RESEARCH CENTRE

Inria Center at Rennes University

IN PARTNERSHIP WITH:

Université Rennes 1

2022

ACTIVITY REPORT

Project-Team

HYCOMES

Modélisation hybride & conception par contrats pour les systèmes embarqués multi-physiques

IN COLLABORATION WITH: Institut de recherche en informatique et systèmes aléatoires (IRISA)

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Embedded and Real-time Systems

Inria

Contents

Project-Team HYCOMES 1

1 Team members, visitors, external collaborators 3

2 Overall objectives 3

3 Research program 4
3.1 Hybrid Systems Modeling . . . 4
3.2 Background on non-standard analysis . . . 4
3.3 Structural Analysis of DAE Systems . . . 5
3.3.1 Pantelides method . . . 6
3.3.2 Pryce's Sigma-method . . . 6
3.3.3 Block triangular decomposition . . . 7
3.4 Contract-Based Design, Interfaces Theories, and Requirements Engineering . . . 7
3.5 Efficient Symbolic Computation for Sparse Systems . . . 9

4 Application domains 10
4.1 Modelica . . . 10
4.2 Dynamical Systems Verification . . . 10

5 Social and environmental responsibility 11
5.1 Impact of research results . . . 11

6 Highlights of the year 11

7 New software and platforms 12
7.1 New software . . . 12
7.1.1 IsamDAE . . . 12
7.1.2 snowflake . . . 13

8 New results 14
8.1 Handling Multimode Models and Mode Changes in Modelica . . . 14
8.2 Constraint System Decomposition . . . 14
8.3 Characterizing Q-matrices . . . 15
8.4 Characterizing Positively Invariant Sets: Inductive and Topological Methods . . . 15

9 Partnerships and cooperations 16
9.1 International research visitors . . . 16
9.1.1 Visits of international scientists . . . 16

10 Dissemination 16
10.1 Promoting scientific activities . . . 16
10.1.1 Scientific events: selection . . . 16
10.1.2 Journal . . . 16
10.2 Teaching - Supervision - Juries . . . 17
10.2.1 Teaching . . . 17
10.2.2 Supervision . . . 17
10.3 Popularization . . . 17
10.3.1 Internal or external Inria responsibilities . . . 17

11 Scientific production 17
11.1 Major publications . . . 17
11.2 Publications of the year . . . 18
11.3 Cited publications . . . 19

Project-Team HYCOMES

Creation of the Project-Team: 2016 September 01

Keywords

Computer sciences and digital sciences

A2. – Software

A2.1. – Programming Languages

A2.1.1. – Semantics of programming languages

A2.1.5. – Constraint programming

A2.1.9. – Synchronous languages

A2.1.10. – Domain-specific languages

A2.2. – Compilation

A2.2.1. – Static analysis

A2.2.8. – Code generation

A2.3. – Embedded and cyber-physical systems

A2.3.1. – Embedded systems

A2.3.2. – Cyber-physical systems

A2.3.3. – Real-time systems

A2.4. – Formal method for verification, reliability, certification

A2.4.1. – Analysis

A2.4.3. – Proofs

A2.5. – Software engineering

A2.5.1. – Software Architecture & Design

A2.5.2. – Component-based Design

A6. – Modeling, simulation and control

A6.1. – Methods in mathematical modeling

A6.1.1. – Continuous Modeling (PDE, ODE)

A6.1.5. – Multiphysics modeling

A6.3. – Computation-data interaction

A6.3.4. – Model reduction

A8. – Mathematics of computing

A8.4. – Computer Algebra

Other research topics and application domains

B4. – Energy

B4.4. – Energy delivery

B4.4.1. – Smart grids

B5.1. – Factory of the future

B5.2. – Design and manufacturing

B5.2.1. – Road vehicles

B5.2.2. – Railway

B5.2.3. – Aviation

B5.9. – Industrial maintenance

B8. – Smart Cities and Territories

B8.1. – Smart building/home

B8.1.1. – Energy for smart buildings

B8.2. – Connected city

B8.3. – Urbanism and urban planning

1 Team members, visitors, external collaborators

Research Scientists

- Benoit Caillaud [Team leader, INRIA, Senior Researcher, HDR]
- Albert Benveniste [INRIA, Emeritus, HDR]
- Khalil Ghorbal [INRIA, Researcher]

PhD Students

- Maxime Bridoux [INRIA, from Oct 2022]
- Christelle Kozaily [INRIA, until Sep 2022]
- Joan Thibault [UNIV RENNES I]

Technical Staff

- Mathias Malandain [INRIA, Engineer, full-time in Hycomes until Sep 2022; Half-time in Hycomes since Dec 2022]

Interns and Apprentices

- Carybe Bégué [ENS Rennes, Intern, until Jun 2022]
- Íñigo Íncer Romeo [University of California, Berkeley, Intern, until May 2022]

Administrative Assistant

- Armelle Mozziconacci [CNRS]

2 Overall objectives

Hycomes was created a local team of the Rennes - Bretagne Atlantique Inria research center in 2013 and has been created as an Inria Project-Team in 2016. The team is focused on two topics in cyber-physical systems design:

- Hybrid systems modeling, with an emphasis on the design of modeling languages in which software systems, in interaction with a complex physical environment, can be modelled, simulated and verified. A special attention is paid to the mathematical rigorous semantics of these languages, and to the correctness (wrt. such semantics) of the simulations and of the static analyses that must be performed during compilation. The Modelica language is the main application field. The team aims at contributing language extensions facilitating the modeling of physical domains which are poorly supported by the Modelica language. The Hycomes team is also designing new structural analysis methods for hybrid (aka. multi-mode) Modelica models. New simulation and verification techniques for large Modelica models are also in the scope of the team.
- Contract-based design and interface theories, with applications to requirements engineering in the context of safety-critical systems design. The objective of our research is to bridge the gap between system-level requirements, often expressed in natural, constrained or semi-formal languages and formal models, that can be simulated and verified.

3 Research program

3.1 Hybrid Systems Modeling

Systems industries today make extensive use of mathematical modeling tools to design computer controlled physical systems. This class of tools addresses the modeling of physical systems with models that are simpler than usual scientific computing problems by using only Ordinary Differential Equations (ODE) and Difference Equations but not Partial Differential Equations (PDE). This family of tools first emerged in the 1980's with SystemBuild by MatrixX (now distributed by National Instruments) followed soon by Simulink by Mathworks, with an impressive subsequent development.

In the early 90's control scientists from the University of Lund (Sweden) realized that the above approach did not support component based modeling of physical systems with reuse [1]. For instance, it was not easy to draw an electrical or hydraulic circuit by assembling component models of the various devices. The development of the Omola language by Hilding Elmqvist was a first attempt to bridge this gap by supporting some form of Differential Algebraic Equations (DAE) in the models. Modelica quickly emerged from this first attempt and became in the 2000's a major international concerted effort with the Modelica Consortium. A wider set of tools, both industrial and academic, now exists in this segment [2]. In the Electronic Design Automation (EDA) sector, VHDL-AMS was developed as a standard [64] and also enables the use of differential algebraic equations. Several domain-specific languages and tools for mechanical systems or electronic circuits also support some restricted classes of differential algebraic equations. Spice is the historic and most striking instance of these domain-specific languages/tools [3]. The main difference is that equations are hidden and the fixed structure of the differential algebraic results from the physical domain covered by these languages.

Despite these tools are now widely used by a number of engineers, they raise a number of technical difficulties. The meaning of some programs, their mathematical semantics, is indeed ambiguous. A main source of difficulty is the correct simulation of continuous-time dynamics, interacting with discrete-time dynamics: How the propagation of mode switchings should be handled? How to avoid artifacts due to the use of a global ODE solver causing unwanted coupling between seemingly non interacting subsystems? Also, the mixed use of an equational style for the continuous dynamics with an imperative style for the mode changes and resets, is a source of difficulty when handling parallel composition. It is therefore not uncommon that tools return complex warnings for programs with many different suggested hints for fixing them. Yet, these "pathological" programs can still be executed, if wanted so, giving surprising results — See for instance the Simulink examples in [30], [21] and [22].

Indeed this area suffers from the same difficulties that led to the development of the theory of synchronous languages as an effort to fix obscure compilation schemes for discrete time equation based languages in the 1980's. Our vision is that hybrid systems modeling tools deserve similar efforts in theory as synchronous languages did for the programming of embedded systems.

3.2 Background on non-standard analysis

Non-Standard analysis plays a central role in our research on hybrid systems modeling [21, 30, 23, 22, 28], [3]. The following text provides a brief summary of this theory and gives some hints on its usefulness in the context of hybrid systems modeling. This presentation is based on our paper [2], a chapter of Simon Bliudze's PhD thesis [36], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [71].

Non-standard numbers allowed us to reconsider the semantics of hybrid systems and propose a radical alternative to the *super-dense time semantics* developed by Edward Lee and his team as part of the Ptolemy II project, where cascades of successive instants can occur in zero time by using $\mathbb{R}_+ \times \mathbb{N}$ as a time index. In the non-standard semantics, the time index is defined as a set $\mathbb{T} = \{n\partial \mid n \in {}^*\mathbb{N}\}$, where ∂ is an *infinitesimal* and ${}^*\mathbb{N}$ is the set of *non-standard integers*. Remark that (1) $\mathbb{T}$ is dense in $\mathbb{R}_+$, making it "continuous", and (2) every $t \in \mathbb{T}$ has a predecessor in $\mathbb{T}$ and a successor in $\mathbb{T}$, making it "discrete". Although it is not effective from a computability point of view, the *non-standard semantics* provides a framework

[1] Origins of Equation-Based Modeling
[2] SimScape by Mathworks, Amesim by LMS International, now Siemens PLM, and more.
[3] Such as the Spice3 electronic circuit simulator.

that is familiar to the computer scientist and at the same time efficient as a symbolic abstraction. This makes it an excellent candidate for the development of provably correct compilation schemes and type systems for hybrid systems modeling languages.

Non-standard analysis was proposed by Abraham Robinson in the 1960s to allow the explicit manipulation of "infinitesimals" in analysis [82, 56, 52]. Robinson's approach is axiomatic; he proposes adding three new axioms to the basic Zermelo-Fraenkel (ZFC) framework. There has been much debate in the mathematical community as to whether it is worth considering non-standard analysis instead of staying with the traditional one. We do not enter this debate. The important thing for us is that non-standard analysis allows the use of the non-standard discretization of continuous dynamics "as if" it was operational.

Not surprisingly, such an idea is quite ancient. Iwasaki et al. [65] first proposed using non-standard analysis to discuss the nature of time in hybrid systems. Bliudze and Krob [37, 36] have also used non-standard analysis as a mathematical support for defining a system theory for hybrid systems. They discuss in detail the notion of "system" and investigate computability issues. The formalization they propose closely follows that of Turing machines, with a memory tape and a control mechanism.

3.3 Structural Analysis of DAE Systems

The Modelica language is based on Differential Algebraic Equations (DAE). The general form of a DAE is given by:

$$F(t, x, x', x'', \ldots) \tag{1}$$

where F is a system of n_e equations $\{f_1, \ldots, f_{n_e}\}$ and x is a finite list of n_v independent real-valued, smooth enough, functions $\{x_1, \ldots, x_{n_v}\}$ of the independent variable t. We use x' as a shorthand for the list of first-order time derivatives of x_j, $j = 1, \ldots, n_v$. High-order derivatives are recursively defined as usual, and $x^{(k)}$ denotes the list formed by the k-th derivatives of the functions x_j. Each f_i depends on the scalar t and some of the functions x_j as well as a finite number of their derivatives.

Let $\sigma_{i,j}$ denote the highest differentiation order of variable x_j effectively appearing in equation f_i, or $-\infty$ if x_j does not appear in f_i. The *leading variables* of F are the variables in the set

$$\left\{ x_j^{(\sigma_j)} \mid \sigma_j = \max_i \sigma_{i,j} \right\}$$

The *state variables* of F are the variables in the set

$$\left\{ x_j^{(\nu_j)} \mid 0 \leq \nu_j < \max_i \sigma_{i,j} \right\}$$

A leading variable $x_j^{(\sigma_j)}$ is said to be *algebraic* if $\sigma_j = 0$ (in which case, neither x_j nor any of its derivatives are state variables). In the sequel, v and u denote the leading and state variables of F, respectively.

DAE are a strict generalization of *ordinary differential equations* (*ODE*), in the sense that it may not be immediate to rewrite a DAE as an explicit ODE of the form $v = G(u)$. The reason is that this transformation relies on the Implicit Function Theorem, requiring that the Jacobian matrix $\frac{\partial F}{\partial v}$ have full rank. This is, in general, not the case for a DAE. Simple examples, like the two-dimensional fixed-length pendulum in Cartesian coordinates [79], exhibit this behaviour.

For a square DAE of dimension n (i.e., we now assume $n_e = n_v = n$) to be solved in the neighborhood of some (v^*, u^*), one needs to find a set of non-negative integers $C = \{c_1, \ldots, c_n\}$ such that system

$$F^{(C)} = \{f_1^{(c_1)}, \ldots, f_n^{(c_n)}\}$$

can locally be made explicit, i.e., the Jacobian matrix of $F^{(C)}$ with respect to its leading variables, evaluated at (v^*, u^*), is nonsingular. The smallest possible value of $\max_i c_i$ for a set C that satisfies this property is the *differentiation index* [45] of F, that is, the minimal number of time differentiations of all or part of the equations f_i required to get an ODE.

In practice, the problem of automatically finding a "minimal" solution C to this problem quickly becomes intractable. Moreover, the differentiation index may depend on the value of (v^*, u^*). This is why, in lieu of numerical nonsingularity, one is interested in the *structural nonsingularity* of the Jacobian

matrix, i.e., its almost certain nonsingularity when its nonzero entries vary over some neighborhood. In this framework, the *structural analysis* (*SA*) of a DAE returns, when successful, values of the c_i that are independent from a given value of (v^*, u^*).

A renowned method for the SA of DAE is the *Pantelides method*; however, Pryce's Σ*-method* is introduced also in what follows, as it is a crucial tool for our works.

3.3.1 Pantelides method

In 1988, Pantelides proposed what is probably the most well-known SA method for DAE [79]. The leading idea of his work is that the structural representation of a DAE can be condensed into a bipartite graph whose left nodes (resp. right nodes) represent the equations (resp. the variables), and in which an edge exists if and only if the variable occurs in the equation.

By detecting specific subsets of the nodes, called *Minimally Structurally Singular* (*MSS*) subsets, the Pantelides method iteratively differentiates part of the equations until a perfect matching between the equations and the leading variables is found. One can easily prove that this is a necessary and sufficient condition for the structural nonsingularity of the system.

The main reason why the Pantelides method is not used in our work is that it cannot efficiently be adapted to multimode DAE (mDAE). As a matter of fact, the adjacency graph of a mDAE has both its nodes and edges parametrized by the subset of modes in which they are active; this, in turn, requires that a parametrized Pantelides method must branch every time no mode-independent MSS is found, ultimately resulting, in the worst case, in the enumeration of modes.

3.3.2 Pryce's Sigma-method

Albeit less renowned that the Pantelides method, Pryce's Σ-method [80] is an efficient SA method for DAE, whose equivalence to the Pantelides method has been proved by the author. This method consists in solving two successive problems, denoted by primal and dual, relying on the Σ*-matrix*, or *signature matrix*, of the DAE F.

This matrix is given by:

$$\Sigma = (\sigma_{ij})_{1 \leq i,j \leq n} \tag{2}$$

where σ_{ij} is equal to the greatest integer k such that $x_j^{(k)}$ appears in f_i, or $-\infty$ if variable x_j does not appear in f_i. It is the adjacency matrix of a weighted bipartite graph, with structure similar to the graph considered in the Pantelides method, but whose edges are weighted by the highest differentiation orders. The $-\infty$ entries denote non-existent edges.

The *primal problem* consists in finding a *maximum-weight perfect matching* (*MWPM*) in the weighted adjacency graph. This is actually an assignment problem, for the solving of which several standard algorithms exist, such as the push-relabel algorithm [63] or the Edmonds-Karp algorithm [58] to only give a few. However, none of these algorithms are easily parametrizable, even for applications to mDAE systems with a fixed number of variables.

The *dual problem* consists in finding the component-wise minimal solution $(C, D) = (\{c_1, \ldots, c_n\}, \{d_1, \ldots, d_n\})$ to a given linear programming problem, defined as the dual of the aforementioned assignment problem. This is performed by means of a *fixpoint iteration* (*FPI*) that makes use of the MWPM found as a solution to the primal problem, described by the set of tuples $\{(i, j_i)\}_{i \in \{1,\ldots,n\}}$:

1. Initialize $\{c_1, \ldots, c_n\}$ to the zero vector.
2. For every $j \in \{1, \ldots, n\}$,
$$d_j \leftarrow \max_i(\sigma_{ij} + c_i)$$
3. For every $i \in \{1, \ldots, n\}$,
$$c_i \leftarrow d_{j_i} - \sigma_{i,j_i}$$
4. Repeat Steps 2 and 3 until convergence is reached.

From the results proved by Pryce in [80], it is known that the above algorithm terminates if and only if it is provided a MWPM, and that the values it returns are independent of the choice of a MWPM whenever there exist several such matchings. In particular, a direct corollary is that the Σ-method succeeds as long as a perfect matching can be found between equations and variables.

Another important result is that, if the Pantelides method succeeds for a given DAE F, then the Σ-method also succeeds for F and the values it returns for C are exactly the differentiation indices for the equations that are returned by the Pantelides method. As for the values of the d_j, being given by $d_j = \max_i(\sigma_{ij} + c_i)$, they are the differentiation indices of the leading variables in $F^{(C)}$.

Working with this method is natural for our works, since the algorithm for solving the dual problem is easily parametrizable for dealing with multimode systems, as shown in our recent paper [42].

3.3.3 Block triangular decomposition

Once structural analysis has been performed, system $F^{(C)}$ can be regarded, for the needs of numerical solving, as an algebraic system with unknowns $x_j^{(d_j)}$, $j = 1 \dots n$. As such, (inter)dependencies between its equations must be taken into account in order to put it into block triangular form (BTF). Three steps are required:

1. the *dependency graph* of system $F^{(C)}$ is generated, by taking into account the perfect matching between equations $f_i^{(c_i)}$ and unknowns $x_j^{(d_j)}$;
2. the *strongly connected components* (*SCC*) in this graph are determined: these will be the *equation blocks* that have to be solved;
3. the *block dependency graph* is constructed as the condensation of the dependency graph, from the knowledge of the SCC; a BTF of system $F^{(C)}$ can be made explicit from this graph.

3.4 Contract-Based Design, Interfaces Theories, and Requirements Engineering

System companies such as automotive and aeronautic companies are facing significant difficulties due to the exponentially raising complexity of their products coupled with increasingly tight demands on functionality, correctness, and time-to-market. The cost of being late to market or of imperfections in the products is staggering as witnessed by the recent recalls and delivery delays that many major car and airplane manufacturers had to bear in the recent years. The root causes of these design problems are complex and relate to a number of issues ranging from design processes and relationships with different departments of the same company and with suppliers, to incomplete requirement specification and testing.

We believe the most promising means to address the challenges in systems engineering is to employ formal design methodologies that seamlessly and coherently combine the various viewpoints of the design space (behavior, time, energy, reliability, ...), that provide the appropriate abstractions to manage the inherent complexity, and that can provide correct-by-construction implementations. The following issues must be addressed when developing new approaches to the design of complex systems:

- The overall design flows for heterogeneous systems and the associated use of models across traditional boundaries are not well developed and understood. Relationships between different teams inside a same company, or between different stake-holders in the supplier chain, are not supported by precise mathematical specifications of the components each party is expected to deliver.
- System requirements capture and analysis is in large part a heuristic process, where informal text and natural language-based techniques in use today are facing significant challenges [67]. Formal requirements engineering is in its infancy: mathematical models, formal analysis techniques and links to system implementation must be developed.
- Dealing with variability, uncertainty, and life-cycle issues, such as extensibility of a product family, are not well-addressed using available systems engineering methodologies and tools.

The challenge is to address the entire process and not to consider only local solutions of methodology, tools, and models that ease part of the design.

Contract-based design has been proposed as a new approach to the system design problem that is rigorous and effective in dealing with the problems and challenges described before, and that, at the same time, does not require a radical change in the way industrial designers carry out their task as it cuts across design flows of different types. Indeed, contracts can be used almost everywhere and at nearly all stages of system design, from early requirements capture, to embedded computing infrastructure and detailed design involving circuits and other hardware. Intuitively, a contract captures two properties, respectively representing the assumptions on the environment and the guarantees of the system under these assumptions. Hence, a contract can be defined as a pair $C = (A, G)$ of assumptions and guarantees characterizing in a formal way 1) under which context the design is assumed to operate, and 2) what its obligations are. Assume/Guarantee reasoning has been known for a long time, and has been used mostly in software engineering [77]. However, contract-based design is not limited to types and values in a piece of software. It can also be used to capture its performances (time, memory consumption, energy) and reliability. This amounts to enrich a component's interface with, on one hand, formal specifications of the behavior of the environment in which the component may be instantiated and, on the other hand, of the expected behavior of the component itself. To leverage contract-based reasoning as a technique of choice for system engineers, we aim to develop:

- mathematical foundations of contracts, that enable the design of formal verification frameworks;
- System engineering methodologies and tools, that focus on requirements modeling, contract specification and verification, at multiple abstraction levels.

A detailed bibliography on contract and interface theories for embedded system design can be found in [4]. In a nutshell, contract and interface theories fall into two main categories:

Assume/guarantee contracts. By explicitly relying on the notions of assumptions and guarantees, A/G-contracts are intuitive. This makes them appealing for the engineer. In A/G-contracts, assumptions and guarantees are just properties regarding the behavior of a component and of its environment. The typical case is when these properties are formal languages or sets of traces. This includes the class of safety properties [68, 48, 76, 20, 50]. Contract theories were initially developed as specification formalisms able to refuse some inputs from the environment [57]. A/G-contracts were advocated in [31] and are is still a very active research topic, with several contributions dealing with the timed [35] and probabilistic [43, 44] viewpoints in system design, and even hybrid systems design [78].

Automata theoretic interfaces. Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch's Input/Output Automata [75, 74]. Interface Automata [16, 15, 17, 46] focus primarily on parallel composition and compatibility: two interfaces are compatible if there exists at least one environment where they can work together. The idea is that the resulting composition exposes as an interface the needed information to ensure that incompatible pairs of states cannot be reached. This can be achieved by using the possibility, for an Interface Automaton, to refuse some inputs from the environment in a given state. This amounts to the implicit assumption that the environment will never produce any of the refused inputs, when the interface is in this state. Modal Interfaces [81] inherit from both Interface Automata and the originally unrelated notion of Modal Transition System [70, 19, 38, 69]. Modal Interfaces are strictly more expressive than Interface Automata by decoupling the I/O orientation of an event and its deontic modalities (mandatory, allowed or forbidden). Informally, a *must* transition is offered in every component that realizes the modal interface, while a *may* transition is optional. Research on interface theories is still very active. For instance, timed [18, 32, 34, 54, 53, 33], probabilistic [43, 55] and energy-aware [47] interface theories have been proposed recently.

Requirements Engineering is one of the major concerns in large systems industries today, particularly so in sectors where certification prevails [83]. Most requirements engineering tools offer a poor structuring of the requirements and cannot be considered as formal modeling frameworks today. They are nothing less, but nothing more than an informal structured documentation enriched with hyperlinks.

We see Contract-Based Design and Interfaces Theories as innovative tools in support of Requirements Engineering. The Software Engineering community has extensively covered several aspects of Requirements Engineering, in particular:

- the development and use of large and rich *ontologies*; and
- the use of Model Driven Engineering technology for the structural aspects of requirements and resulting hyperlinks (to tests, documentation, PLM, architecture, and so on).

Behavioral models and properties, however, are not properly encompassed by the above approaches. This is the cause of a remaining gap between this phase of systems design and later phases where formal model based methods involving behavior have become prevalent. We believe that our work on contract-based design and interface theories is best suited to bridge this gap.

3.5 Efficient Symbolic Computation for Sparse Systems

This project consists in exploiting the parsimony of sparse systems to accelerate their symbolic manipulation (quantifiers elimination [51], differential-algebraic reductions [84] etc.). Let us cite two typical examples as a motivation: Boolean functions $(a \vee b \wedge \neg c)$ and polynomial systems with inequalities $(x^2 + y \leq 1 \wedge x + y = 0)$. We seek precisely to decompose these systems, automatically, in order to be able to manipulate them at an advantageous computational cost (in time and in memory) by attacking the pieces thus obtained rather than considering the system as a single monolithic block.

The current algorithms suffer from a theoretical complexity that is at best exponential (in the size of the input) limiting their use to instances of very modest size. The classic approach to overcome this problem is to develop/use numerical methods (with their limits and intrinsic problems) when possible of course. We aim to explore a different avenue.

In this project, we wish to exploit the structure of sparse systems to push the symbolic approach beyond its theoretical limits (for this class). The a priori limited application of our methods for dense systems is compensated by the fact that in practice, the problems are very often structured (in this regard, let us content ourselves with quoting the SAT solvers which successfully tackle industrial instances of a theoretically NP-complete problem).

The idea of exploiting the structure to speed up calculations that are a priori complex is not new. It has notably been developed and successfully used in signal processing via Factor Graphs [73], where one restricts oneself to local propagation of information, guided by an abstract graph which represents the structure of the system overall. Our approach is similar: we basically seek to use expensive algorithms sparingly on only subsystems involving only a small number of variables, thus hoping to reduce the theoretical worst case. One could then legitimately wonder why it is not enough to apply what has already been done on Factor Graphs? The difficulty (and the novelty for that matter) lies in the implementation of this idea for the problems that interest us. Let's start by emphasizing that the propagation of information has a significantly different impact depending on the operator (or quantifier) to be eliminated: a minimization or a summation do not look like a projection at all! This will obviously not prevent us from importing good ideas applicable to our problems and vice versa.

More related to symbolic computation, to our knowledge, at least two recent attempts exist: chordal networks [49] which propose a representation of the ideals of the ring of polynomials (therefore algebraic sets), and triangular block shapes [86], initiated independently and under development in our team and which tackle Boolean functions, or, if you will, the algebraic sets over the field of Booleans. The similarity between the two approaches is striking and suggests that there is a common way of doing things that could be exploited beyond these two examples. It is this unification that interests us in the first place in this project.

We identify three research problems to explore:

T1. Unify several optimization problems on graphs as a single problem parameterized by a cost function.

T2. Adapt (and possibly improve) the algorithm of [85] to WAP and consequently to all instances of the single problem detailed in T1.

T3. Propose a unified and modular method consisting of: (1) an elimination algorithm, (2) a data structure and (3) an efficient algorithm to solve the problem (with a cost function adequate).

The work on chordal networks and our work on Boolean functions immediately become special cases. For example, for Boolean functions, one could use Binary Decision Diagrams (BDDs) [39] to represent each piece of the initial system thus obtained. In fact, the final representation will no longer be a single monolithic BDD as is currently the case, but rather a graph of BDDs. In the same way, an algebraic set will be represented by a graph where each node is a Gröbner basis (or any other data structure used to represent systems of equations).

The structure of the system becomes thus apparent and is exploited to optimize the used representation, opening the way to a better understanding and therefore to a more efficient and better targeted manipulation. Let's remember a simple fact here: symbolic manipulation often solves the problem exactly (without approximation or compromise). Therefore, pushing the limits of applicability of these techniques to scale them can only be appreciated and will undoubtedly have a significant impact on all the areas where they apply and the list is as long as it is varied. (compilation, certification, validation, synthesis, etc.).

4 Application domains

The Hycomes team contributes to the design of mathematical modeling languages and tools, to be used for the design of cyberphysical systems. In a nutshell, two major applications can be clearly identified: (i) our work on the structural analysis of multimode DAE systems has a sizeable impact on the techniques to be used in Modelica tools; (ii) our work on the verification of dynamical systems has an impact on the design methodology for safety-critical cyberphysical systems. These two applications are detailed below.

4.1 Modelica

Mathematical modeling tools are a considerable business, with major actors such as MathWorks, with Matlab/Simulink, or Wolfram, with Mathematica. However, none of these prominent tools are suitable for the engineering of large systems. The Modelica language has been designed with this objective in mind, making the best of the advantages of DAEs to support a component-based approach. Several industries in the energy sector have adopted Modelica as their main systems engineering language.

Although multimode features have been introduced in version 3.3 of the language [59], proper tool support of multimode models is still lagging behind. The reason is not a lack of interest from tool vendors and academia, but rather that multimode DAE systems poses several fundamental difficulties, such as a proper definition of a concept of solutions for multimode DAEs, how to handle mode switchings that trigger a change of system structure, or how impulsive variables should be handled. Our work on multimode DAEs focuses on these crucial issues [29].

Thanks to our IsamDAE software [42, 40], a larger class of Modelica models are expected to be compiled and simulated correctly. This should enable industrial users to have cleaner and simpler multimode Modelica models, with dynamically changing structure of cyberphysical systems. On the longer term, our ambition is to provide efficient code-generation techniques for the Modelica language, supporting, in full generality, multimode DAE systems, with dynamically changing differentiation index, structure and dimension.

4.2 Dynamical Systems Verification

In addition to well-defined operational semantics for hybrid systems, one often needs to provide formal guarantees about the behavior of some critical components of the system, or at least its main underlying logic. To do so, we are actively developing new techniques to automatically verify whether a hybrid system complies with its specifications, and/or to infer automatically the envelope within which the system behaves safely. The approaches we developed have been already successfully used to formally verify the intricate logic of the ACAS X, a mid-air collision avoidance system that advises the pilot to go upward or downward to avoid a nearby airplane which requires mixing the continuous motion of the aircraft with the discrete decisions to resolve the potential conflict [66]. This challenging example is nothing

but an instance of the kind of systems we are targeting: autonomous smart systems that are designed to perform sophisticated tasks with an internal tricky logic. What is even more interesting perhaps is that such techniques can be often "reverted" to actually synthesize missing components so that some property holds, effectively helping the design of such complex systems.

5 Social and environmental responsibility

5.1 Impact of research results

The expected impact of our research is to allow both better designs and better exploitation of energy production units and distribution networks, enabling large-scale energy savings. At least, this is what we could observe in the context of the FUI ModeliScale collaborative project (2018–2021), focused on electric grids, urban heat networks and building thermal modeling.

The rationale is as follows: system engineering models are meant to assess the correctness, safety and optimality of a system under design. However, system models are still useful after the system has been put in operation. This is especially true in the energy sector, where systems have an extremely long lifespan (for instance, more than 50 years for some nuclear power plants) and are upgraded periodically, to integrate new technologies. Exactly like in software engineering, where a software and its model co-evolve throughout the lifespan of the software, a co-evolution of the system and its physical models has to be maintained. This is required in order to maintain the safety of the system, but also its optimality.

Moreover, physical models can be instrumental to the optimal exploitation of a system. A typical example are model-predictive control (MPC) techniques, where the model is simulated, during the exploitation of the system, in order to predict system trajectories up to a bounded-time horizon. Optimal control inputs can then be computed by mathematical programming methods, possibly using multiple simulation results. This has been proved to be a practical solution [62], whenever classical optimal control methods are ineffective, for instance, when the system is non-linear or discontinuous. However, this requires the generation of high-performance simulation code, capable of simulating a system much faster than real-time.

The structural analysis techniques implemented in IsamDAE [42] generate a conditional block dependency graph, that can be used to generate high-performance simulation code : static code can be generated for each block of equations, and a scheduling of these blocks can be computed, at runtime, at each mode switching, thanks to an inexpensive topological sort algorithm. Contrarily to other approaches (such as [61]), no structural analysis, block-triangular decompositions, or automatic differentiation has to be performed at runtime.

6 Highlights of the year

Members of the Hycomes team have contributed to two journal papers in 2022:

- An extended version of our three Modelica'21 papers [27, 26, 41] has been assembled and published as a 63 pages long journal paper [7] —more details in Section 8.1. This paper also details the use of CoSTreD [14] (see also Section 8.2), a message-passing technique, decomposing the resolution of constraint systems into the resolution of several, smaller systems, and that turns out to be instrumental to reduce the empirical computational complexity of the multimode Pryce index-reduction method, implemented in the IsamDAE software (Section 7.1.1). An open-source implementation of CoSTreD is available in the Snowflake OCaml library (Section 7.1.2).

- Two characterizations of positive invariance of sets for systems of ordinary differential equations are proposed in [8]. Although these characterizations are essentially equivalent, they lead to different decision procedures for polynomial differential equations —see Section 8.4.

7 New software and platforms

7.1 New software

7.1.1 IsamDAE

Name: Implicit Structural Analysis of Multimode DAE systems

Keywords: Structural analysis, Differential algebraic equations, Multimode, Scheduling, Consistent initialization, Code generation

Scientific Description: Modeling languages and tools based on Differential Algebraic Equations (DAE) bring several specific issues that do not exist with modeling languages based on Ordinary Differential Equations. The main problem is the determination of the differentiation index and latent equations. Prior to generating simulation code and calling solvers, the compilation of a model requires a structural analysis step, which reduces the differentiation index to a level acceptable by numerical solvers.

The Modelica language, among others, allows hybrid models with multiple modes, mode-dependent dynamics and state-dependent mode switching. These Multimode DAE (mDAE) systems are much harder to deal with. The main difficulties are (i) the combinatorial explosion of the number of modes, and (ii) the correct handling of mode switchings.

The IsamDAE software aims at providing a compilation chain for mDAE-based modeling languages that make it possible to efficiently generate correct simulation code for multimode models. Novel structural analysis methods for mDAE systems were designed and implemented, based on an implicit representation of the varying structure of such systems. Several standard algorithms, such as J. Pryce's Sigma-method and the Dulmage-Mendelsohn decomposition, were adapted to the multimode case, using Binary Decision Diagrams (BDD) to represent the mode-dependent structure of an mDAE system.

IsamDAE determines, as a function of the mode, the set of latent equations, the leading variables and the state vector. This is then used to compute a conditional dependency graph (CDG) of the system, that can be used to generate simulation code with a mode-dependent scheduling of the blocks of equations. The software is also fit for generating simulation code for the hybrid dynamical system simulation tool Siconos, as well as handling the structural analysis of the multimode consistent initialization problem associated with an mDAE system.

Functional Description: IsamDAE (Implicit Structural Analysis of Multimode DAE systems) is a software library implementing new structural analysis methods for multimode DAE systems, based on an implicit representation of incidence graphs, matchings between equations and variables, and block decompositions. The input of the software is a variable dimension multimode DAE system consisting in a set of guarded equations and guarded variable declarations. It computes a mode-dependent structural index reduction of the multimode system and is able to produce a mode-dependent graph for the scheduling of blocks of equations in long modes, check the structural nonsingularity of the associated consistent initialization problem, or generate simulation code for the nonsmooth dynamical system simulation tool Siconos.

IsamDAE is coded in OCaml, and uses the following packages: GuaCaml by Joan Thibault, MLBDD by Arlen Cox, Menhir by François Pottier and Yann Régis-Gianas, Pprint by François Pottier, Snowflake by Joan Thibault, XML-Light by Nicolas Cannasse and Jacques Garrigue.

Release Contributions: New features:

* XML representations of the structure of a multimode DAE model are accepted as inputs by the IsamDAE tool, in order to enable weak coupling with tools based on existing DAE-based languages. IsamDAE distinguishes between MEL and XML inputs based on the extension of the input file (.mel versus .mdae.xml).

Bug fixes:

* A better handling of the model structure for consistent initialization prevents subtle bugs that were observed for a few models and initial events. Specific error messages are returned when initial equations involve variables that are not active in the corresponding modes.

Performance improvement:

* Better handling of sets of equations/variables labeled with propositional formulas, thanks to an adapted data structure.

Various:

* Verbosity option -v now takes as a parameter an integer ranging from 0 ("quiet") to 5 ("deep debug"). The detailed output of CoSTreD is only available in "deep debug" mode.

News of the Year: XML inputs representing the mode-dependent structure of a multimode DAE system are now handled by IsamDAE, enabling for the weak coupling with existing modeling and simulation tools for DAE-based languages such as Modelica.

URL: https://team.inria.fr/hycomes/software/isamdae/

Publications: hal-03768331, hal-02572879, hal-03320499, hal-02476541

Contact: Benoit Caillaud

Participants: Benoit Caillaud, Mathias Malandain, Joan Thibault, Alexandre Rocca, Bertrand Provot

7.1.2 snowflake

Name: Snowflake : A Generic Symbolic Dynamic Programming framework

Keywords: Ocaml, Symbolic computation, Binary decision diagram

Scientific Description: Complex systems (either physical or logical) are structured and sparse, that is, they are build from individual components linked together, and any component is only linked to rather small number of other components with respects to the size of the global system.

RBTF exploits this structure, by over-approximating the relations between components as a tree (called decomposition tree in the graph literature) each node of this tree being a set of components of the initial systems. Then, starting from leaves, each sub-system is solved and the solutions are projected as a new constraints on their parents node, this process is iterated until all sub-systems are solved. This step allows to condensate all constraints and check their satisfiability. We call this step the **Forward Reduction Process** (FRP).

Finally, we can propagate all the constraints back into their initial sub-system by performing those same projection in the reverse direction. That is, each sub-system update its set of solution given the information from its parent then send the information to its children sub-systems (possibly none, if its a leaf). We call this step the **Backward Propagation Process** (BPP).

Functional Description: Snowflake interfaces a WAP-solver (Weighted Adjacency Propagation problem), a functor-based implementation of CoSTreD (Constraint System Tree Decomposition), along with a minimalist MLBDD (Arlen Cox's BDD package) toolbox.

Release Contributions: 2022/07 : published Research Report 9478 (https://hal.archives-ouvertes.fr/hal-03740562/) 2022/06/30 : renamed RBTF into CoSTreD 2022/06/19 : added basic constraint system export 2022/06/02 : add small graphviz interface 2022/06/02 : added small graphviz interface 2022/06/02 : added sorted test on input to MlbddUtils.subst

URL: https://gitlab.com/boreal-ldd/snowflake/-/wikis/home

Authors: Joan Thibault, Joan Thibault

Contact: Joan Thibault

8 New results

8.1 Handling Multimode Models and Mode Changes in Modelica

Participants: Albert Benveniste, Benoît Caillaud, Mathias Malandain, Joan Thibault.

Since version 3.3, the Modelica language offers the possibility of specifying *multimode dynamics*, by describing state machines with different DAE dynamics in each different state [60]. This feature enables describing large complex cyber-physical systems with different behaviors in different modes.

While being undoubtedly valuable, multimode modeling has been the source of serious difficulties for non-expert users of the current generation of Modelica tools. Indeed, while many large-scale Modelica models are properly handled, some physically meaningful models do not result in correct simulations with most Modelica tools. As such problematic models are actually easy to construct, the likelihood of such bad cases occurring in large models is significant.

It is unfortunately unclear which multimode Modelica models will be properly handled, and which ones will fail. As a consequence, quite often, end users have to ask Modelica experts, or even tool developers themselves, to tweak their models in order to make them work as expected. While it is accepted that physical modeling itself requires expertise, requiring expertise in how to get around tool idiosyncrasies is not desirable. This situation hinders the dissemination of Modelica tools among a larger class of users, such as Simulink-trained engineers.

Several examples, presented in [7] reveal that this problem is due to an inadequate structural analysis, performed during compilation. As far as we know, no industrial-strength Modelica tool implements a mode-dependent structural analysis. Worse, it is not even understood what kind of structural analysis should be associated with mode change events.

Some years ago, we started a project aiming at addressing all the above issues [25, 24, 29]. In [7], we cast our approach in the context of the Modelica language, by illustrating it on two simple yet physically meaningful examples that current Modelica tools fail to properly simulate. The use of nonstandard analysis allows us to perform the analysis of both modes and mode changes in a unified framework, including the handling of transient modes and that of impulsive mode changes. Standardization techniques are then used in order to generate effective code for restarts at mode changes.

As an efficient implementation of such methods in Modelica compilers would greatly expand the class of multimode models amenable to reliable numerical simulation, multimode DAE structural analysis algorithms are also detailed in [7]. This extends previous work presented in [42]: mode enumeration is avoided thanks to the use of an implicit, BDD-based symbolic representations of the structure of a multimode DAE system. However, the scalability of the algorithm is greatly improved thanks to the use of CoSTreD [14], a message-passing technique, that allows to decompose the resolution of the primal problem of the multimode Pryce method into a set of smaller parametric optimization problems —more details in Section 8.2.

A compile-time calculus that evaluates the impulse order of algebraic variables is also detailed in [7]. Finite impulse orders can be used to renormalize impulsive variables when implementing a numerical scheme that approximates the restart values for each state variable of the system. We also detail in this paper, a systematic way of rewriting a multimode Modelica model, based on the results of a multimode structural analysis. The rewritten Modelica model is guaranteed to have a reduced index and a mode-independent structure. This suffices to guarantee correctly compiled by state-of-the-art Modelica tools. Simulation results are presented on a simple, yet meaningful, physical system whose original Modelica model is not correctly handled by state-of-the-art Modelica tools.

We demonstrate how the results of this multimode structural analysis can be used for transforming a multimode Modelica model into its RIMIS (Reduced Index Mode-Independent Structure) form, which is guaranteed to yield correct execution on state-of-the-art Modelica tools.

8.2 Constraint System Decomposition

Participants: Joan Thibault.

Various classical problems in computer science can be formulated as Constraint Solving Problems (CSP), consisting in a query on a conjunction of constraints. Typical instances of such queries are satisfiability problems, optimization under constraints, model enumeration, model counting and normalization. Constraint systems can be Conjunctive Normal Form (CNF) formulas, as well as Integer Linear Programs (ILP), and, in its most generic form, Constraint Programs (CP). In both industrial and academic contexts, instances are generally structured and, in most cases, sparse: each constraint involves only a small set of variables, and variables are only involved in a small set of constraints. Moreover, large practical instances tend to have a tree-like structure, which can efficiently be captured by the notion of treewidth, as commonly considered in the fixed-parameter tractability community. Using dynamic programming to solve problems for which a "good" tree decomposition is available is well known, and has been rediscovered many times in the history of computer science, under various names: message passing in factor graphs, belief propagation in belief networks, arc consistency in constraint networks, etc. In [14], we introduce the CoSTreD (Constraint System Tree Decomposition) method, based on symbolic representations and operators on them to improve the scalability of CSP solving. CoSTreD is based upon two operators: a projection operator which allows to deal with satisfiability and canonicalization locally on the tree decomposition, and a co-projection operator, extending the method to optimization queries. We establish sufficient conditions under which these operators preserve the semantics of the CSP. Finally, CoSTreD is extended to deal with parameter (or mode) variables, mostly by (i) adapting the notion of tree decomposition to deal with parameter variables, (ii) using symbolic representations to avoid the combinatorial explosion of mode enumeration, and (iii) mitigating the contamination of constraints by parameter variables during message passing.

8.3 Characterizing Q-matrices

Participants: Khalil Ghorbal, Christelle Kozaily.

In [13], we provide a geometric equivalent reformulation of a relatively old, yet unsolved, problem that originated in the optimization community: under which conditions on the $n \times n$ matrix M , does the so called linear complementarity problem given by $w - Mz = q$, $w, z \geq 0$, and $w.z = 0$, have a solution (w, z) for all vectors $q \in \mathbb{R}^n$. If the latter property holds, the matrix M is said to be a Q-matrix. We have shown that the existence of solutions amounts to a covering (not necessarily a partition) of the entire space by a set of finite cones defined by the involved vectors as well as the standard basis. We give a full characterization in dimension 3 by reducing the problem to several similar (and well-understood) problems on dimension 2.

8.4 Characterizing Positively Invariant Sets: Inductive and Topological Methods

Participants: Khalil Ghorbal.

In [8], we present two characterizations of positive invariance of sets for systems of ordinary differential equations. The first characterization uses inward sets which intuitively collect those points from which the flow evolves within the set for a short period of time, whereas the second characterization uses the notion of exit sets, which intuitively collect those points from which the flow immediately leaves the set. Our proofs emphasize the use of the real induction principle as a generic and unifying proof technique that captures the essence of the formal reasoning justifying our results and provides cleaner alternative proofs of known results. The two characterizations presented in this article, while essentially equivalent, lead to two rather different decision procedures (termed respectively LZZ and

ESE) for checking whether a given semi-algebraic set is positively invariant under the flow of a system of polynomial ordinary differential equations. The procedure LZZ improves upon the original work by Liu, Zhan and Zhao [72]. The procedure ESE, introduced in this article, works by splitting the problem, in a principled way, into simpler sub-problems that are easier to check, and is shown to exhibit substantially better performance compared to LZZ on problems featuring semi-algebraic sets described by formulas with non-trivial Boolean structure.

9 Partnerships and cooperations

9.1 International research visitors

9.1.1 Visits of international scientists

Participants: Albert Benveniste, Íñigo Íncer Romeo.

Íñigo Íncer Romeo, PhD student at UC Berkeley (CA, USA), visited the Hycomes team from December 2021 until May 2022. His internship has been funded by a Chateaubriand grant of the French Consulate in San Francisco. During his stay, he worked with Albert Benveniste on topics related to Contract-based Design method and more particularly on Hypercontracts [10].

10 Dissemination

10.1 Promoting scientific activities

10.1.1 Scientific events: selection

Member of the conference program committees

- Khalil Ghorbal. PC Member. Hybrid Systems: Computation and Control (HSCC) 2022.
- Khalil Ghorbal. PC Member. International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS) 2023.
- Benoît Caillaud has served on the program committee of the FDL'22 conference (Forum on specification & Design Languages, Linz, Austria, September 2022).

Reviewer

- Benoît Caillaud has evaluated collaborative project proposals for the ANR (French national research funding agency).
- Benoît Caillaud has reviewed an application for the Caseau EDF / French Academy of Technologies Best PhD Award 2022.
- Benoît Caillaud has reviewed papers for the following conferences and workshops: TACAS'22, WODES'22.

10.1.2 Journal

Member of the editorial boards

- Benoît Caillaud has been appointed member of the editorial boards of the Cambridge University Press, Research Directions: Cyber-Physical Systems and of the MDPI Computation journals.

Reviewer - reviewing activities

- Benoît Caillaud has reviewed a paper for the ACM TECS (Transactions on Embedded Computing Systems) journal.
- Khalil Ghorbal has reviewed a paper for the Journal of Automated Reasoning (JAR).
- Khalil Ghorbal has reviewed a paper for the Journal of Theoretical Computer Science (TCS).

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- Master : Khalil Ghorbal, Category Theory, Monads, and Computation, M2, (enseignant principal), 30h EqTD, ENS Rennes, France
- Agregation informatique : Khalil Ghorbal, oraux blancs et preparations de cours. 20h EqTD, ENS Rennes, France.

10.2.2 Supervision

- Maxime Bridoux is a 1st year PhD student (AEx Backbone) supervised by Khalil Ghorbal and Benoît Caillaud. He is currently working on effective data structures for storing and querying polynomials systems.
- Christelle Kozaily is a 4th/5th year PhD student, supervised by Khalil Ghorbal. She is currently writing her thesis. Christelle Kozaily worked on a particular class of hybrid systems known as linear complementarity systems and was primarily interested in the non-smoothness of they underlying spaces (Section 8.3).
- Joan Thibault is a 3rd/4th year PhD student, supervised by Benoît Caillaud and Khalil Ghorbal. His research is on efficient and scalable data-structures for solving constraint systems and some optimization problems on them (Section 8.2), with applications in multimode DAE systems structural analysis (Section 8.1).

10.3 Popularization

Joan Thibault participated to MT180 (*Ma these en 180s*) in February 2022.

10.3.1 Internal or external Inria responsibilities

Khalil Ghorbal is the main organizer of 68NQRT, the seminar of the Language and Software Engineering department of the IRISA UMR (Rennes).

The programs of the previous years are available online (abstract, slides, and playbacks). For instance the program from October 2020 till June 2021 can be found here. The seminar's frequency (on average over the academic year) is twice a month.

11 Scientific production

11.1 Major publications

[1] A. Benveniste, T. Bourke, B. Caillaud, J.-L. Colaço, C. Pasteur and M. Pouzet. 'Building a Hybrid Systems Modeler on Synchronous Languages Principles'. In: *Proceedings of the IEEE.* Design Automation for Cyber-Physical Systems 106.9 (Sept. 2018), pp. 1568–1592. DOI: 10.1109/JPROC.2018.2858016. URL: https://hal.inria.fr/hal-01879026.

[2] A. Benveniste, T. Bourke, B. Caillaud and M. Pouzet. 'Non-standard semantics of hybrid systems modelers'. English. In: *Journal of Computer and System Sciences* 78.3 (2012). This work was supported by the SYNCHRONICS large scale initiative of INRIA, pp. 877–910. DOI: 10.1016/j.jcss.2011.08.009. URL: http://hal.inria.fr/hal-00766726.

[3] A. Benveniste, B. Caillaud and M. Malandain. 'The mathematical foundations of physical systems modeling languages'. In: *Annual Reviews in Control* 50 (2020), pp. 72–118. DOI: 10.1016/j.arcontrol.2020.08.001. URL: https://hal.inria.fr/hal-03045498.

[4] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger and K. G. Larsen. 'Contracts for System Design'. In: *Foundations and Trends in Electronic Design Automation* 12.2-3 (2018), pp. 124–400. DOI: 10.1561/1000000053. URL: https://hal.inria.fr/hal-01971429.

[5] J.-B. Jeannin, K. Ghorbal, Y. Kouskoulas, A. Schmidt, R. Gardner, S. Mitsch and A. Platzer. 'A Formally Verified Hybrid System for Safe Advisories in the Next-Generation Airborne Collision Avoidance System'. In: *International Journal on Software Tools for Technology Transfer* 19.6 (Nov. 2017), pp. 717–741. DOI: 10.1007/s10009-016-0434-1. URL: https://hal.archives-ouvertes.fr/hal-01232365.

[6] A. Sogokon, K. Ghorbal and T. T. Johnson. 'Operational Models for Piecewise-Smooth Systems'. In: *ACM Transactions on Embedded Computing Systems (TECS)* 16.5s (Oct. 2017), 185:1–185:19. DOI: 10.1145/3126506. URL: https://hal.inria.fr/hal-01658196.

11.2 Publications of the year

International journals

[7] A. Benveniste, B. Caillaud, M. Malandain and J. Thibault. 'Algorithms for the Structural Analysis of Multimode Modelica Models'. In: *Electronics* 11.17 (1st Sept. 2022), pp. 1–63. DOI: 10.3390/electronics11172755. URL: https://hal.inria.fr/hal-03768331.

[8] K. Ghorbal and A. Sogokon. 'Characterizing Positively Invariant Sets: Inductive and Topological Methods'. In: *Journal of Symbolic Computation* (1st Nov. 2022). URL: https://hal.archives-ouvertes.fr/hal-03540862.

Scientific book chapters

[9] A. Benveniste, B. Caillaud and M. Malandain. 'From Hybrid Automata to DAE-Based Modeling'. In: *Principles of Systems Design*. Vol. 13660. Lecture Notes in Computer Science. Springer Nature Switzerland, 29th Dec. 2022, pp. 3–20. DOI: 10.1007/978-3-031-22337-2_1. URL: https://hal.inria.fr/hal-03921708.

[10] I. Incer, A. Benveniste, A. Sangiovanni-Vincentelli and S. Seshia. 'Hypercontracts'. In: *NASA Formal Methods*. Vol. 13260. Lecture Notes in Computer Science. Springer International Publishing, 20th May 2022, pp. 674–692. DOI: 10.1007/978-3-031-06773-0_36. URL: https://hal.inria.fr/hal-03898326.

Reports & preprints

[11] A. Benveniste, B. Caillaud and M. Malandain. *Exact Structural Analysis of Multimode Modelica Models: Towards the Generation of Correct Simulation Code*. RR-9459. Inria Rennes - Bretagne Atlantique, 18th Feb. 2022, pp. 1–46. URL: https://hal.inria.fr/hal-03580636.

[12] A. Benveniste and J.-B. Raclet. *Mixed Nondeterministic-Probabilistic Automata: Blending graphical probabilistic models with nondeterminism*. RR-9447. Inria Rennes - Bretagne Atlantique, Jan. 2022, pp. 1–52. URL: https://hal.inria.fr/hal-03531059.

[13] K. Ghorbal and C. Kozaily. *On Covering Smooth Manifolds with a Q-arrangement of Simplicies: An inductive Characterization of Q-matrices*. 2022. URL: https://hal.inria.fr/hal-03897633.

[14] J. Thibault. *Constraint System Decomposition*. RR-9478. Inria Rennes, 29th July 2022, pp. 1–68. DOI: 10.13140/RG.2.2.13004.49285. URL: https://hal.inria.fr/hal-03740562.

11.3 Cited publications

[15] L. de Alfaro. 'Game Models for Open Systems'. In: *Verification: Theory and Practice*. Vol. 2772. Lecture Notes in Computer Science. Springer, 2003, pp. 269–289. DOI: 10.1007/978-3-540-39910-0_12.

[16] L. de Alfaro and T. A. Henzinger. 'Interface automata'. In: *Proc. of the 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'01)*. ACM Press, 2001, pp. 109–120. DOI: 10.1145/503271.503226.

[17] L. de Alfaro and T. A. Henzinger. 'Interface-based design'. In: *In Engineering Theories of Software Intensive Systems, proceedings of the Marktoberdorf Summer School*. Kluwer, 2004. DOI: 10.1007/1-4020-3532-2_3.

[18] L. de Alfaro, T. A. Henzinger and M. Stoelinga. 'Timed Interfaces'. In: *Proc. of the 2nd International Workshop on Embedded Software (EMSOFT'02)*. Vol. 2491. Lecture Notes in Computer Science. Springer, 2002, pp. 108–122. DOI: 10.1007/3-540-45828-X_9.

[19] A. Antonik, M. Huth, K. G. Larsen, U. Nyman and A. Wasowski. '20 Years of Modal and Mixed Specifications'. In: *Bulletin of European Association of Theoretical Computer Science* 1.94 (2008). URL: https://dblp.org/rec/journals/eatcs/AntonikHLNW08.bib.

[20] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, Cambridge, 2008. URL: https://mitpress.mit.edu/9780262026499/principles-of-model-checking/.

[21] A. Benveniste, T. Bourke, B. Caillaud, J.-L. Colaço, C. Pasteur and M. Pouzet. 'Building a Hybrid Systems Modeler on Synchronous Languages Principles'. In: *Proceedings of the IEEE*. Design Automation for Cyber-Physical Systems 106.9 (Sept. 2018), pp. 1568–1592. DOI: 10.1109/JPROC.2018.2858016. URL: https://hal.inria.fr/hal-01879026.

[22] A. Benveniste, T. Bourke, B. Caillaud, B. Pagano and M. Pouzet. *A Type-Based Analysis of Causality Loops In Hybrid Systems Modelers*. Deliverable D3.1_1 v 1.0 of the Sys2soft collaborative project "Physics Aware Software". Dec. 2013. URL: https://hal.inria.fr/hal-00938866.

[23] A. Benveniste, T. Bourke, B. Caillaud and M. Pouzet. *Semantics of multi-mode DAE systems*. Deliverable D.4.1.1 of the ITEA2 Modrio collaborative project. Aug. 2013. URL: https://hal.inria.fr/hal-00938891.

[24] A. Benveniste, B. Caillaud, H. Elmqvist, K. Ghorbal, M. Otter and M. Pouzet. 'Multi-Mode DAE Models - Challenges, Theory and Implementation'. In: *Computing and Software Science: State of the Art and Perspectives*. Vol. 10000. Lecture Notes in Computer Science. Springer, Oct. 2019, pp. 283–310. DOI: 10.1007/978-3-319-91908-9_16. URL: https://hal.inria.fr/hal-02333603.

[25] A. Benveniste, B. Caillaud, H. Elmqvist, K. Ghorbal, M. Otter and M. Pouzet. 'Structural Analysis of Multi-Mode DAE Systems'. In: *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, HSCC 2017*. Pittsburgh, PA, United States, Apr. 2017. DOI: 10.1145/3049797.3049806. URL: https://hal.inria.fr/hal-01521918.

[26] A. Benveniste, B. Caillaud and M. Malandain. 'Compile-Time Impulse Analysis in Modelica'. In: *MODELICA 2021 - 14th International Modelica Conference*. Linköping, Sweden, Sept. 2021, pp. 1–11. URL: https://hal.inria.fr/hal-03281394.

[27] A. Benveniste, B. Caillaud and M. Malandain. 'Handling Multimode Models and Mode Changes in Modelica'. In: *Modelica 2021 - 14th International Modelica Conference*. Linköping, Sweden, Sept. 2021, pp. 1–11. DOI: 10.3384/ecp21181507. URL: https://hal.inria.fr/hal-03281410.

[28] A. Benveniste, B. Caillaud and M. Malandain. *Structural Analysis of Multimode DAE Systems: summary of results*. Research Report RR-9387. Inria Rennes – Bretagne Atlantique, Jan. 2021, p. 27. URL: https://hal.inria.fr/hal-03104030.

[29] A. Benveniste, B. Caillaud and M. Malandain. 'The mathematical foundations of physical systems modeling languages'. In: *Annual Reviews in Control* 50 (2020), pp. 72–118. DOI: 10.1016/j.arcontrol.2020.08.001. URL: https://hal.inria.fr/hal-03045498.

[30] A. Benveniste, B. Caillaud, B. Pagano and M. Pouzet. 'A type-based analysis of causality loops in hybrid modelers'. In: *HSCC '14: International Conference on Hybrid Systems: Computation and Control*. Proceedings of the 17th international conference on Hybrid systems: computation and control (HSCC '14). Berlin, Germany: ACM Press, Apr. 2014, p. 13. DOI: 10.1145/2562059.2562125. URL: https://hal.inria.fr/hal-01093388.

[31] A. Benveniste, B. Caillaud, A. Ferrari, L. Mangeruca, R. Passerone and C. Sofronis. 'Multiple Viewpoint Contract-Based Specification and Design'. In: *Proceedings of the Software Technology Concertation on Formal Methods for Components and Objects (FMCO'07)*. Vol. 5382. Revised Lectures, Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Oct. 2008. DOI: 10.1007/978-3-540-92188-2_9.

[32] N. Bertrand, A. Legay, S. Pinchinat and J.-B. Raclet. 'A Compositional Approach on Modal Specifications for Timed Systems.' In: *11th International Conference on Formal Engineering Methods (ICFEM'09)*. Vol. 5885. LNCS. Rio de Janeiro, Brazil: Springer, Dec. 2009, pp. 679–697. URL: https://hal.inria.fr/inria-00424356.

[33] N. Bertrand, A. Legay, S. Pinchinat and J.-B. Raclet. 'Modal event-clock specifications for timed component-based design'. In: *Science of Computer Programming* 77 (2012), pp. 1212–1234. DOI: 10.1016/j.scico.2011.01.007. URL: https://hal.inria.fr/hal-00752449.

[34] N. Bertrand, S. Pinchinat and J.-B. Raclet. 'Refinement and Consistency of Timed Modal Specifications.' In: *3rd International Conference on Language and Automata Theory and Applications (LATA'09)*. Vol. 5457. LNCS. Tarragona, Spain: Springer, Apr. 2009, pp. 152–163. DOI: 10.1007/978-3-642-00982-2_13. URL: https://hal.inria.fr/inria-00424283.

[35] P. Bhaduri and I. Stierand. 'A proposal for real-time interfaces in SPEEDS'. In: *Design, Automation and Test in Europe (DATE'10)*. IEEE, 2010, pp. 441–446. DOI: 10.1109/DATE.2010.5457163.

[36] S. Bliudze. 'Un cadre formel pour l'étude des systèmes industriels complexes: un exemple basé sur l'infrastructure de l'UMTS'. PhD thesis. Ecole Polytechnique, 2006.

[37] S. Bliudze and D. Krob. 'Modelling of Complex Systems: Systems as Dataflow Machines'. In: *Fundamenta Informaticae* 91.2 (2009), pp. 251–274. DOI: 10.3233/FI-2009-0043. URL: https://hal.science/hal-02561099.

[38] G. Boudol and K. G. Larsen. 'Graphical versus logical specifications'. In: *Theoretical Computer Science* 106.1 (1992), pp. 3–20. DOI: https://doi.org/10.1016/0304-3975(92)90276-L. URL: https://www.sciencedirect.com/science/article/pii/030439759290276L.

[39] R. E. Bryant. 'Graph-Based Algorithms for Boolean Function Manipulation'. In: *IEEE Trans. Comput.* 35.8 (Aug. 1986), pp. 677–691. DOI: 10.1109/TC.1986.1676819. URL: http://dx.doi.org/10.1109/TC.1986.1676819.

[40] B. Caillaud, M. Malandain and J. Thibault. *Demo: IsamDAE, an Implicit Structural Analysis Tool for Multimode DAE Systems*. HSCC 2020 - 23rd ACM International Conference on Hybrid Systems: Computation and Control. Poster. Apr. 2020. URL: https://hal.inria.fr/hal-02545380.

[41] B. Caillaud, M. Malandain and A. Benveniste. 'A Reduced Index Mode-Independent Structure Model Transformation for Multimode Modelica Models'. In: *MODELICA 2021 - 14th International Modelica Conference*. Linköping, Sweden, Sept. 2021, pp. 1–11. URL: https://hal.inria.fr/hal-03320499.

[42] B. Caillaud, M. Malandain and J. Thibault. 'Implicit structural analysis of multimode DAE systems'. In: *HSCC 2020 - 23rd ACM International Conference on Hybrid Systems: Computation and Control*. Sydney New South Wales Australia, France: ACM, Apr. 2020, pp. 1–11. DOI: 10.1145/3365365.3382201. URL: https://hal.inria.fr/hal-02572879.

[43] B. Caillaud, B. Delahaye, K. G. Larsen, A. Legay, M. L. Pedersen and A. Wasowski. 'Compositional design methodology with constraint Markov chains'. In: *QEST 2010*. Williamsburg, Virginia, United States, Sept. 2010. DOI: 10.1109/QEST.2010.23. URL: http://hal.inria.fr/inria-00591578/en.

[44] B. Caillaud, B. Delahaye, K. G. Larsen, A. Legay, M. L. Pedersen and A. Wasowski. 'Constraint Markov Chains'. In: *Theoretical Computer Science* 412.34 (May 2011), pp. 4373–4404. DOI: 10.1016/j.tcs.2011.05.010. URL: http://hal.inria.fr/hal-00654003/en.

[45] S. L. Campbell and C. W. Gear. 'The index of general nonlinear DAEs'. In: *Numerische Mathematik* 72.2 (Dec. 1995), pp. 173–196. DOI: 10.1007/s002110050165. URL: http://dx.doi.org/10.1007/s002110050165.

[46] A. Chakrabarti. 'A Framework for Compositional Design and Analysis of Systems'. PhD thesis. EECS Department, University of California, Berkeley, Dec. 2007. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-174.html.

[47] A. Chakrabarti, L. de Alfaro, T. A. Henzinger and M. Stoelinga. 'Resource Interfaces'. In: *Embedded Software, Third International Conference, EMSOFT 2003, Philadelphia, PA, USA, October 13-15, 2003, Proceedings*. Vol. 2855. Lecture Notes in Computer Science. Springer, 2003, pp. 117–133. DOI: 10.1007/978-3-540-45212-6_9.

[48] E. Y. Chang, Z. Manna and A. Pnueli. 'Characterization of Temporal Property Classes'. In: *ICALP*. Vol. 623. Lecture Notes in Computer Science. Springer, 1992, pp. 474–486. DOI: 10.1007/3-540-55719-9_97.

[49] D. Cifuentes and P. A. Parrilo. 'Chordal Networks of Polynomial Ideals'. In: *SIAM J. Appl. Algebra Geom.* 1.1 (2017), pp. 73–110. DOI: 10.1137/16M106995X. URL: https://doi.org/10.1137/16M106995X.

[50] E. Clarke, O. Grumberg and D. Peled. *Model Checking*. MIT Press, 1999. URL: https://mitpress.mit.edu/9780262038836/model-checking/.

[51] G. E. Collins and H. Hong. 'Partial Cylindrical Algebraic Decomposition for Quantifier Elimination'. In: *J. Symb. Comput.* 12.3 (1991), pp. 299–328. DOI: 10.1016/S0747-7171(08)80152-6.

[52] N. J. Cutland, ed. *Nonstandard analysis and its applications*. Cambridge Univ. Press, 1988. DOI: 10.1017/CBO9781139172110.

[53] A. David, K. G. Larsen, A. Legay, U. Nyman and A. Wasowski. 'ECDAR: An Environment for Compositional Design and Analysis of Real Time Systems'. In: *Automated Technology for Verification and Analysis - 8th International Symposium, ATVA 2010, Singapore, September 21-24, 2010. Proceedings*. 2010, pp. 365–370. DOI: 10.1007/978-3-642-15643-4_29.

[54] A. David, K. G. Larsen, A. Legay, U. Nyman and A. Wasowski. 'Timed I/O automata: a complete specification theory for real-time systems'. In: *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2010, Stockholm, Sweden, April 12-15, 2010*. 2010, pp. 91–100. DOI: 10.1145/1755952.1755967.

[55] B. Delahaye, J.-P. Katoen, K. G. Larsen, A. Legay, M. L. Pedersen, F. Sher and A. Wasowski. 'Abstract Probabilistic Automata'. In: *Verification, Model Checking, and Abstract Interpretation - 12th International Conference, VMCAI 2011, Austin, TX, USA, January 23-25, 2011. Proceedings*. Vol. 6538. Lecture Notes in Computer Science. 2011, pp. 324–339. DOI: 10.1007/978-3-642-18275-4_23.

[56] F. Diener and G. Reeb. *Analyse non standard*. Hermann, 1989. URL: https://www.editions-hermann.fr/livre/analyse-non-standard-francine-diener.

[57] D. L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits*. ACM Distinguished Dissertations. MIT Press, 1989. DOI: 10.7551/mitpress/6874.001.0001.

[58] J. Edmonds and R. M. Karp. 'Theoretical improvements in algorithmic efficiency for network flow problems'. In: *Journal of the ACM* 19.2 (1972), pp. 248–264. DOI: 10.1145/321694.321699. URL: http://dx.doi.org/10.1145/321694.321699.

[59] H. Elmqvist, S. E. Mattsson and M. Otter. 'Modelica extensions for Multi-Mode DAE Systems'. In: *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Linköping University Electronic Press, Mar. 2014. DOI: 10.3384/ecp14096183.

[60] H. Elmqvist, F. Gaucher, S. E. Mattsson and F. Dupont. 'State Machines in Modelica'. In: *Proc. of the Int. Modelica Conference*. Ed. by M. Otter and D. Zimmer. Modelica Association. Munich, Germany, Sept. 2012, pp. 37–46. DOI: 10.3384/ecp1207637.

[61] H. Elmqvist, A. Neumayr and M. Otter. 'Modia-dynamic modeling and simulation with julia'. In: *Juliacon'18*. University College London, UK, Aug. 2018. URL: https://elib.dlr.de/124133/.

[62] H. J. Ferreau, S. Almér, H. Peyrl, J. L. Jerez and A. Domahidi. 'Survey of industrial applications of embedded model predictive control'. In: *2016 European Control Conference (ECC)*. 2016, pp. 601–601. DOI: 10.1109/ECC.2016.7810351.

[63] A. V. Goldberg and R. E. Tarjan. 'A new approach to the maximum flow problem'. In: *Proceedings of the eighteenth annual ACM symposium on Theory of computing (STOC'86)*. 1986. DOI: 10.1145/12130.12144. URL: http://dx.doi.org/10.1145/12130.12144.

[64] *IEEE Standard VHDL Analog and Mixed-Signal Extensions, Std 1076.1-1999*. 1999. DOI: 10.1109/IEEESTD.1999.90578. URL: http://dx.doi.org/10.1109/IEEESTD.1999.90578.

[65] Y. Iwasaki, A. Farquhar, V. Saraswat, D. Bobrow and V. Gupta. 'Modeling Time in Hybrid Systems: How Fast Is "Instantaneous"?' In: *IJCAI*. 1995, pp. 1773–1781. URL: https://www.ijcai.org/Proceedings/95-2/Papers/097.pdf.

[66] J.-B. Jeannin, K. Ghorbal, Y. Kouskoulas, R. Gardner, A. Schmidt, E. Zawadzki and A. Platzer. 'Formal verification of ACAS X, an industrial airborne collision avoidance system'. In: *2015 International Conference on Embedded Software, EMSOFT 2015, Amsterdam, Netherlands, October 4-9, 2015*. Ed. by A. Girault and N. Guan. Amsterdam, Netherlands: IEEE, 2015, pp. 127–136. DOI: 10.1109/EMSOFT.2015.7318268. URL: https://hal.science/hal-01660902.

[67] A. Lamercerie. 'Principe de transduction sémantique pour l'application de théories d'interfaces sur des documents de spécification'. Theses. Université Rennes 1 ; Rennes 1, Apr. 2021. URL: https://theses.hal.science/tel-03366457.

[68] L. Lamport. 'Proving the Correctness of Multiprocess Programs'. In: *IEEE Trans. Software Eng.* 3.2 (1977), pp. 125–143. DOI: 10.1109/TSE.1977.229904.

[69] K. G. Larsen, U. Nyman and A. Wasowski. 'On Modal Refinement and Consistency'. In: *Proc. of the 18th International Conference on Concurrency Theory (CONCUR'07)*. Springer, 2007, pp. 105–119. DOI: 10.1007/978-3-540-74407-8_8.

[70] K. G. Larsen and B. Thomsen. 'A Modal Process Logic'. In: *Proceedings of the Third Annual Symposium on Logic in Computer Science (LICS'88)*. IEEE, 1988, pp. 203–210. DOI: 10.1109/LICS.1988.5119.

[71] T. Lindstrøm. 'An Invitation to Nonstandard Analysis'. In: *Nonstandard Analysis and its Applications*. Ed. by N. J. Cutland. Cambridge Univ. Press, 1988, pp. 1–105. DOI: 10.1017/CBO9781139172110.002.

[72] J. Liu, N. Zhan and H. Zhao. 'Computing semi-algebraic invariants for polynomial dynamical systems'. In: *Proceedings of the 11th International Conference on Embedded Software, EMSOFT 2011, part of the Seventh Embedded Systems Week, ESWeek 2011, Taipei, Taiwan, October 9-14, 2011*. Ed. by S. Chakraborty, A. Jerraya, S. K. Baruah and S. Fischmeister. ACM, 2011, pp. 97–106. DOI: 10.1145/2038642.2038659. URL: https://doi.org/10.1145/2038642.2038659.

[73] H.-A. Loeliger. 'An introduction to factor graphs'. In: *IEEE Signal Processing Magazine* 21.1 (2004), pp. 28–41. DOI: 10.1109/MSP.2004.1267047.

[74] N. A. Lynch. 'Input/Output Automata: Basic, Timed, Hybrid, Probabilistic and Dynamic'. In: *CONCUR 2003 - Concurrency Theory, 14th International Conference, Marseille, France, September 3-5, 2003, Proceedings*. Vol. 2761. Lecture Notes in Computer Science. Springer, 2003, pp. 187–188. DOI: 10.1007/978-3-540-45187-7_12.

[75] N. A. Lynch and E. W. Stark. 'A Proof of the Kahn Principle for Input/Output Automata'. In: *Inf. Comput.* 82.1 (1989), pp. 81–92. DOI: 10.1016/0890-5401(89)90066-7.

[76] Z. Manna and A. Pnueli. *Temporal verification of reactive systems: Safety*. Springer, 1995. DOI: 10.1007/978-1-4612-4222-2.

[77] B. Meyer. 'Applying "Design by Contract"'. In: *Computer* 25.10 (Oct. 1992), pp. 40–51. DOI: 10.1109/2.161279. URL: http://dx.doi.org/10.1109/2.161279.

[78] P. Nuzzo, A. L. Sangiovanni-Vincentelli, X. Sun and A. Puggelli. 'Methodology for the Design of Analog Integrated Interfaces Using Contracts'. In: *IEEE Sensors Journal* 12.12 (Dec. 2012), pp. 3329–3345. DOI: 10.1109/JSEN.2012.2211098.

[79] C. Pantelides. 'The consistent initialization of differential-algebraic systems'. In: *SIAM J. Sci. Stat. Comput.* 9.2 (1988), pp. 213–231. DOI: 10.1137/0909014.

[80] J. D. Pryce. 'A Simple Structural Analysis Method for DAEs'. In: *BIT Numerical Mathematics* 41.2 (Mar. 2001), pp. 364–394. DOI: 10.1023/a:1021998624799. URL: http://dx.doi.org/10.1023/a:1021998624799.

[81] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay and R. Passerone. 'A Modal Interface Theory for Component-based Design'. In: *Fundamenta Informaticae* 108.1-2 (2011), pp. 119–149. DOI: 10.3233/FI-2011-416. URL: http://hal.inria.fr/inria-00554283/en.

[82] A. Robinson. *Non-Standard Analysis*. Princeton Landmarks in Mathematics, 1996. URL: https://press.princeton.edu/books/paperback/9780691044903/non-standard-analysis.

[83] E. Sikora, B. Tenbergen and K. Pohl. 'Industry needs and research directions in requirements engineering for embedded systems'. In: *Requirements Engineering* 17 (2012), pp. 57–78. DOI: 10.1007/s00766-011-0144-x. URL: http://link.springer.com/article/10.1007/s00766-011-0144-x.

[84] W. Y. Sit. 'The Ritt–Kolchin theory for differential polynomials'. In: *Differential Algebra and Related Topics*. 2002, pp. 1–70. DOI: 10.1142/9789812778437_0001.

[85] H. Tamaki. 'Positive-instance driven dynamic programming for treewidth'. In: *J. Comb. Optim.* 37.4 (2019), pp. 1283–1311. DOI: 10.1007/s10878-018-0353-z.

[86] J. Thibault and K. Ghorbal. 'Leveraging Structural Analysis for Quantified Boolean Formulae'. In: *Summer School on Modelling and Verification of Parallel Processes, Grenoble, France* 6 (2020). http://khalilghorbal.info/assets/pdf/papers/RBTF_movep.pdf.