47

Equivalences of discrete-event systems and of hybrid systems

J.H. van Schuppen

CWI
P.O. Box 94079
1090 GB Amsterdam
THE NETHERLANDS
J.H.van.Schuppen@cwi.nl

1 Description of the problem

The problem is to develop concepts and theorems for equivalences of discrete-event systems and of hybrid systems.
For the equivalence of a discrete-event system consider a partially-observed deterministic automaton

$$A_1 = (Q_1, \Sigma_1, \delta_1, q_{10}, Q_{1m}, \Sigma_{1o}, p_1),$$

where Q_1 is an infinite discrete set called the state set; Σ_1 is a finite set called the event set; $\delta_1 : Q_1 \times \Sigma_1 \to Q_1$ is a partial function called the transition function; $q_{10} \in Q_1$ is the initial state; $Q_{1m} \subseteq Q_1$ is a subset of the state set representing accept states; Σ_{1o} is a finite set of observable events; Σ_1^* is the set of all finite strings with elements in Σ_1 and the empty string ϵ; and $p_1 : \Sigma_1^* \to \Sigma_{1o}^*$ is a causal map representing the observation map of the system.
The observation map is discussed in detail. Let Σ and Σ_0 be two alphabets and $L \subset \Sigma^*$ be a language. The map $p : L \to \Sigma_o^*$ is said to be causal if $p(\epsilon) = \epsilon$ and for all $s \in L$ and for all $\sigma \in \Sigma$ such that $s\sigma \in L$, either $p(s\sigma) = p(s)$ or there exists a $\sigma_o \in \Sigma_o$ such that $p(s\sigma) = p(s)\sigma_o \in \Sigma_o^*$. A projection is a special case of a causal map. The map $p : \Sigma^* \to \Sigma_o^*$, where $\Sigma_o \subseteq \Sigma$, is said to be projection if $p(\epsilon) = \epsilon$ and for $s \in \Sigma^*$, $p(p(s)) = p(s)$. A causal map is equivalent to a prefix-preserving map. See for references on causal and prefix-preserving maps [31] and [29, p. 248].
It is essential to the problem that the state set Q_1 is infinite and that the map p_1 is not injective.

Define the languages

$$\begin{aligned} L(A_1) &= \{s \in \Sigma_1^* \mid \delta_1(q_{10}, s) \in Q_{1m}\}, \\ p_1(L(A_1)) &= \{r \in \Sigma_{1o}^* \mid \exists\ s \in L^*(A_1) \text{ such that } r = p_1(s)\}. \end{aligned}$$

The problem is to formulate conditions on the discrete-event system A_1 such that there exists a partially-observed deterministic automaton

$$A_2 = (Q_2, \Sigma_2, \delta_2, q_{20}, Q_{2m}, \Sigma_{1o}, p_2),$$

with a *finite* state set Q_2 such that

$$p_1(L(A_1)) = p_2(L(A_2)). \tag{47.1}$$

It will be nice if the conditions have a system theoretic interpretation. Questions of the problem include:

1. What are necessary and sufficient conditions for the existence of the second system?
2. For which other concepts of equivalence than language equivalence as in (47.1) does the problem admit a solution that is moreover useful for problems of control and system theory?
3. How to exploit the equivalences and the corresponding classes for problems of control and system theory of discrete-event systems and of hybrid systems?

The emphasis in the problem is on the formulation of the concepts of equivalence, on system theoretic concepts for the problem, and on the use of equivalences for control and system theoretic problems. The emphasis is not exclusively on the computational properties of the solution. For particular instances the problem or the conditions of its solution are likely to be undecidable.

If one allows the state of the second system to be directly observable (p_2 is the identity map) then it is known that the equivalence exists iff the language $p_1(L(A_1))$ is regular, see [19, Th. 1.28]. Note however that the conditions asked for are to be formulated in terms of the system A_1 because in control theory that is the way the problem arises.

A hint for the problem is to think of decompositions of the state set as in decompositions of automata and of linear multivariable systems. The concept of decomposition of an automaton has been developed by K. Krohn and J. Rhodes, see [13]. For an exposition on decompositions see [10].

General references include: control and system theory [22]; automata and languages [10, 17, 19, 26, 30]; and control of discrete-event systems [16, 25, 29].

2 Motivation and history of the problem

The problem is motivated by problems of control and system theory for discrete-event systems and for hybrid systems. The aim of system theory is to identify classes of dynamic systems that admit the development of a theoretically rich and practically useful theory. Because a discrete-event system does have neither an analytic nor an algebraic structure, the currently existing system theory for finite-dimensional linear systems is not always an appropriate source of inspiration. System theorists may take inspiration from theoretical computer science but will have to adjust the available theory to control and system theoretic modeling.

Problems of signal representation, of the construction of observers, or of control synthesis can with some creativity be reformulated as realization problems, including the construction of equivalent systems. In addition, in control of discrete-event systems one considers the relation between the closed-loop system and the control objectives. The control objectives are often presented in the form of a language.

Conditions may have to be imposed on the second system to make the problem useful for engineering problems. Decidability of problems and limitation of the complexity of algorithims is of major importance to control and system theory. In the theory of computation a language is called *decidable* if there exists a Turing machine such that when any string of that language is fed to the machine, the machine will stop after a finite number of steps, see [19, Def. 3.3]. It is termed *undecidable* otherwise. A Turing machine is a deterministic automaton with an infinite number of states, see [19, Def. 3.1]. There are questions for Turing machines which are undecidable.

Decidability questions in control and system theory have been dealt with by several authors, see [3, 20, 21, 23, 27]. But the development of control and system theory requires a broader and deeper application of the concepts and results of decidability and of complexity. In control of discrete-event systems the concept of decidability deserves a place as prominent as finite-dimensionality in the system theory of linear systems. If a problem is decidable then computational properties of the problem may be used to delimit further the class of discrete-event systems for which theory can be developed. Approaches like hierarchical modeling and modularity are also used to combat complexity. But it is far from clear how to apply hierarchical modeling in specific situations.

In the research area of hybrid systems much attention has been given both by computer scientists and by control theorists to equivalences. R. Alur and D. Dill, see [1], identified a class of timed automata of which the untimed language is equivalent to a language of a finite-state automaton. This result may be considered as a solution to an extension of the problem formulated above. In the extension the first system is a timed-automaton and the map p_1 projects away the time component of the timed-event sequence. This approach has motivated much research in hybrid systems, see for example [12]. It seems of interest to also explore other equivalence relations and other classes of equivalent systems.

3 Extensions of the problem

The problem may be extended in several directions of which a few are described below.

The problem above was formulated for a deterministic automaton. Other classes of discrete-event systems for which the problem is of interest include nondeterministic automata, see [19], Petri nets, see [9], and process algebras, see [2]. Chomsky, [6], has introduced a hierarchy of languages according to which the classes of languages generated by finite-state automata, by Petri nets, and by process algebras are strictly contained in each other respectively. There exist subclasses of Petri nets for which certain control problems are decidable and there exists other such classes for which these are undecidable. The case of a decidable problem is an example of a solution to the proposed problem formulated above. Which conditions on the subclass and on the problem result in decidability of the problem? Yet another extension is to consider infinite string languages represented by finite state automata, see [24, 25, 26].

A possibly fruitfull extension is to consider for nondeterministic automata or for transition systems other concepts of equivalence as treated in the area of semantics of computer science. For nondeterministic systems failure semantics, see [15], failure trace semantics, see [18], and bisimulation equivalence, see [14], are used. Bisimulation is the finest of these relations, see [2]. Sources on bisimulation are [2, 11, 14]. For control problems more explicit control induced semantics may be required. The semantics may be structured by the underlying control system and by the problem, such as the hierarchical or decentralized structure of the system.

For the class of discrete-event systems A_1 for which the required equivalence does not exist, a further decomposition may be of interest to theory development. Do their exist subclasses of that class of systems which are equivalent with other classes of discrete-event systems having specified computational and complexity properties? In computer science a Turing machine is used to define decidability. This approach to computation and complexity started with the work of A. Church, see [7]. Would it be useful for complexity theory to relate complexity concepts to another automaton or to another discrete-event system? If parallel processing is allowed, then other properties of the second system may be considered.

Finally, the extension of the problem to equivalences of hybrid systems is of interest. Research is required into useful equivalence relations and their equivalence classes. For a paper on this approach see [8]. The concept of

abstraction is used for an equivalence relation in which part of the trajectory set is projected away, see [1, 12]. A paper by the author of this chapter for reachability of hybrid control systems, see [28], is based on the same approach. For hybrid systems the relation between discrete objects and real vector spaces needs to be exploited further. For an entry into this point the following references are recommended [4, 5].
Decidability of stability and of controllability of hybrid systems is discussed in the paper [3] but stability is not part of the problem proposed in this chapter.

4 References

[1] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183 - 235, 1994.

[2] J.C.M. Baeten and W.P. Weijland. *Process algebra.* Cambridge University Press, Cambridge, 1990.

[3] V. Blondel and J. Tsitsiklis. Complexity of stability and controllability of elementary hybrid systems. To appear in *Automatica.* Report LIDS-P 2388, LIDS, Massachusetts Institute of Technology, Cambridge, MA, 1997.

[4] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation.* Springer, New York, 1997.

[5] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers. *Bull. Amer. Math. Soc.*, 21:1–46, 1989.

[6] N. Chomsky. Three models for the description of language. *IRE Trans. Information Theory*, 2:113–124, 1956.

[7] A. Church. Logic, arithmetic, and automata. In *Proc. 1962 Internat. Congress of Math.*, pages 15–22, X, 1963. Mittag-Leffler Inst.

[8] B. Dasgupta and E.D. Sontag. A polynomial-time algorithm for an equivalence problem which arises in hybrid system theory. In *Proceedings of the Conference on Decision and Control*, page to appear, New York, 1998. IEEE Press.

[9] R. David and H. Alla. *Petri nets and grafcet.* Prentice Hall, New York, 1992.

[10] S. Eilenberg. *Automata, languages, and machines (Volumes A and B).* Academic Press, New York, 1974, 1976.

[11] T.A. Henzinger. Hybrid automata with finite bisimulations. In Z. Fülöp and F. Géseg, editors, *ICALP95: Automata, languages, and programming*, pages 324–335, Berlin, 1995. Springer-Verlag.

[12] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. In X, editor, *Proceedings of the 27th Annual Symposium on Theory of Computing*, pages 373–382, X, 1995. ACM Press.

[13] K. Krohn and J. Rhodes. Algebraic theory of machines. I. Prime decomposition theorem for finite semigroups and machines. *Trans. Amer. Math. Soc.*, 116:450–464, 1965.

[14] R. Milner. *Communication and concurrency.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[15] A. Overkamp. Supervisory control using failure semantics and partial specifications. *IEEE Trans. Automatic Control*, 42:498–510, 1997.

[16] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proc. IEEE*, 77:81–98, 1989.

[17] G. Rozenberg and A. Salomaa, editors. *Handbook of formal languages, Volumes 1-3.* Springer, Berlin, 1997.

[18] M. Shayman and R. Kumar. Supervisory control of nondeterministic systems with driven events via prioritized synchronization and trajectory models. *SIAM J. Control & Opt.*, 33:469–497, 1995.

[19] M. Sipser. *Introduction to the theory of computation.* PWS Publishing Company, Boston, 1997.

[20] E.D. Sontag. On certain questions of rationality and decidability. *J. Comp. Syst. Sci.*, 11:375–381, 1975.

[21] E.D. Sontag. Controllability is harder to decide than accessibility. *SIAM J. Control & Opt.*, 26:1106–1118, 1988.

[22] E.D. Sontag. *Mathematical control theory: Deterministic finite dimensional systems.* Springer-Verlag, New York, 1990.

[23] E.D. Sontag. From linear to nonlinear: Some complexity comparisons. In *Proc. IEEE Conference on Decision and Control*, pages 2916–2920, New York, 1995. IEEE Press.

[24] L. Staiger. ω-languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, pages 339–387. Springer, Berlin, 1997.

[25] J.G. Thistle and W.M. Wonham. Control of infinite behavior of finite automata. *SIAM J. Control & Opt.*, 32:1075–1097, 1994.

[26] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of theoretical computer science, Volume B*, pages 133 - 191. Elsevier Science Publishers B.V., Amsterdam, 1990.

[27] J.N. Tsitsiklis. On the control of discrete-event systems. *Math. Control Signals Systems*, 2:95–107, 1989.

[28] J.H. van Schuppen. A sufficient condition for controllability of a class of hybrid systems. In T.A. Henzinger and S. Sastry, editors, *Hybrid systems: Computation and control*, number 1386 in Lecture Notes in Computer Science, pages 374–383, Berlin, 1998. Springer.

[29] K.C. Wong and W.M. Wonham. Hierarchical control of discrete-event systems. *J. Discrete Event Dynamics Systems*, 6:241–273, 1996.

[30] Sheng Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages, Volume 1*, pages 41–110. Springer, Berlin, 1997.

[31] H. Zhong and W.M. Wonham. On the consistency of hierarchical supervisors in discrete-event systems. *IEEE Trans. Automatic Control*, 35:1125–1134, 1990.