae with respect to the bisimulation relation coincide with the formulae that are consistent and complete, where a modal formula φ is complete, when for every modal formula ψ on the same propositional variables as φ, we can derive from φ either ψ or its negation. Note that in the case of bisimulation, a formula is prime iff it is complete. When one wants to reduce model checking to equivalence checking, the study of the complexity of identifying characteristic formulae modulo bisimilarity within (extensions of) **HML** is of relevance and has been addressed in [7, 8]. Typically, checking whether a formula is characteristic modulo bisimilarity has the same complexity as validity.

We described characteristic formulae using the example of bisimilarity, as it is the relation between processes that underlies the seminal Hennessy-Milner theorem and was used in much of the above-mentioned work. However there are a plethora of other preorder and equivalence relations that classify processes according to other possible behaviours; these and their logical characterizations have been extensively studied in concurrency theory—see e.g. [9, 10]. In this

[*] This work has been funded by the projects "Open Problems in the Equational Logic of Processes (OPEL)" (grant no 196050), "Mode(l)s of Verification and Monitorability" (MoVeMent) (grant no 217987), and "Learning and Applying Probabilistic Systems" (grant no. 206574-051) of the Icelandic Research Fund.of the Icelandic Research Fund.

work, we address the complexity of deciding and finding characteristic formulae with respect to four different preorders in van Glabbeek's branching-time spectrum, namely *simulation* ($\lesssim_S$), *complete simulation* ($\lesssim_{CS}$), *ready simulation* ($\lesssim_{RS}$), and *trace simulation* ($\lesssim_{TS}$) [9].

Our goal in this work is to study the complexity of determining whether a formula $\varphi \in \mathcal{L}_X$ is characteristic for some process p_φ modulo $\lesssim_X$, where $X \in \{S, CS, RS, TS\}$, or equivalently whether it is consistent and prime. For example, note that all consistent formulae in $\mathcal{L}_S$ that do not contain disjunctions are also prime. Thus, in this case deciding characteristic formulae reduces to deciding consistent formulae. However, when disjunctions are added to the language, the situation gets more complicated. For instance, formula $\langle a\rangle\mathbf{tt} \vee \langle b\rangle\mathbf{tt}$ is not prime, since $\langle a\rangle\mathbf{tt} \vee \langle b\rangle\mathbf{tt} \not\models \langle a\rangle\mathbf{tt}$ and $\langle a\rangle\mathbf{tt} \vee \langle b\rangle\mathbf{tt} \not\models \langle b\rangle\mathbf{tt}$, whereas formula $(\langle a\rangle\mathbf{tt} \vee \langle b\rangle\mathbf{tt}) \wedge \langle b\rangle\mathbf{tt}$ is prime.

In the sequel, we first give the necessary definitions and then we mention known complexity results on deciding preorders $\lesssim_S$, $\lesssim_{CS}$, and $\lesssim_{RS}$ respectively. We present our results on the complexity of deciding $\lesssim_{TS}$ and then, we state propositions and theorems establishing the complexity of identifying and finding characteristic formulae for the aforementioned preorders.

2 Definitions

Our semantic model is that of *labelled transition systems* (LTS) $\mathcal{S} = (P, A, \longrightarrow)$, where P is a set of states (or processes), A is a set of actions and $\longrightarrow \subseteq P \times A \times P$ is a transition relation on processes. We write $p \xrightarrow{a} q$ instead of $(p, a, q) \in \longrightarrow$. We say that a state p is deadlocked iff it has no outgoing transition. In this work, we consider finite LTSs.

For $X \in \{S, CS, RS, TS\}$, the preorder $\lesssim_X$ is the largest relation over the set of processes satisfying the following conditions for every p, q.

1. **Simulation (S):** $p \lesssim_S q \Leftrightarrow$ for all $p \xrightarrow{a} p'$ there exists some $q \xrightarrow{a} q'$ such that $p' \lesssim_S q'$.
2. **Complete simulation (CS):** $p \lesssim_{CS} q \Leftrightarrow$
   (a) for all $p \xrightarrow{a} p'$ there exists some $q \xrightarrow{a} q'$ such that $p' \lesssim_{CS} q'$, and
   (b) p is deadlocked iff q is deadlocked.
3. **Ready simulation (RS):** $p \lesssim_{RS} q \Leftrightarrow$
   (a) for all $p \xrightarrow{a} p'$ there exists some $q \xrightarrow{a} q'$ such that $p' \lesssim_{RS} q'$, and
   (b) the initial sets of actions of p and q coincide. (The set of initial actions of a state is the collection of actions that label its outgoing transitions.)
4. **Trace simulation (TS):** $p \lesssim_{TS} q \Leftrightarrow$
   (a) for all $p \xrightarrow{a} p'$ there exists some $q \xrightarrow{a} q'$ such that $p' \lesssim_{TS} q'$, and
   (b) the sets of traces of p and q coincide. (The set of traces of p is the set of all possible sequences of actions that can be observed by executing p.)

It is well-known that $\lesssim_{TS} \subsetneq \lesssim_{RS} \subsetneq \lesssim_{CS} \subsetneq \lesssim_S$. We denote by $\mathcal{L}_S$, $\mathcal{L}_{CS}$, $\mathcal{L}_{RS}$, and $\mathcal{L}_{TS}$ respectively, the fragments of **HML** that characterize these four preorders [9, 6]. For $X \in \{S, CS, RS, TS\}$, $\mathcal{L}_X$ is defined to be the set of formulae given by the corresponding grammar as follows:

1. $\mathcal{L}_S$: $\varphi_S ::= \; \mathbf{tt} \;\mid\; \mathbf{ff} \;\mid\; \varphi_S \wedge \varphi_S \;\mid\; \varphi_S \vee \varphi_S \;\mid\; \langle a\rangle\varphi_S$.
2. $\mathcal{L}_{CS}$: $\varphi_{CS} ::= \; \mathbf{tt} \;\mid\; \mathbf{ff} \;\mid\; \varphi_{CS} \wedge \varphi_{CS} \;\mid\; \varphi_{CS} \vee \varphi_{CS} \;\mid\; \langle a\rangle\varphi_{CS} \;\mid\; \mathbf{0}$, where $\mathbf{0} = \bigwedge_{a\in A}[a]\mathbf{ff}$.

3. $\mathcal{L}_{RS}$: $\varphi_{RS} ::= \mathbf{tt} \mid \mathbf{ff} \mid \varphi_{RS} \wedge \varphi_{RS} \mid \varphi_{RS} \vee \varphi_{RS} \mid \langle a \rangle \varphi_{RS} \mid [a]\mathbf{ff}$.
4. $\mathcal{L}_{TS}$: $\varphi_{TS} ::= \mathbf{tt} \mid \mathbf{ff} \mid \varphi_{TS} \wedge \varphi_{TS} \mid \varphi_{TS} \vee \varphi_{TS} \mid \langle a \rangle \varphi_{TS} \mid \psi_{TS}$,
   $\psi_{TS} ::= \mathbf{ff} \mid [a]\psi_{TS}$

Truth in an LTS $\mathcal{S} = (P, A, \longrightarrow)$ is defined through relation $\models$ in the standard way. In particular,

- $p \models \langle a \rangle \varphi$ iff there is some $p \xrightarrow{a} q$ such that $q \models \varphi$ and
- $p \models [a]\varphi$ iff for all $p \xrightarrow{a} q$ it is the case that $q \models \varphi$.

We say that φ is *true* or *satisfied* in p if $p \models \varphi$. An **HML** formula is *consistent* or *satisfiable* if it is satisfied in a process p.

$\mathcal{L}_X$ characterizes $\lesssim_X$, where $X \in \{S, CS, RS, TS\}$, in the following sense: for all p, q, $p \lesssim_X q$ iff for every $\varphi \in \mathcal{L}_X$, $p \models \varphi \implies q \models \varphi$.

3 Deciding preorders

Let $\lesssim \in \{\lesssim_S, \lesssim_{CS}, \lesssim_{RS}\}$. Given two finite processes p and q, deciding whether $p \lesssim q$ can be done in polynomial time [9]. To the best of our knowledge, the complexity of deciding the trace simulation preorder has not been examined yet. The following propositions state that deciding trace simulation is hard.

Proposition 1. *Deciding $\lesssim_{TS}$ on finite processes is* PSPACE*-complete under polynomial-time Turing reductions.*

Proposition 2. *Deciding $\lesssim_{TS}$ on finite loop-free processes is* coNP*-complete under polynomial-time Turing reductions.*

Note that we use polynomial-time oracle reductions instead of the more standard Karp reductions between decision problems. This means that deciding $\lesssim_{TS}$ on finite loop-free processes is also NP-hard under polynomial-time Turing reductions. Moreover, Proposition 2 implies that if $p \lesssim_{TS} q$ can be solved in polynomial time for some finite loop-free p, q, then P = NP.

In Propositions 1 and 2, hardness is established by showing that the trace equivalence of two processes can be decided by making two oracle calls to the problem of deciding the trace simulation preorder. Since deciding trace equivalence is PSPACE- and coNP-hard under Karp reductions on finite and finite loop-free processes respectively [11, 12], we obtain our hardness results. Membership in PSPACE can be easily proven for Proposition 1, whereas membership in coNP for Proposition 2 is based on an NP algorithm for deciding $\not\lesssim_{TS}$ on finite loop-free processes.

4 Deciding characteristic formulae modulo some preorder

Recall that a formula is characteristic iff it is consistent and prime. We determine the complexity of deciding whether a formula is characteristic modulo one of the preorders $\lesssim_S$, $\lesssim_{CS}$, and $\lesssim_{RS}$, by providing results about the satisfiability and primality problems for the respective logics.

Theorem 3. *Let $\mathbf{\Lambda}$ be one of the modal logics $\mathcal{L}_S$ and $\mathcal{L}_{CS}$. Given $\varphi \in \mathbf{\Lambda}$, deciding whether φ is satisfiable and prime is in* P.

Theorem 4.

(a) Let $|Act| = k$, where k is a constant. Given $\varphi \in \mathcal{L}_{RS}$, deciding whether φ is satisfiable and prime is in P.

(b) Let $|Act|$ be unbounded. Satisfiability in $\mathcal{L}_{RS}$ is NP*-complete, whereas primality in $\mathcal{L}_{RS}$ is* coNP*-complete.*

Polynomial-time complexity of the satisfiability problem in Theorems 3 and 4(a) is proven by a uniform algorithm that can be appropriately adjusted in each case. For primality in $\mathcal{L}_S$, there are rules that allow us to check whether a given formula φ is prime by checking the relationship between polynomially many subformulae of φ. In conclusion, the problem can be reduced to the reachability problem in an alternating graph, the nodes of which represent tuples of φ's subformulae. This algorithm can be extended to solve primality in $\mathcal{L}_{CS}$ and $\mathcal{L}_{RS}$ with a bounded action set. We also obtain the following corollary.

Corollary 5. *Let $\mathbf{\Lambda}$ be either $\mathcal{L}_S$, $\mathcal{L}_{CS}$, or $\mathcal{L}_{RS}$ with a bounded action set.*

(a) Given a characteristic formula $\varphi \in \mathbf{\Lambda}$, there is a polynomial-time algorithm that outputs a process p, for which φ is characteristic within $\mathbf{\Lambda}$.

(b) Given $\varphi \in \mathbf{\Lambda}$ and process p, verifying whether φ is characteristic within $\mathbf{\Lambda}$ for p is in P.

5 Finding characteristic formulae modulo some preorder

Given a process p, the problem of constructing the characteristic formula for p has been studied for a variety of preorders and equivalences [13, 4, 14, 15]. To resolve the complexity of the problem we consider two different ways of representing formulae and measuring their size. Given a formula φ, the first approach is to write φ explicitly and define its size to be equal to the number of symbols that appear in φ as above; the second one involves representing φ using recursive equations called declarations, and defining its declaration-size as the number of required declarations. We denote the former by $|\varphi|$ and the latter by $\mathrm{decl}(\varphi)$. For example, formula $\varphi_2 = \langle a\rangle(\langle a\rangle\mathbf{tt} \wedge \langle b\rangle\mathbf{tt}) \wedge \langle b\rangle(\langle a\rangle\mathbf{tt} \wedge \langle b\rangle\mathbf{tt})$ has size $|\varphi_2| = 13$ and declaration-size $\mathrm{decl}(\varphi_2) = 3$, as it can be represented by the equations $\varphi_2 = \langle a\rangle\varphi_1 \wedge \langle b\rangle\varphi_1$, $\varphi_1 = \langle a\rangle\varphi_0 \wedge \langle b\rangle\varphi_0$, and $\varphi_0 = \mathbf{tt}$. The following propositions hold.

Proposition 6. *Let $\mathbf{\Lambda}$ be one of the modal logics $\mathcal{L}_S$, $\mathcal{L}_{CS}$, $\mathcal{L}_{RS}$ with a bounded action set. Given a finite loop-free process p, finding the characteristic formula $\chi(p)$ for p within $\mathbf{\Lambda}$ is* NP*-hard under polynomial-time Turing reductions, if $\chi(p)$ is explicitly written.*

Proposition 7. *Let $\mathbf{\Lambda}$ be one of the modal logics $\mathcal{L}_S$, $\mathcal{L}_{CS}$, $\mathcal{L}_{RS}$. Given a finite loop-free process p, finding the characteristic formula $\chi(p)$ for p within $\mathbf{\Lambda}$ is in* P*, if $\chi(p)$ is given as a set of declarations.*

For example, consider process p_2 of Figure 1. Formula $\varphi_2 = \langle a\rangle(\langle a\rangle\mathbf{tt} \wedge \langle b\rangle\mathbf{tt}) \wedge \langle b\rangle(\langle a\rangle\mathbf{tt} \wedge \langle b\rangle\mathbf{tt})$ is characteristic for p_2 within $\mathcal{L}_S$. As we already mentioned, φ_2 can be given much more efficiently in declarative form than in explicit form. In general, the characteristic formula (within $\mathcal{L}_S$) φ_n for process p_n, where p_n has the form of p_2 and length n, is of exponential size in $|p_n|$, when φ_n is given in explicit form.

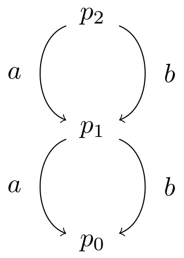

Figure 1: Process p_2 for which φ_2 is characteristic within $\mathcal{L}_S$.

Proposition 8. *Assume that for every finite loop-free process p, there is a characteristic formula within $\mathcal{L}_{TS}$ for p, denoted by φ_p, such that $\mathrm{decl}(\varphi_p)$ is polynomial in $|p|$ and every declaration is of polynomial size in $|p|$. Given a finite loop-free process p, if φ_p can be computed in polynomial time, then $\mathsf{P} = \mathsf{NP}$.*

Proposition 9. *Assume that the following two conditions are true:*

1. *For every finite loop-free process p, there is a characteristic formula within $\mathcal{L}_{TS}$ for p, denoted by φ_p, such that $\mathrm{decl}(\varphi_p)$ and every declaration are of polynomial size in $|p|$.*
2. *Given a finite loop-free process p and a formula φ in declarative form, deciding whether φ is characteristic within $\mathcal{L}_{TS}$ for p is in NP.*

Then $\mathsf{NP} = \mathsf{coNP}$.

Thus, when $\chi(p)$ is given as a set of declarations, we isolate a sharp difference between the complexity of finding $\chi(p)$ within any $\mathbf{\Lambda} \in \{\mathcal{L}_S, \mathcal{L}_{CS}, \mathcal{L}_{RS}\}$, and finding $\chi(p)$ within $\mathcal{L}_{TS}$.

6 Conclusions

Finally, we mention some problems that still remain open and whose solutions we are currently pursuing. First, we conjecture that for the trace simulation, deciding whether a formula is satisfiable is NP-complete, deciding primality of formulae is coNP-complete, whereas if we assume that $|A| = 1$, deciding both satisfiability and primality is in P. Yet another relevant problem is the complexity of deciding whether an **HML** formula φ is logically equivalent to a formula φ' in $\mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is one of $\mathcal{L}_S$, $\mathcal{L}_{CS}$, $\mathcal{L}_{RS}$, and $\mathcal{L}_{TS}$. Moreover, we want to address all the aforementioned problems for other relations in van Glabbeek's spectrum and over finite processes with loops.

References

[1] R. Milner, *Communication and Concurrency.* Prentice Hall, 1989.

[2] M. Hennessy and R. Milner, "Algebraic laws for nondeterminism and concurrency," *J. ACM*, vol. 32, no. 1, pp. 137–161, 1985. [Online]. Available: https://doi.org/10.1145/2455.2460

[3] G. Boudol and K. G. Larsen, "Graphical versus logical specifications," *Theor. Comput. Sci.*, vol. 106, no. 1, pp. 3–20, 1992. [Online]. Available: https://doi.org/10.1016/0304-3975(92)90276-L

[4] B. Steffen and A. Ingólfsdóttir, "Characteristic formulae for processes with divergence," *Inf. Comput.*, vol. 110, no. 1, pp. 149–163, 1994. [Online]. Available: https://doi.org/10.1006/inco.1994.1028

[5] L. Aceto, I. Fábregas, D. de Frutos-Escrig, A. Ingólfsdóttir, and M. Palomino, "Graphical representation of covariant-contravariant modal formulae," in *Proc. of EXPRESS 2011*, ser. EPTCS, vol. 64, 2011, pp. 1–15. [Online]. Available: https://doi.org/10.4204/EPTCS.64.1

[6] L. Aceto, D. D. Monica, I. Fábregas, and A. Ingólfsdóttir, "When are prime formulae characteristic?" *Theor. Comput. Sci.*, vol. 777, pp. 3–31, 2019. [Online]. Available: https://doi.org/10.1016/j.tcs.2018.12.004

[7] A. Achilleos, "The completeness problem for modal logic," in *Proc. of LFCS 2018*, ser. Lecture Notes in Computer Science, vol. 10703. Springer, 2018, pp. 1–21. [Online]. Available: https://doi.org/10.1007/978-3-319-72056-2_1

[8] L. Aceto, A. Achilleos, A. Francalanza, and A. Ingólfsdóttir, "The complexity of identifying characteristic formulae," *J. Log. Algebraic Methods Program.*, vol. 112, p. 100529, 2020. [Online]. Available: https://doi.org/10.1016/j.jlamp.2020.100529

[9] R. J. van Glabbeek, "The linear time - branching time spectrum I," in *Handbook of Process Algebra*. North-Holland / Elsevier, 2001, pp. 3–99. [Online]. Available: https://doi.org/10.1016/b978-044482830-9/50019-9

[10] D. de Frutos-Escrig, C. Gregorio-Rodríguez, M. Palomino, and D. Romero-Hernández, "Unifying the linear time-branching time spectrum of process semantics," *Log. Methods Comput. Sci.*, vol. 9, no. 2, 2013. [Online]. Available: https://doi.org/10.2168/LMCS-9(2:11)2013

[11] P. C. Kanellakis and S. A. Smolka, "CCS expressions, finite state processes, and three problems of equivalence," *Inf. Comput.*, vol. 86, no. 1, pp. 43–68, 1990. [Online]. Available: https://doi.org/10.1016/0890-5401(90)90025-D

[12] H. B. Hunt III, D. J. Rosenkrantz, and T. G. Szymanski, "On the equivalence, containment, and covering problems for the regular and context-free languages," *J. Comput. Syst. Sci.*, vol. 12, no. 2, pp. 222–268, 1976. [Online]. Available: https://doi.org/10.1016/S0022-0000(76)80038-4

[13] S. Graf and J. Sifakis, "A modal characterization of observational congruence on finite terms of CCS," *Information and Control*, vol. 68, no. 1-3, pp. 125–145, 1986. [Online]. Available: https://doi.org/10.1016/S0019-9958(86)80031-6

[14] L. Aceto, A. Ingólfsdóttir, M. L. Pedersen, and J. Poulsen, "Characteristic formulae for timed automata," *RAIRO Theor. Informatics Appl.*, vol. 34, no. 6, pp. 565–584, 2000. [Online]. Available: https://doi.org/10.1051/ita:2000131

[15] L. Aceto, A. Ingólfsdóttir, P. B. Levy, and J. Sack, "Characteristic formulae for fixed-point semantics: a general framework," *Math. Struct. Comput. Sci.*, vol. 22, no. 2, pp. 125–173, 2012. [Online]. Available: https://doi.org/10.1017/S0960129511000375