



HAL
open science

Synthesis: Words and Traces

Igor Walukiewicz

► **To cite this version:**

| Igor Walukiewicz. Synthesis: Words and Traces. ATVA, 2010, Singapore. pp.18-21. hal-00536789

HAL Id: hal-00536789

<https://hal.science/hal-00536789>

Submitted on 16 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Synthesis: Words and Traces

Igor Walukiewicz

LaBRI (Université de Bordeaux - CNRS)

Abstract. The problem of synthesising a reactive system is discussed. The most standard instance of this problem ask to construct a finite input-output automaton satisfying a given regular specification. During fifty years since its introduction by Church, numerous extensions of the initial formulation have been considered. One particularly challenging case is that of distributed synthesis where a construction of a network of input/output automata is required.

1 General Context

Synthesis is about constructing a system from a given specification. For example: constructing of a circuit realizing a given boolean function. This case is easy until one adds constraints on placement of gates, etc. In some other settings the task is impossible from the very beginning: there is no algorithm constructing from an arithmetic formula a program realising the specified I/O function. Yet some, severe, restrictions of this problem are decidable, e.g., when considering only formulas of Presburger arithmetic. Here we will be interested in an extension of the first example to infinite, reactive behaviour.

The starting point of this research is the setting proposed by A. Church more than half a century ago [6]. He considered devices that transform an infinite sequence of input bits into an infinite sequence of output bits. The device is required to work “on-line”: for each input bit read, it should produce an output bit. Church asked for an algorithm that constructs such a device from a given specification. The specification language he considered is monadic second-order logic (MSOL) over natural numbers with order, $\langle \mathbb{N}, \leq \rangle$. In this case a specification is a formula $\varphi(X, Y)$, where X and Y stand for subsets of \mathbb{N} , or equivalently, infinite sequences of bits. So the formula defines a desired relation between the input sequence X and the output sequence Y .

The problem of Church is fundamentally different from decidability of MSOL theory of $\langle \mathbb{N}, \leq \rangle$. Satisfiability of the formula $\forall X. \exists Y. \varphi(X, Y)$ is just a necessary condition for the existence of a required device: not only for every input sequence there should exist a good output sequence, but moreover this sequence should be produced “on-line”. Indeed, while Büchi has shown decidability of MSOL theory of $\langle \mathbb{N}, \leq \rangle$ in 1960 [3], the solution to the synthesis problem came almost a decade later [2, 19, 20].

At the end of the eighties, Ramadge and Wonham introduced the theory of control of discrete event systems [21, 9, 4]. In this theory we start with a finite automaton, also called a plant, that defines all possible sequential behaviours of

the system. The goal is to find for a given plant another finite automaton, called controller, such that the synchronous product of the plant and the controller satisfies desired properties: MSOL properties of the language of the product.

This kind of synthesis problem would be interesting neither from theoretical nor from practical point of view if there were no additional restrictions on controllers. In the most standard form, some events of the plant are declared to be uncontrollable, and some others to be unobservable. The controller is not allowed to block uncontrollable events, and is not supposed to see unobservable events. More precisely these restrictions are determined by two subsets A_{unc} and A_{uobs} of the alphabet of events with the associated requirement that:

- (C) For every state q of the controller, and for every uncontrollable event $a \in A_{unc}$, there is a transition from q labelled by a .
- (O) For every state q of the controller, and for every unobservable event $a \in A_{uobs}$, if there is a transition from q labelled by a then this transition is a loop over q .

Ramadge and Wonham setting is more general than Church formulation. Interestingly though, the tools developed for the Church problem are sufficient to solve this case too. One important lesson here is that synthesis is ultimately about branching properties: properties of trees rather than properties of sequences. Once MSOL theory of trees is well understood, the rest is relatively easy.

At present, Ramadge and Wonham setting as described above is well established. Starting from there, many extensions have been studied: richer automata models, richer specification languages, introduction of time constraints, quantitative constraints, . . . One of the most challenging and promising directions is the extension of the framework to the distributed case. Here, one puts restrictions on the form of synthesised device: it should be distributed into several modules, each with limited capacities of observing the plant.

2 Distributed Synthesis

In a distributed system one can have multiple processes. The system specifies possible interactions between the processes and the environment, as well as, the interactions between the processes themselves. The synthesis problem is to find a program for each of the processes such that the overall behaviour of the system satisfies a given specification.

The problem can be modeled by a game with incomplete information. In such a game we have a team of controllers playing against a single player representing environment. Finding a program for each controller is then equivalent to computing a distributed winning strategy for each of the controllers. In general, multiplayer games with incomplete information are undecidable [17, 16]. For similar reasons the distributed control problem is also undecidable in most cases [18, 11, 10, 12, 14, 1]. Thanks to these works we understand some sources for undecidability, but we do not have the whole picture yet. It is fair to say

that the examples leading to undecidability can be qualified as unrealistic. It would be very interesting to refine the setting to rule out these examples, but no satisfactory proposal is known at present.

One important attempt to get a decidable framework of distributed synthesis is to change the way information is distributed in the system. In the case above, every controller sees only its inputs and its outputs. In order to deduce some information about the global state of the system a controller can use only his knowledge about the architecture and the initial state of the system. In particular, controllers are not permitted to pass additional information during communication. It is clear though that when we allow some transfer of information during communication, we give more power to controllers.

Pushing the idea of sharing information to the limit, we obtain a model where two processes involved in a communication share all the information they have about the global state of the system [8]. This point of view is not as unrealistic as it may seem at the first glance. It is rooted in the theory of traces that studies finite communicating automata with this kind of information transfer. A fundamental result of Zielonka [22, 7] implies that in fact there is a bound on the size of additional information that needs to be transferred during communication. In our terms, the theory of traces considers the case of synthesis for closed systems, i.e., systems without environment. For the distributed synthesis with environment, some decidability results for some special cases are known [8, 13, 15, 5]. Moreover, similarly to Zielonka's Theorem, these results give a bound on additional information that needs to be transferred. The decidability of the general case is open. Interestingly, the general case can be formulated as an extension of the Ramadge and Wonham setting from words, that is linear orders, to special partial orders called Mazurkiewicz traces.

References

1. A. Arnold and I. Walukiewicz. Nondeterministic controllers of nondeterministic processes. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata*, volume 2 of *Texts in Logic and Games*, pages 29–52. Amsterdam University Press, 2007.
2. J. Büchi and L. Landweber. Solving sequential conditions by finite state strategies. *Trans. Amer. Math. Soc.*, 138:367–378, 1969.
3. J. R. Büchi. On a decision method in restricted second-order arithmetic. In *Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.
4. C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
5. Th. Chatain, P. Gastin, and N. Sznajder. Natural specifications yield decidability for distributed synthesis of asynchronous systems. In *SOFSEM'09*, volume 5404 of *LNCS*, pages 141–152, 2009.
6. A. Church. Applications of recursive arithmetic to the problem of circuit synthesis. In *Summaries of the Summer Institute of Symbolic Logic*, volume I, pages 3–50. Cornell Univ., Ithaca, N.Y., 1957.

7. V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.
8. P. Gastin, B. Lerman, and M. Zeitoun. Distributed games with causal memory are decidable for series-parallel systems. In *FSTTCS*, volume 3328 of *Lecture Notes in Computer Science*, pages 275–286, 2004.
9. R. Kumar and V. K. Garg. *Modeling and control of logical discrete event systems*. Kluwer Academic Pub., 1995.
10. O. Kupferman and M. Vardi. Synthesizing distributed systems. In *Proc. 16th IEEE Symp. on Logic in Computer Science*, 2001.
11. P. Madhusudan and P. Thiagarajan. Distributed control and synthesis for local specifications. In *ICALP'01*, volume 2076 of *Lecture Notes in Computer Science*, pages 396–407, 2001.
12. P. Madhusudan and P. Thiagarajan. A decidable class of asynchronous distributed controllers. In *CONCUR'02*, volume 2421 of *Lecture Notes in Computer Science*, pages 145–160, 2002.
13. P. Madhusudan, P. S. Thiagarajan, and S. Yang. The MSO theory of connectedly communicating processes. In *FSTTCS*, volume 3821 of *LNCS*, pages 201–212, 2005.
14. S. Mohalik and I. Walukiewicz. Distributed games. In *FSTTCS*, volume 2914 of *Lecture Notes in Computer Science*, pages 338–351, 2003.
15. A. Muscholl, I. Walukiewicz, and M. Zeitoun. A look at the control of asynchronous automata. In *Perspectives in Concurrency Theory – Festschrift for P.S. Thiagarajan.*, pages 356–371. Universities Press, 2008.
16. G. Peterson, J. Reif, and S. Azhar. Lower bounds for multiplayer noncooperative games of incomplete information. *Comput. Math. Appl.*, 41:957–992, 2001.
17. G. L. Peterson and J. H. Reif. Multi-person alternation. In *Proc. IEEE FOCS*, pages 348–363, 1979.
18. A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In *31th IEEE Symposium Foundations of Computer Science (FOCS 1990)*, pages 746–757, 1990.
19. M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the AMS*, 141:1–23, 1969.
20. M. O. Rabin. *Automata on Infinite Objects and Church's Problem*. American Mathematical Society, Providence, RI, 1972.
21. P. J. G. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(2):81–98, 1989.
22. W. Zielonka. Notes on finite asynchronous automata. *RAIRO–Theoretical Informatics and Applications*, 21:99–135, 1987.