Preprocessing Information for Nontrivial Goals / Advanced Compilation of Knowledge (PING/ACK)

Pré-traitement d'informations pour la résolution de tâches complexes / Compilation avancée de connaissances

---

Project summary

Designing algorithms ensuring fast response times is a fundamental problem in Computer Science. Its significance is all the more salient for algorithms requiring frequent interactions with humans. Indeed, one faces this issue quite often in everyday life (e.g., when using applications on the Web or on a smartphone, short response-time guarantees are mandatory). Furthermore, in many applications, much of the information given at the start or exchanged with the user take the form of or can be interpreted as pieces of "knowledge" which must be processed. Unfortunately, knowledge-based tasks are typically NP-hard, which implies that in the general case they suffer from intrinsic computational limitations disallowing response-time guarantees (unless P = NP).

Knowledge compilation is an approach for circumventing such limitations by pre-processing, during an off-line phase, some part of the available knowledge (the so-called "fixed" part F). Two main issues have been investigated for the past twenty years in this research area. On the one hand, developing a compilability model for deciding whether a task of interest is compilable or not (that is – loosely speaking – tractable provided that a polynomial-size compiled form of F has been computed first). On the other hand, drawing up knowledge compilation maps for selecting a representation language L into which F should be compiled. This is a multi-criteria decision, the choice of L depending on both its space efficiency (i.e., its relative ability to encode information using little space) as well as its time efficiency (i.e, the time complexity of the elementary subtasks which are needed to solve the problem under consideration). However, the existing model for compilability has the major drawback of classifying as non-compilable many tasks for which knowledge compilation appears as a valuable approach in practice. Such a gap between theory and practice comes from the facts that the compilability framework focuses on the worst case and is not fine-grained enough to take into account specific features of the inputs. Furthermore, the expressiveness of the languages L which have been used so far in knowledge compilation maps is still too limited or not adequate for some applications requiring more sophisticated constructs.

The objective of PING/ACK is to attack these shortcomings of existing work in knowledge compilation, extending its scope on both the theoretical side and the practical side. In a nutshell, we plan to define new knowledge compilation maps suited to more expressive representation languages than existing ones and to improve the applicability of knowledge compilation; we aim to refine the concept of compilability, drawing up more fine-grained maps (that would not be focused on worst-case scenarios), and investigating other approaches for enhancing the benefits that knowledge compilation offers. This will produce characterization results of various types (expressiveness, succinctness, complexity), but also involve the definition of new concepts and languages, and the conception of new algorithms (compilers and reasoners) as well as their implementation and experimental evaluation on problems from several areas, both inside and outside the standard AI scope, including recommendation, configuration, robotics, and in a more risky perspective, bioinformatics.

Contents

1 Proposal context, positioning and objective(s) **2**
1.1 Objectives and scientific hypotheses . . . 2
1.2 Originality and relevance in relation to the state-of-the-art . . . 4
1.3 Risk management and methodology . . . 4

2 Project organization and means implemented **5**
2.1 Consortium . . . 5
2.2 Scientific coordinator . . . 6
2.3 Scientific program . . . 6
2.3.1 WP1: Drawing compilation maps suited for complex representations . . . 6
2.3.2 WP2: Parameterized complexity for KC maps . . . 8
2.3.3 WP3: Fine-grained KC maps . . . 10
2.3.4 WP4: Beyond standard benchmarks and applications for KC . . . 13

2.3.5 WP5: Coordination of the project and dissemination of the results 14
2.4 Partners' contributions, schedule of the project and deliverables 15
2.5 Justification of the budget. 16

3 Impact and benefits of the project 16

Consortium

The success of PING/ACK will be based on a consortium bringing together 19 researchers, 16 of them coming from four academic labs, CRIL, GREYC, IRIT and LaBRI, plus 3 researchers who will be hired by the project. The consortium covers a wide range of scientific and technological expertise about knowledge representation, constraint programming, theory of algorithms and computational complexity, but also has a strong experience with the solving of industrial problems. PING/ACK will involve a total manpower of 291.6 person.month. The requested budget for PING/ACK is 432 081 €, including the funding of two co-supervised Ph.D. students for three years and a post-doc for one year, and the management costs (8 % added).

Summary table of persons involved in the project

Partner	Name	First name	Current position	Involvement (person.month)	Role and responsibilities in the project
CRIL	Koriche	Frédéric	PR	12	WPs 1, 2, 3
CRIL	Lagniez	Jean-Marie	MCF	19.2	WPs 1, 3, 4
CRIL	Marquis	Pierre	PR	24	Scientific coordinator, WPs 1, 2, 3, 4, 5 (leader)
CRIL	Mengel	Stefan	CR	14.4	WPs 1, 2 (leader), 3, 5
CRIL	Wallon	Roman	Ph.D. student	6	WP 2
GREYC	Clément	Julien	CR-HDR	12	WPs 2, 3
GREYC	Grandjean	Étienne	PR	12	WPs 1, 2, 3
GREYC	Niveau	Alexandre	MCF	19.2	WPs 1, 2, 3 (leader), 4, 5
GREYC	Zanuttini	Bruno	MCF-HDR	12	WPs 1, 3
IRIT	Cooper	Martin	PR	12	WPs 1, 2
IRIT	Fargier	Hélène	DR	14.4	WPs 1 (leader), 3, 4, 5
IRIT	Maris	Frédéric	MCF	14.4	WPs 1, 4
IRIT	Mengin	Jérôme	MCF	12	WPs 1, 4
LaBRI	Gimbert	Hugo	CR	7.2	WPs 1, 4
LaBRI	Ly	Olivier	MCF	7.2	WPs 1, 4
LaBRI	Simon	Laurent	PR	9.6	WPs 1, 4 (leader), 5
CRIL/GREYC	X	X	Ph.D. student	36	WPs 2, 3
GREYC/IRIT	Y	Y	Ph.D. student	36	WPs 1, 4
IRIT/LaBRI	Z	Z	Post-doc	12	WP 4

Any changes that have been made in the full proposal compared to the pre-proposal:
As to the financial aspects, the budget has been slightly updated because the costs induced by the recruitment of a post-doc are a bit higher than expected initially (51 935 € instead of 47 760 €). Clearly, the modifications lead to a budget increase which remains far below the limit of 15% allowed by ANR.

---

1 Proposal context, positioning and objective(s)

1.1 Objectives and scientific hypotheses

Designing algorithms that ensure fast response times is a fundamental task in Computer Science. Unfortunately, as problems become more challenging, they also become *computationally harder* (typically, at least NP-hard), preventing any direct approach from succeeding in a general setting.

Let us consider two application areas that will be considered in the project: Recommender systems and Robots. Recommender systems have been successfully applied for guiding users in the enormous sets of products and services accessible on the Web. However, as soon as recommendation is not limited to simple products (like books or movies), current approaches become intractable or inaccurate. This is the case for instance for travel offers and configurable products for which the number of possibilities is beyond the reach of standard collaborative filtering techniques. The second example concerns autonomous robots. In many cases, such robots need to make complex decisions as quickly as possible, despite limited capacities of reasoning and memory.

Knowledge-based approaches have recently emerged as an alternative for the next generation of recommenders. Based on expressive representation languages, they are flexible enough to describe products and users in a compact and intelligible way (they include, for example, constraint-based recommenders and Bayesian networks). However, processing the encoded knowledge to infer recommendations is NP-hard, which is a major obstacle for Web applications. Users expect relevant recommendations but also fluent interaction in terms of response times. As an example, consider a car configuration scenario: each time the user makes an elementary choice (e.g., the engine she wants), it is important to let her know in less than a second some possible values (and more than that, some recommended values) of the other components of the car, which are compatible with the choices made so far (for instance, the fact that they enforce a specific gear-box, or that the customers having made the same choice for the engine have usually considered this or that gear-box). Unfortunately, all these tasks are NP-hard, and thus direct approaches for them do not offer any response time guarantee.

Knowledge compilation (KC) (see Cadoli and Donini [1998] and Marquis [2015] for surveys) is a valuable approach to overcoming these limitations. Roughly, KC consists in preprocessing some pieces of information (available off-line) in order to improve from a computational point of view one or several tasks to be achieved. One can compile pieces of information of many different natures, such as a catalog of configurable products, the model of a system to be diagnosed, the transition function of a Markov decision process, etc. Other pieces of information (e.g., the desiderata of the customer; the symptoms of the system to be diagnosed; the planning goals) are necessary to achieve the tasks under consideration (e.g., recommend a gear-box; compute a most probable failure; define a plan) but are only available on-line. KC is then useful when the compilation phase offers some guaranteed response times for the on-line phase, or when it leads to a reduction of the overall computation time.

In more formal terms, suppose that the input information can be split into two parts, the "fixed part" F and the "varying part" V, and that the task to be achieved is modeled in an abstract way as a mapping f, the goal thus being to compute $f(F,V)$. Such a partition of information is natural in many applications: most of the information exploited by the application is user-independent (e.g., the product catalog) and can be considered as the "fixed" part F, while the amount of information V expected from the user is much smaller. Pre-processing F means using a *compilation function* C, during an off-line phase, to turn F into a compiled form $C(F)$. Then, given the variable part V in an on-line phase, one computes $f(F,V)$ in polynomial time using the compiled form F.

An important issue in KC is to determine whether the task f of interest is *compilable* in some way. When f is a Boolean function, f is considered as compilable (formally, as a member of the class compP) if there exists a compilation function C such that, on the one hand, for all F the size of $C(F)$ is polynomial in the size of F, and on the other hand the computation of $f(F,V)$ can be achieved in time polynomial if $C(F)$ is given as an additional input. Another important issue in KC is the choice of the target language L into which the "fixed" part F must be compiled, which is reminiscent of the choice of a data structure for a given abstract data type, a well-known issue in Computer Science. This choice relies on the expressiveness and the spatial efficiency of the potential target languages [Gogic et al., 1995; Cadoli et al., 2000], as well as their time efficiency w.r.t. the tasks to be achieved. For instance, for a product configuration application, one needs an operation for incorporating the user's choices into the L representation of the feasible products ("conditioning"), and a query for determining whether the resulting set of products is empty or not ("consistency"). A *KC map* is a systematic, multi-criteria comparative analysis of representation languages. The choice of a target language, based on a KC map, is thus guided by its spatial efficiency and its time efficiency.

The goal of PING/ACK is to address new challenges in KC to extend its applicability to new domains, integrating both a theoretical and a practical perspective. In a nutshell, we plan to define new KC maps suited to representation languages which are more expressive than the ones considered in existing maps (those languages are merely propositional ones) and to improve the applicability of KC; we aim to refine the concept of compilability, drawing up more fine-grained maps (that would not be focused on worst-case scenarios), and investigating other approaches for enhancing the benefits that KC offers. This will produce characterization results of various types (expressiveness, succinctness, complexity), but also involve the definition of new concepts and languages, and the conception of new algorithms. Special attention will be payed to the design, implementation, and experimental evaluation of new compilers and reasoners, i.e., programs that compile the fixed part of an input and programs that, given a compiled form $C(F)$ and an input V, solve the reasoning task at hand. We want also to go beyond standard application areas for KC, by investigating the use of KC techniques for robotics and bioinformatics. The expected outputs of the project will take the form of publications in top-ranked journals and conferences, as well as software and benchmarks (including the design, the implementation and the evaluation of the compilers and reasoners).

1.2 Originality and relevance in relation to the state-of-the-art

A map for propositional formulae (Boolean functions) has been drawn up [Darwiche and Marquis, 2002] and since then enriched with new languages, new queries and new transformations. Many languages for representing propositional information have been introduced and/or located on the map. Among them are the influential language of OBDDs (Ordered Binary Decision Diagrams) [Bryant, 1986] and that of DNNFs (Decomposable Negation Normal Form representations) [Darwiche, 2001]. Such languages have been exploited in several applications, such as car configuration. For example, the car configurator developed by Renault[1] is based on cluster tree compilation, while that of Toyota takes advantage of DNNF representations.[2] However, the expressiveness achieved by those frameworks (focused on propositional information) is still too restricted or not adequate in some applications. Similarly, the set of queries and transformations considered in the existing maps is not suited for possible settings outside of product configuration. For example, as we will detail in the definition of the work packages, no compilation map exists that can be used in planning; in this domain, we need to efficiently handle epistemic states – i.e., models of what is known about the actual state of the world – and temporal information, that cannot be well represented by simple Boolean functions. We will also see that compilation maps are not well-suited for recommendation, where the main task is to reason on preferences.

Beyond compilation maps, a "compilability framework" was developed two decades ago (see among others Liberatore [2001]; Cadoli et al. [2002]). Several compilability classes have been introduced and the (non)-compilability of a number of problems has been determined (see e.g., Liberatore [1998]; Liberatore and Schaerf [2007]). Unfortunately, this pioneering work has the major drawback of classifying as non-compilable (formally, not in the class compP) many problems for which KC is in fact a valuable approach in practice, e.g., product configuration or model-based planning. Such a gap between theory and practice comes from the fact that the compilability framework focuses on the worst case and is not fine-grained enough to take into account specific features of the inputs.

The results that will be produced by the PING/ACK project aim to circumvent the limitations listed above. This is crucial for tackling the new, yet challenging applications we plan to consider.

1.3 Risk management and methodology

Organization of the project. PING/ACK will be organized into 4 work packages (WP1 to WP4), plus an "administrative" work package (WP5), and will last 48 months:

- **WP 1: Drawing compilation maps suited to complex representations** [t_0, t_0+42]. This work package aims to extend KC maps to languages that are not Boolean in essence, but are concerned for instance with rankings, modal atoms or propositional atoms subject to constraints which cannot be encoded in the language itself. It will explore, through the establishment of KC maps on three particular domains (preferences handling, temporal reasoning and epistemic reasoning), the problems raised by domains of knowledge that are more complex than propositional logics. These maps will be completed by experimental maps that compare the practical succinctness and the on-line efficiency of the languages on existing benchmarks in the standard scope of AI.
- **WP 2: Parameterized complexity for KC maps** [t_0, t_0+24]. In parameterized complexity theory [Downey and Fellows, 2013], the efficiency of an algorithm is evaluated by considering an additional parameter k. This setting allows for Fixed-Parameter Tractable (FPT)-size pre-processing and FPT-time query answering, making it possible to extend the compilability classes to FPT-size compilability classes [de Haan, 2015]. We plan to take advantage of this richer setting to classify in a more precise way the compilability of many requests from existing maps (and from the new maps produced by WP1).
- **WP 3: Fine-grained KC maps** [t_0, t_0+36]. In order to make a better choice of a compilation language, this work package will investigate the generation of more fine-grained maps, where one goes beyond the identification of P vs NP-hard requests and worst-case complexity analyses, by using average-case and distribution analysis, low complexity classes, and approximation schemes both for the off-line and on-line phases.
- **WP 4: Beyond standard benchmarks and applications for KC** [t_0+6, t_0+48]. We expect this WP also to cross-fertilize methods and needs between KC and other fields. The use of KC for bioinformatics (specifically for compiling the complex preference relations one wants to reason about) will be considered. We will also take advantage of the strong expertise of some participants in autonomous robots for the conception of challenging problems and benchmarks. In particular, we plan to investigate how to help Rhoban, the winner of the last two Robocup Humanoid competitions

---

[1] See e.g., ⟨https://www.renault.fr/vehicules/vehicules-particuliers/twingo.html⟩.

[2] See ⟨http://newsroom.ucla.edu/releases/artificial-intelligence-framework-developed-by-ucla-professor-now-powers-toyota-websites⟩.

(“kid size” category; see ⟨http://rhoban.com⟩), to make complex decisions on the soccer field with response time guarantees. The computing clusters at CRIL will be exploited for performing large-scale experiments.

- **WP 5: Coordination of the project and dissemination of the results** [t_0, t_0+48]. This work package follows and coordinates the previous “scientific” work packages. It will be dedicated also to the dissemination of the results.

Risk management. The objective of PING/ACK, namely extending the scope of the KC approach to problem solving, from the theoretical side to the practical side, has not been subject to systematic study so far. Thus, there are many interesting open problems which we are planning to attack. Due to the large number of questions, we consider it very likely that we will successfully make progress and open the field for further research.

WPs 1, 2, 3 are complementary, and as such, they may progress mostly simultaneously (this independence limits the risk of the project not being successful at all). WP 4 will accompany the project throughout all stages for practical validation. In particular, in order to ensure that the project will lead to practical developments, the last year will be exclusively dedicated to WP 4 and to the implementation and empirical evaluation parts of WP 1.

2 Project organization and means implemented

2.1 Consortium

PING/ACK will be based on a consortium bringing together 16 researchers (group A) from four academic labs, CRIL, GREYC, IRIT and LaBRI, plus 3 additional researchers hired by the project (group $\bar{A}$). Each WP will involve at least three of the four labs (see Section 1.3). Group A consists of the following 15 permanent researchers, plus one first-year Ph.D. student:

- CRIL (UMR 8188, Artois University): Pierre Marquis (CRIL leader, PR); Frédéric Koriche (PR); Jean-Marie Lagniez (MCF); Stefan Mengel (CR); Romain Wallon (Ph.D.).
- GREYC (UMR 6072, University of Caen Normandie): Julien Clément (CR-HDR), Etienne Grandjean (PR), Alexandre Niveau (GREYC leader, MCF), Bruno Zanuttini (MCF-HDR).
- IRIT (UMR 5505, University Paul Sabatier – Toulouse): Martin Cooper (PR), Hélène Fargier, (IRIT leader, DR), Frédéric Maris (MCF), Jérôme Mengin (MCF).
- LaBRI (UMR 5800, University of Bordeaux – Bordeaux INP): Hugo Gimbert (CR), Olivier Ly (MCF-HDR), Laurent Simon (LaBRI leader, PR).

This consortium covers a wide range of scientific and technological expertise in knowledge representation, constraint programming, theory of algorithms and computational complexity, but also has a strong experience in solving industrial problems. Thus, synergetic effects will be obtained by the combination of diverse experience and the various abilities of the participants. For instance, IRIT and LaBRI will provide instances corresponding to real industrial problems they study. The past collaborations of participants from the different teams have resulted in a number of common publications in top-ranked journals and conferences (they can be found in the Web pages of the participants – for a sample, see also the résumés of the WP leaders which have been uploaded). Some pairwise collaborations are long-term. Consequently, there is no doubt that the teams involved in PING/ACK will be able to work together in a fruitful way. In addition, the project will also offer the participants the opportunity to start new collaborations, especially on practical aspects (but not only), through the co-supervision of the participants from group $\bar{A}$. Those collaborations would not exist otherwise.

Group $\bar{A}$ consists of three additional participants to be recruited: two co-supervised Ph. D. students for three years (named X and Y), and 12 months of a co-supervised post-doc (named Z).

- A first Ph.D. student X will be co-supervised by CRIL and GREYC and will start in the first year of the project. The research of this student will mainly be in WP 2 and 3. In particular, parameterized compilability of important problems in artificial intelligence will be considered as well as a refined version of the KC map where the parameterized complexity of transformations and queries will be studied.
- A second Ph.D. student Y hired by the project will be co-supervised by GREYC and IRIT and will start in the first year of the project. The research of this student will mainly be in WPs 1 and 4. The objective of the thesis will be the elaboration of a KC map suited for the representation of temporal problems, through the development of methodologies devoted to (i) the comparison of heterogeneous languages and (ii) the handling of external constraints.

- A post-doc Z for one year will be co-supervised by IRIT and LaBRI; she/he will start at the end of the third year of the project. The research of this post-doc will be in WP 4 (in connection with WP 1). This post-doc will be involved in the design, the implementation and the evaluation of compilers and reasoners, based on the results arising from the three other scientific work packages.

2.2 Scientific coordinator

Pierre Marquis is a professor at Artois University since 1998 and a member of the *Institut Universitaire de France* since 2017. He is a Fellow of the European Association for AI since 2009. KC has been one of the main research topics of Pierre Marquis for the past twenty years. His work on KC is recognized at the international level; it notably gave rise to more than 30 publications in the main AI journals and conferences, such as AIJ, JAIR, IJCAI, AAAI, ECAI and KR (see ⟨http://www.cril.fr/~marquis/⟩). Pierre Marquis gave a tutorial on KC at ECAI'08, and an introductory course on KC during the European School on AI ACAI'15 (see ⟨http://www.cril.fr/acai15/⟩). He collaborated on this topic with many internationally renowned researchers in France and abroad, including Lucas Bordeaux (Microsoft Research, Cambridge), Adnan Darwiche (UCLA, Los Angeles) and João Marques-Silva (INESC-ID, Lisboa). With Stefan Szeider (TU Wien), he co-organized in June 2015 the first symposium on "New Frontiers in Knowledge Compilation", supported by the Vienna Center for Logic and Algorithms and by the Wolfgang Pauli Institute (see ⟨http://www.vcla.at/kc2015/⟩). With Adnan Darwiche, Dan Suciu (U. Washington) and Stefan Szeider, he co-organized a Dagstuhl seminar centered on KC in September 2017 (see ⟨http://www.dagstuhl.de/17381⟩ and [Darwiche et al., 2017]). He is also (for the French part) the proponent of a bilateral BARRANDE France/Czech Republic project (2017-18), centered on KC for constraint programming. With Adnan Darwiche, he has been invited to give a tutorial on "Recent Advances in Knowledge Compilation" at the next IJCAI-ECAI'18 conference. His strong international network will be useful for leveraging the results of the project.

2.3 Scientific program

2.3.1 WP1: Drawing compilation maps suited for complex representations

As previously said, one of the cornerstones of KC is the building of *KC maps*. Much effort has been done in the past ten years to explore languages based on propositional atoms (propositional variables or variable/value pairs). Important (and nontrivial) theoretical results have been obtained, that provide a toolbox for designing algorithms with solid guarantees and for the development of applications within precise guidelines.

In order to cover a broader range of problems and applications, it is necessary to push these limits forward and to draw new KC maps dedicated to languages suited to the representation of more sophisticated data types, namely, ones that are not based solely on propositional atoms. For instance, we plan to consider languages built up for objects such as rankings, modal atoms, or propositional atoms subject to constraints which cannot be (for expressiveness or spatial efficiency reasons) encoded in the language itself. We will consider three particular families of data types, namely those involving preferences (used e.g. in recommender systems), and those encapsulating, respectively, temporal and epistemic information (both typically used in planners). The work package will not be limited to these domains but we believe that focusing on these flagship domains will be useful in general by making the problems to be solved clearer and the difficulties to be overcome more salient. The second advantage of the focus is that it will make the empirical evaluation easier and may lead to a quicker transfer to practical applications.

Preference models are (more or less) compact representations of some user preferences over a large set of possible outcomes from a combinatorial domain. Like probabilistic graphical models, preference models can be classified into two main categories, namely directed preference models, including Conditional Preference networks (CP-nets, [Boutilier et al., 2004]) and their variants [Brafman and Domshlak, 2002; Boutilier et al., 2001], and undirected preference models, including Generalized Additive Independence networks (GAI-nets, [Bacchus and Grove, 1995]), which share strong similarities with weighted constraint networks (see e.g. [Meseguer et al., 2006]). Undirected preference models also include well-studied families of ranking models used in Economics, such as the Luce model, the Mallows model, and the Bradley-Terry model [Fligner and Verducci, 1993].

The main queries of interest for preference models are the *ranking queries* (e.g., find the first k alternatives, rank the values of a variable) and the optimization queries (e.g., find an alternative that maximizes the accordance with the user's preferences). These queries are intractable in many cases. For example, the optimization task in weighted constraint networks is NP-hard, and the ranking task in Mallows models (and their Luce or Bradley-Terry variants) is #P-hard [Lu and Boutilier, 2014]. Getting a detailed KC map of preference models is useful per se and will benefit many AI applications in decision making. This setting also raises three particularly exciting and difficult questions:

- *The problem of the heterogeneity of the domains* – unlike in the classical, Boolean case, we have to compare different, non-isomorphic, ways to represent information (and not only different ways of representing the same data): for example, user preferences can be represented by (partial) rankings or by utility functions.
- *The existence of an implicit "theory"* (in the sense of SMT solvers): for instance, when dealing with a preference relation modeled as a preorder, the three pieces of information $a < b$, $b < c$ and $a < c$ cannot be considered as independent (as distinct propositional atoms would be) since the two former imply the latter.
- The use of the knowledge represented is not limited to the requests imagined in the seminal KC maps – they are not even limited to simple queries and transformations: in configuration, the aim is not only to point out an optimal option, but to learn from the past choices of the user(s); more generally, the criteria used to compare languages for, e.g., preference-based recommender systems, are strongly dependent on tasks of passive and active learning. *The performance of a language also depends on the feasibility of the learning task.*

Temporal and planning problems share some of these difficult aspects, and open other questions as well. As representation languages, even the simplest ones, like Allen's algebra [Allen, 1983], the point algebra [Vilain and Kautz, 1986] or temporal constraint satisfaction problems [Dechter et al., 1991] (all of them being propositional) do not share a common semantics – hence, again, the necessity of drawing heterogeneous compilation maps. The queries considered in planning problems are also not as pure as what the classical map assumes (again, one must take into account, during the planning phase, the fact that one will learn from the environment during the on-line execution phase – this is typically the problem raised by exploration tasks and handled by approaches like reinforcement learning). Moreover, like preference frameworks, even the simplest temporal frameworks involve an exogenous theory (typically, the transitivity of the sequencing of instants or the composition of difference constraints).

Planning problems also call for the investigation of *sets* of representations considered in the existing maps (sets of sets of states, sets of probability distributions [Liberatore, 2002]). Since planning asks an agent to decide on a course of actions to take before actually acting, when the environment is not fully observable or the actions are nondeterministic, the agent must consider different possible executions. It must hence reason about sets of knowledge states, sets of belief states, or even probability distributions over belief states. Then, when simulating actions and observations in the course of planning, the planner must again transform and query these representations.

In summary, WP 1 will consider new KC maps dedicated to languages suited to the representation of data types which are not based solely on propositional atoms. Through the establishment of KC maps on three particular subjects, taken from various subfields of AI (preference modeling, temporal reasoning and automated planning), this WP will explore the problems raised by domains of knowledge that are more complex than propositional logic, in particular along three mostly unexplored axes: compilation modulo a theory, heterogeneous KC maps and learning maps.

Task 1.1: On the compilation of preference representation languages. The compilation of preference relations, utility functions and probability distributions by decision diagrams has been touched upon in the literature; see e.g. Darwiche [2003]; Darwiche and Marquis [2004]; Coste-Marquis et al. [2004]; Chavira and Darwiche [2008]; Kisa et al. [2014]; Fargier et al. [2014]; Bart et al. [2016]. In most cases, each of these works provides a complexity analysis of a sole language, or looks at the problems from the algorithmic point of view. In this task, our first work is to draw maps in order to compare languages with respect to their relative expressiveness, their relative succinctness, and their ability to (efficiently) answer requests and transformations. This study clearly comes out of the problem of the comparison of heterogeneous languages; it calls for the extension and the validation of the work of Fargier et al. [2013a] – given our strong background on this topic, results should be obtained quickly.

The second important question raised by this task is the complexity of learning. We will explore which tools and results from computational learning theory can be used in a KC map, typically the notions of Vapnik-Chervonenkis dimension [Blumer et al., 1989] and of sample complexity – applied to preference languages. The first objective of this "learning map" is to systematically study, compare and probably extend the results provided by the literature about the VC dimension of languages for representing preferences [Booth et al., 2010; Chevaleyre et al., 2011; Bigot et al., 2012; Friedman and Yakhini, 1996]. We will also aim for general results relating the expressiveness and the compilability of the languages to the complexity of learning – are less expressive models easier to learn? – and investigate the interplay between the expressiveness of the language chosen for the learning phase, which has a strong influence on the generalization properties of this language and on its sample complexity, and the compilability of the learnt knowledge. Finally, and this will be especially important in settings where the learning process continues on-line, we will investigate when it is possible to directly learn the compiled form – which is linked to the issue of incremental compilation.

Task 1.2: On the compilation of temporal problems. As in the previous task, the languages to compare are heterogeneous, and a part of the study will be done conjointly. The domain is however less explored than the previous one, even if we include into the corpus of papers all those looking for polynomial subclasses of temporal CSPs, scheduling problems, or Allen's algebra. It is known that the classical notion of decision diagrams already pioneered by Bryant [1986] hardly applies (see e.g. the works of Møller et al. [1999] and Fargier et al. [2015] about difference decision diagrams, and the work of Niveau et al. [2010] about interval automata), so the field of investigation is still to be explored – e.g., not much is known about the applicability of important notions from KC such as decomposability, AND/OR trees, closures of languages.

This task is also very promising because temporal frameworks implicitly rely on exogenous knowledge – typically, on the transitivity of many temporal relations. What is called for here is a formalization of the idea of "Compilation Modulo a Theory". To the best of our knowledge, nothing has been written yet in the KC literature, but this task can inherit from the domain from which we borrowed its name, namely SMT ("Satisfiability Modulo Theory") [Barrett et al., 2009], which does consider such problems (typically, for constraint satisfaction problems). As a matter of fact, compilers modulo a theory can be envisioned as SMT solvers with a trace, by analogy with top-down compilers that obey a "DPLL with a trace" principle [Huang and Darwiche, 2005].

Task 1.3: On the compilation of belief states. In contingent planning, where plans are conditional on the sequence of actions taken and observations received, the agent *executing* a plan must maintain a *belief state*, i.e., the set of states which it considers likely to be the actual one; each state is in turn a Boolean assignment to the relevant set of propositional variables. Consequently, when *searching for*, or *verifying* a plan, a planner must consider all possible contingencies and hence work with *sets* of belief states. This calls for languages able to represent such states in a compact form and supporting a certain number of queries, like those allowing to compute the belief states resulting from an observation. In other words, contingent planning calls for efficient languages for sets of sets of propositional assignments or, equivalently, for epistemic (S5, mono-agent) formulae.

Surprisingly, there are very few studies about the compilation of epistemic formulae [Bienvenu et al., 2010; Niveau and Zanuttini, 2016]. Still, some very efficient planners use indirect, or very ad hoc ways of representing belief states [Hoffmann and Brafman, 2005; Albore et al., 2010]. We believe that studying existing languages in a systematic way and providing new languages for representing sets of belief states would open up new directions in this very active research field. Moreover, evaluating the impact of such languages will be made easier by the possibility to plug them directly into existing planning algorithms and evaluating them on benchmarks from this domain (such as those of the International Planning Competition[3]).

Each of the three tasks will lead to an (extended) theoretical KC map, to the implementation of compilers and to the establishing of *experimental KC maps* that compare the practical succinctness and the on-line efficiency of the languages on (i) existing benchmarks in the standard scope of AI, namely benchmarks from planning repositories and benchmarks from configuration problems coming from the automotive industry (conceived by IRIT with the courtesy of Renault). Because the three tasks share common questions, all the members of WP 1 will be involved in Task 1.1, 1.2 and 1.3 and meet at least twice a year. The collaboration will be made stronger by the co-supervision of the GREYC-IRIT Ph.D. student Y, whose thesis will deal with the compilation of temporal problems (Task 1.2).

Participants. Martin Cooper, Hélène Fargier (coordinator), Hugo Gimbert, Étienne Grandjean, Frédéric Koriche, Jean-Marie Lagniez, Olivier Ly, Pierre Marquis, Frédéric Maris, Stefan Mengel, Jérôme Mengin, Alexandre Niveau, Laurent Simon, Bruno Zanuttini, Ph.D. student Y – for a total manpower of 99.8 person.month.

Schedule and deliverables. Task 1.1 will start at t_0, followed (at t_0+6) by Tasks 1.2 and 1.3. All will end at t_0+42. A report will be written at the end of each year: t_0+12 (KC maps of preference languages), t_0+24 (KC maps of temporal and epistemic languages), t_0+36 (compilation modulo a theory), t_0+42 (experimental KC maps).We will also write a final report at t_0+48 to summarize our findings in this work package.

2.3.2 WP2: Parameterized complexity for KC maps

As discussed before, a compilation problem is called *compilable* if, given the off-line part F, one can compute a polynomial size compiled form $C(F)$ such that the on-line queries can be answered in polynomial time with the help of $C(F)$. Unfortunately, it turns out that the restriction to polynomial size is often too restrictive so that – despite the successes in practice – many important problems are hard in compilability theory as formulated by Cadoli et al. [2002]. In this work package, we will aim to improve this situation by relaxing the size requirement for $C(F)$ in a direction suggested by parameterized complexity theory, see e.g., Downey and Fellows [2013]. To this end, we will allow a dependence on a *parameter* of the instance that is superpolynomial, where the parameter is problem-dependent, e.g., the size of a solution or some structural property of the input such as the treewidth of some associated graph. A pair of a computational problem and a parameterization, i.e., a mapping that assigns a parameter value to every instance, is called a parameterized problem.

---

[3] See ⟨http://www.icaps-conference.org/index.php/Main/Competitions⟩.

Given an instance together with a parameter k, for many natural problems compilation into a form $C(F)$ of size roughly $|F|^k$ is relatively easy to achieve. Unfortunately, even for moderate values of k and sizes of F, the resulting compiled form is infeasibly large. Therefore, in parameterized compilation, one aims to instead achieve compilation sizes of the form $f(k)|F|^c$ for a function f and a constant c. Note that in this size bound the exponent of $|F|$ is independent of the parameter and thus, if f is reasonable and k and c are not too large, compilation may be feasible. Parameterized problems that allow compilation of the off-line part into a compiled form of size $f(k)|F|^c$ are called *fixed-parameter compilable* and they are considered the class of tractable problems in parameterized compilability. This is motivated by the fact that, also in practice, if a problem is fixed-parameter compilable and the parameter values on interesting instances are relatively small, then treating this problem may be feasible despite the negative results of Cadoli et al. [2002]. The aim of this work package is to systematically analyze which problems that are generally hard have useful parameterizations that make them fixed-parameter compilable.

The problems we will consider will come mostly from two sources: on the one hand we will consider compilation problems that were identified as hard in the polynomial size setting in WP1. On the other other hand we will revisit problems from the KC literature for which hardness in the traditional compilability framework has already been established.

The work package will be decomposed into two individual but interacting tasks which build on one common question.

Common question: the parameter choice. One central challenge in both tasks will be choosing the right parameters: on the one hand, we want them to be restrictive enough to allow compilation into representations of interesting size. On the other hand, they need to be general enough to capture many instances in practice. Thus choosing good parameters will require the collaboration of more theoretically minded and more practical members of the consortium. In particular, analyzing potentially interesting parameters will require both a theoretical analysis of the compilation properties and an analysis of the distribution of the parameter on practical instances, in collaboration with WP1. Thus this work package will strongly benefit from the breadth of the consortium of this project.

Task 2.1: Fixed-parameter compilability. In this task, we will consider the parameterized compilability of problems in a rather abstract setting similar to Cadoli et al. [2002]. The focus will be on designing algorithms that allow parameterized compilation as discussed above. Of course this is mostly interesting for problems that are intractable for traditional compilation algorithms. An important focus of this task will be not only finding compilation algorithms for concrete parameterized problems but more generally exploring *approaches* to parameterized compilation that can then later be reused for other concrete problems. To this end, we will analyze which approaches from the field of parameterized algorithms can be combined with compilation to yield parameterized compilability results. For example, Bova et al. [2016b] showed that the classical notion of treewidth which is ubiquitous in parameterized algorithms can be extended to *treewidth modulo equivalence* to solve entailment problems for CNF. Here the necessary equivalence checks can, due to the allowed preprocessing, be performed in the compilation setting while they are impossible in parameterized algorithms alone. It appears likely that taking other approaches modulo the right notion of equivalence can be fruitful for parameterized compilation. Note that when allowing for large preprocessing times is insufficient for full equivalence checks to be feasible in practice, even incomplete equivalence checks may be useful, as shown by Bova et al. [2016b] who considered so-called backbones, i.e., sets of variables whose values are fixed in all satisfying assignments, whose computation is often feasible in practice.

Complementing the upper bounds, we will give conditional lower bounds for parameterized compilation problems. In fact, there are several different conditional lower bound frameworks in the literature (Chen [2005, 2015]; de Haan [2015]) which we are planning to apply to the problems we analyze. Besides being inherently interesting, the lower bounds we are aiming at are also important from a more practical point of view. In fact, having lower bounds for some parameterizations of a problem will guide the search for more useful parameterizations. If one parameter is ruled out, then to find better parameterizations one will either have to restrict the problem, e.g. by choosing a more restrictive parameter, or, in case the parameter is already very restrictive, change the approach to be able to capture meaningful classes of instances. Thus upper and lower bounds are inherently linked in order to explore the power of parameterized compilation.

Task 2.2: Compilation to concrete data structures. While Task 2.1 concentrates on general parameterized compilability results, in this task we will focus on compilation to established concrete data structures, in particular those studied in the classical KC map and in WP 1. While we cannot expect the resulting algorithms to be as strong as those from Task 2.1 in which we have more freedom in the compilation phase, targeting these data structures has great advantages. For example, after having compiled into such a form, we will be able to use results on the properties of these representations for different queries and transformations and even use software that is implemented to deal with this data structure.

Consequently, we will design algorithms that allow compilation of parameterized instances to these target representations (DNNF and its subsets). To this end, it will be interesting to understand which algorithmic techniques can be used in this setting. This is because, contrary to the setting of Task 2.1, there are even polynomial time solvable problems which do not compile to specific representations. For example, solutions of linear systems of equations cannot be compiled into the language of DNNF without an exponential size blow-up [Mengel, 2016]. We will also show unconditional parameterized lower bounds. Those are particularly interesting because, as it has long been observed, see e.g. [Huang and Darwiche, 2005], certain representation languages from compilation are strongly related to algorithmic paradigms, so showing lower bounds in KC unconditionally shows that certain algorithmic paradigms cannot solve a problem at hand efficiently. Carrying this over to parameterized complexity could thus give a new perspective on algorithmic approaches for parameterized problems.

As in Task 2.1, we will complement our algorithms with lower bounds showing that compilation into certain representations cannot be done in a fixed-parameter fashion. But in contrast to Task 2.1, we hope that these lower bounds will be unconditional. In fact, there is a recent line of lower bounds for representations used in KC mostly by using tools from communication complexity, see e.g. [Razgon, 2014; Mengel, 2016; Bova et al., 2016a], which we are planning to extend.

Participants. Julien Clément, Martin Cooper, Étienne Grandjean, Frédéric Koriche, Pierre Marquis, Stefan Mengel (coordinator), Alexandre Niveau, Romain Wallon, Ph.D. student X – for a total manpower of 55.9 person.month.

Schedule and deliverables. The research of both tasks will be performed during the whole span of the work package ($[t_0, t_0+24]$). We will write two progress reports: one at t_0+12, the other at t_0+24. The first report will present the state of the art in the literature as well as parameterized algorithms and lower bounds for classical problems from the literature, in particular those from Cadoli et al. [2002], and well-established languages such as those in [Darwiche and Marquis, 2002]. In the second report, we will extend the scope by presenting our work on the problems and languages considered in WP 1. We will also write a final report at t_0+30 to summarize our findings in this work package.

2.3.3 WP3: Fine-grained KC maps

The classical settings in which KC is studied take into account worst-case complexity, and study complexity modulo polynomial reductions. For instance, the definition of the class compP first requires that there exists a compilation function C and a polynomial p such that $C(F)$ has size upper bounded by $p(|F|)$ *for all* fixed parts F. As for on-line reasoning, the classical KC maps classify the complexity of queries as polynomial, NP-hard, or (unconditionally) not polynomial.

As in WP 2, in this work package we will start from the observation that such classifications do not always account for which compilation approaches can or cannot be useful in practice. Contrary to WP 2, however, we will mostly focus on the *analysis* of known target languages and transformations or queries, rather than on the *design* of new algorithms. Our main objective here is to design new approaches to the analysis of KC so as to better account for what is observed in practice.

As a concrete example, data structures like ordered binary decision diagrams (OBDDs, Bryant [1986]) are widely used in practice. Typically, the fixed part F initially encoded in CNF is compiled into an OBDD representation. However, there is no polynomial p guaranteeing that there exists an OBDD of size at most $p(|\varphi|)$ for all CNF formulae φ. In the same vein, in planning, a knowledge state represented by an OBDD will typically be updated by iteratively conjoining formulae representing the effects of actions. However, after an unbounded number of iterative updates the size of the result may blow up. Still, OBDDs are very useful for designing practical planning algorithms [Bertoli et al., 2006], just because (fortunately!) instances encoding concrete problems are generally not worst case instances.

In order to reduce this discrepancy between theory and practice of KC, this work package aims to use fine-grained concepts for analyzing the complexity of compilation functions and requests. More concretely, we will investigate the following approaches (detailed in the task descriptions):

- average-case and distribution analysis, under meaningful probabilistic models inspired from the applications (as studied in WP 4), complementary to worst-case analysis;
- analysis in terms of low complexity classes, in particular linear (rather than polynomial) time and space (see Grandjean and Schwentick [2002]; Grandjean and Olive [2004]), as opposed to polynomial vs. NP-hard analysis;
- parameterized complexity, where contrary to WP 2 we will focus on parameters defined on the compiled form and the query, without allowing the compiled form itself to have “parameterized polynomial size”;
- analysis in terms of approximate (or probably approximate) answers to requests, as opposed to exact compilation followed by exact querying – notably using tools from learning theory.

Since the analyses of the compilation phase (transforming the fixed part F into the compiled form $C(F)$) and of the on-line phase (answering a query V with the help of $C(F)$ and performing transformations of $C(F)$) are different in nature, the former referring more to size and the latter to time complexity, we will divide this work package into two tasks. Nevertheless, the study of transformations, which are typically performed on-line but raise questions about size, will act as a link between both tasks.

For both tasks, we will focus first on standard (propositional) languages and queries, as the tasks call for new analysis tools and these languages already lack such fine-grained analyses. However, depending on our progress, we will also consider the new languages and queries put forward by WP 1. We will also feed our results and tools back to WP 1 as early as possible, so that the design of new languages there can consider from the start the new points of view developed here.

Task 3.1: Fine-grained analysis of compilation. In this task, we will investigate the off-line phase in compilation, viz. the compilation of F into $C(F)$. Starting from applications, we will first identify interesting languages for F and $C(F)$, in the sense that these languages exhibit intractable theoretical (worst-case) behaviour but efficient behaviour in practice. The compilation of conjunctions of constraints into OBDDs will serve as a starting point, since it incurs a worst-case exponential blow-up while being very useful in practice.

Average complexity. Our first aim will be to account for the actual efficiency of the compilation. We will identify meaningful probabilistic models on the initial inputs (F) to the compilation phase, that is, models accounting for realistic distributions of instances in practice. For example, the model of k-CNFs will be a candidate for a probabilistic model on CNFs, as it allows us to model constraints which are bounded in length (a natural restriction for applications), and gives a not too intricate model for average-case studies. Based on such models, we will study the average-case and distribution behaviour (see Flajolet and Sedgewick [2009] for a presentation of the general methodology) of the compilation of CNF to OBDD – for instance, the mean size of the OBDD representation resulting from the transformation of a CNF formula, with expectation taken in the k-CNF model. Vuillemin and Béal [2004] have studied the average size of binary decisions diagrams considering random uniform boolean functions, but this distribution seems to be unnatural and is bound to lead to non-compact representations. Results showing the efficiency of a language with respect to natural distributions of the instances would give principled explanations of why the language is important in practice, while negative results would definitely rule out the language for the corresponding applications.

Approximate compilation. We will investigate approaches to compilation in which one does not require an *exact* representation $C(F)$ of the initial fixed part F. For instance, when the only queries consist in deciding whether F entails positive formulae, it is known that only a positive approximation of F is needed. If, moreover, the user can cope with approximate answers to queries (for instance, an approximation of the number of remaining possible products in a configuration setting, or an answer which is only 99 % likely to be correct), then we can use tools from learning theory (notably, Vapnik-Chervonenkis dimension and PAC analysis [Blumer et al., 1989], or Rademacher complexity [Bartlett and Mendelson, 2001]) to account for the possibility to tackle otherwise intractable compilation problems. This direction is closely related to average-case analyses since it involves probabilistic analyses of the inputs (fixed part and queries). Both approaches are in fact complementary, in the sense that typical analyses with tools from learning theory are distribution-independent, but it is also known that results from this field can be made more precise when specific distributions are assumed. Hence we can expect both approaches to feed each other with their respective results. Importantly also, such investigations will echo the investigation in WP 1 of requests from which the system must learn and revise its knowledge base, by giving a formal framework for formalizing this process (which depends on the distribution of requests).

Low complexity. As a bridge between this task and the next one, we will also investigate compilation from the point of view of linear, as opposed to polynomial, time and space. It is widely accepted that polynomial time/space analyses, though justified by theoretical considerations, do not accurately account for the efficiency of algorithms in practice. For instance, algorithms of effective running time in $\Theta(n^4)$ typically fail to be applicable in practice on moderately large inputs, while algorithms with linear running time (or $O(n\log n)$ running time) scale up well. Moreover, applying m successive transformations to an input of size n results in a double exponential blowup in size, that is n^{2^m}, if each transformation maps an input of size n to one of size n^2, while the overall process is more reasonably $c^m n$, for a constant c, if each transformation is linear. In particular, fixed-parameter complexity can distinguish these two cases if we take m as a parameter, which makes sense if, for instance, m is the number of successive choices made by a user when configuring a product. This leads to two main questions: (i) which languages support (possibly approximate) linear-space compilation procedures and linear-space transformations?, and (ii) which complexity-theoretical tools can be used to accurately account for compilation approaches which guarantee to support iterated transformations of a fixed part F? Hence our analyses here will indicate languages which meet (or do not meet) very stringent requirements.

Task 3.2: Fine-grained analysis of on-line queries. In this task we will investigate the on-line phase, i.e., the phase consisting of answering queries on the compiled form.

Linear and low complexity queries. It seems useless to have a compiled form $C(F)$ of size linear in that of F if queries require time $\Theta(n^3)$ to be answered. Hence we will design refined KC maps for on-line transformations and queries, which classify each as linear-time, quadratic, high-order polynomial, NP-hard, etc. A number of queries seem (hopefully) to be computable in linear time. Typically, for an OBDD of size s with n variables, one can count its models in time $O(s)$ and enumerate them with a delay $O(n)$ (between two successive solutions). More generally, it will be a realistic goal to determine to which extent the polynomial cases of the compilation map of Darwiche and Marquis [2002] can or cannot be improved to linear time or at least to quasi-linear or quadratic time. Related to the fine-grained goal of this WP, it is striking to observe that for a number of classes of queries in logic (e.g., first-order queries on degree-bounded structures) and in database theory (e.g., acyclic conjunctive queries), it has been established that the enumeration of the solutions of any query can be performed with a constant delay between two successive solutions after a linear time pre-computation (see e.g., Durand and Grandjean [2007]; Bagan et al. [2007, 2008]; Segoufin [2014]). In particular, Amarilli et al. [2017] have shown that compilation based techniques can be used in this context.

Average complexity of answering queries. As in the case of compilation, we will analyze on-line queries from the angle of average-case complexity. We will assume natural probability distributions on queries to be answered, while considering an already compiled fixed part $C(F)$. This will allow us to realistically account for the effective time taken to answer queries. For instance, in configuration one can expect the distribution of choices made by the user to be heavily biased towards a subset of the variables in the first place: the user will likely choose whether she wants a sports or a family car before choosing the color of seats. Analyzing the complexity of queries under such biased distributions may reveal complexities very different from those of worst-case analysis, providing new insights as to which languages are suited to given applications.

Parameterized complexity of answering queries. Complementary to the work planned in WP 2, our work here will concentrate on parameters defined for the compiled form $C(F)$ or for the request itself, without allowing a relaxed bound on the compiled form. Such analyses will typically be useful when the knowledge base evolves over time (such as for the aforementioned example of configuration, where the system is expected to learn from past user choices). Indeed, in this case the overall knowledge is only present in the data structure used on-line – the current knowledge base has no “initial form” of which the data structure would be a “compiled form”. It is thus natural to consider parameters of the data structure, independently from any parameterization of the off-line knowledge base. As for requests, in many settings it is natural to associate parameters to them as well, e.g. the size of a disjunction φ in entailment queries of the form $C(F) \models \varphi$: bounding the size of φ by a constant is often too restrictive in applications, but assuming that it is small in practice is reasonable. For instance, in configuration, one would expect a human user to choose between a small number of types of cars.

Approximate answers to queries. Finally, we will investigate the approximability of queries given exact compiled forms (contrary to Task 3.1 where we will investigate approximate compilation). Approximation makes sense in particular for optimization and counting problems – for instance, counting the number of products satisfying the current choices of the user in a configuration setting: typically, the user will be satisfied with a rounded answer, such as 1 000 000, even if the real number is, say, 1 002 946. Building maps of approximability for such requests depending on the compilation language will give an account of the practical usefulness of languages: when some optimization problems are hard from the point of view of worst-case complexity, relying on high-confidence algorithms for answering them may be enough in practice while much more efficient than producing exact results.

As a final note, let us stress that the different fine-grained analysis tools which we will consider in this WP will go hand in hand. For instance, linear space or time might not be achievable for a particular task and language, but may become so if approximate answers to requests are considered or if a nonlinear or non-polynomial dependence on a parameter is allowed; average-case analysis may also exhibit low complexities for some tasks which would otherwise be “only” polynomial; searching for natural parameters may suggest natural distributions for average-case analysis, for instance, distributions biased towards instances with a small parameter value; etc. Hence we expect the work in this WP to benefit from the interactions between seemingly distinct approaches. For this we will focus on few problems (requests and languages) with many approaches rather than considering many different problems.

Participants. Julien Clément, Hélène Fargier, Étienne Grandjean, Frédéric Koriche, Jean-Marie Lagniez, Pierre Marquis, Stefan Mengel, Alexandre Niveau (coordinator), Bruno Zanuttini, Ph.D. student X – for a total manpower of 66.1 person.month.

Schedule and deliverables. Both tasks will be performed during $[t_0, t_0 + 36]$, in parallel since the theoretical tools will be common to both tasks. As for WP 2, we will write progress reports after each year, at $t_0 + 12$, $t_0 + 24$, and $t_0 + 36$, on the progress made with each approach. The first report will survey known results and approaches on fine-grained compilation maps: known results about linear-time vs (only) polynomial-time for queries, known results in probabilistic models for average-case analysis, meaningful probabilistic models, etc. The second one will report on new complexity analyses in the refined frameworks, as produced in the WP. The third one will report on new complexity analyses produced during the third year, but also on new algorithms (efficient in the refined sense) and on advances on frameworks for studying low complexity and approximability. We will also write a final report at $t_0 + 42$ to summarize the findings of this work package.

2.3.4 WP4: Beyond standard benchmarks and applications for KC

The previous packages seek for strong theoretical and/or general results on complex KC languages, operations, or queries. We want to focus on a complementary view on KC, and in particular, to exploit the results of the other work packages in an application-driven package, enforcing a cross-fertilization between KC experts and experts from other fields (robotics and tentatively also bioinformatics). The idea here is to be able to make a strong effort on efficient implementations, with the objective to get significant results at the end of the project. The adoption of KC techniques in our targeted applications will be a clear and objective indicator of the success of this WP 4. However, as mentioned above, WP 4 will also give KC experts a set of real needs from real-world applications, allowing them to irrigate the above WPs. In addition to applications of KC to problems which have already been considered so far in AI (mainly, product configuration and classical planning) and for which benchmarks exist, we would like to be more adventurous and tackle problems outside the standard AI scope. We do not underestimate the difficulties of this goal, because some of our targeted applications already have pragmatic solutions (some of them have a critical real-time aspect), so our own propositions will need to have significant advantages in order to compete with established practice. Moreover, formalizing what experts from fields outside of KC expect may also be challenging. This is why we think that this kind of important cross-fertilization may only be possible with the help of a common project, like PING/ACK.

This last part of the project is thus open to new problems. We expect here to gather other problems where KC can play a positive role. We will try to build a suitable set of interesting problems for KC that needs an increasing expressiveness of languages, queries, and/or operations. This WP will offer the opportunity to confront with complex, real-world problems the expressiveness of the target languages identified in WP1 (and the efficiency of the corresponding compilers), but also the theory-oriented results about parameterized complexity obtained in WP2 and the fine-grained maps resulting from WP3. For instance, KC considers as "easy" any task that is feasible in polynomial time. In real-time applications, this fundamental principle may not hold anymore in the general case. What defines a "feasible" on-line query or operation? This question, which is at the core of WP2 and WP3, will here be investigated from a practical standpoint through the prism of two applications.

The first application we want to target, namely real-time decisions of humanoid robots, is a good example of the goal of this WP. It also explains why designing a specific WP for applications (instead of illustrating the above WPs by adding an experimental section to each of them) was mandatory.

Task 4.1: Real-time decisions for robots. Our first application is about real-time decisions of humanoid robots (kid-sized) in the context of a soccer game. Low-level motion decisions for such robots are probably not directly concerned by KC approaches. However, in more complex situations where higher-level real-time planning is needed, KC may be a good option. Let us describe here how the Rhoban team ⟨http://rhoban.com/fr⟩ managed to solve this problem [Hofer and Rouxel, 2017], which allowed them to win the last soccer cup in their category.[4] During a game, the robot is autonomous: a small embedded x86 computer runs the robot's main program. It reads motors, sensors and camera inputs, analyses and computes the current internal and external state and issues to the servo-motors the next target position at about 100 Hz, which is a very strong real-time constraint. One of the main difficulties the computed policy has to face is the uncertainty on the soccer field, such as slides of the feet on the field, or falls and collisions affecting sensor precision. The proposed method [Hofer and Rouxel, 2017] can be described as an *ad hoc* KC method, based on a pragmatic solution mixing expert knowledge and machine learning: the experts designed a very small algorithm with parameters, which was then tuned by a black-box optimization method, the goal of which was essentially to circumvent as many noisy situations as possible. The final (learned) policy is built on top of a Markov Decision Process (MDP) allowing the robot to query a real-time decision algorithm (at 100 Hz). Pushing the classical framework of KC to this use case is very challenging and presents a clear risk. However, we have already identified interesting starting points to look at. For instance, the notion of *local planning* [Weinstein and Littman, 2013] seems to be a good candidate to formalize what could be expected in real time. Moreover, the work of Zamani et al. [2012] already defined a suitable target language to look at, i.e., a continuous variable extension of the language of algebraic decision diagrams (ADDs).

If this starting point seems reasonable for the current game rules, where taking a few seconds to adjust the kick is the norm, organizers are planning to extend the rules in 2020, using a wider soccer field (20m long) and 6 players instead of 4. In this context, higher-level strategies must be added to the robot (such as an attack/defense strategy, or sending the ball

---

[4] RoboCup Humanoid League rules are available at ⟨https://www.robocuphumanoid.org/materials/rules/⟩.

to an attacker). The game duration will also be extended to 20 minutes, allowing more strategical dribbles to be planned. In this context, it may not be suitable to extend the current *ad hoc* approach, because the small algorithm written by an expert may quickly grow in complexity. We propose to handle the whole process directly from a higher-level description of strategies.

We thus need to use new target languages, including temporal features and suitable for fast, reactive planning of game actions (possibly through the use of approximate compilation techniques). We expect these target languages to be formalized out of the box: the target language will be the input language for the machine learning step that allows it to handle all the noisy situations (see Hofer and Rouxel [2017]). Including the learning step in the compilation paradigm is another possible solution that we will consider. In any case, as we have seen, we expect the cross-fertilization of robotics and more theoretic AI experts to have a deep impact on both fields.

Task 4.2: Metabolic networks. Metabolic networks are used to formalize all the possible modes of operation of a cell, in terms of possible metabolic productions. Laurent Simon has already worked with Sabine Pérès and Philippe Dague from LRI (Orsay) to use SAT solvers to query such kind of networks. However, some networks have millions and millions of solutions, and a KC approach thus seems a good way to circumvent this complexity. The problem here is that biologists use a notion of “minimal pathway” that is in some sense a minimal model w.r.t. composition (a model is minimal if it cannot be split into other models). This kind of minimality has to be studied from a KC perspective. Moreover, biologists may not be interested in exact counting of solutions but in an approximation. We propose in this work package to study how KC can help biologists to navigate efficiently in the set of minimal solutions. We are absolutely convinced that there is room for improvements: the state of the art is currently to explicitly represent all these pathways before exploring them (sometimes in an Excel spreadsheet, by hand!) which immediately condemns any approach addressing genomes of human-level size. Although Sabine Pérès and Philippe Dague (LRI) are not members of the project, we will invite them to expose their problems to the group. We will also try to express pathways at the logical level, depending on which queries are interesting for biologists and what operations are typically done. The goal here is twofold: we want to formalize the set of possible queries and to see, at the theoretical level, what KC can do for this kind of minimality. We expect this task to take advantage of the results of Task 1.1 of WP 1 (compilation of weight/preference distributions). Moreover, we think that this application is a good candidate to be used as a new benchmark for experimentally evaluate the preference languages proposed in WP 1.

In summary, we want this work package to provide a good experimental testbed of techniques proposed in the former work packages. We will collect a library of KC problems with increasing expressiveness of language/queries/operations to be used to feed and assess the performances of techniques proposed in the project. Reciprocally, we will make a strong implementation effort to turn the experimental part of the project into a real application-driven package (decision for robots). Finally, beyond the definition of new benchmarks and the application of the results of the project in real-world applications, our strategy in WP 4 will be to invite researchers from a number of domains (starting with the above ones) to expose their problems during our meetings. We think that cross-fertilization is a problem by itself and that an effort must be done in this direction.

Participants. Hélène Fargier, Hugo Gimbert, Jean-Marie Lagniez, Olivier Ly, Frédéric Maris, Pierre Marquis, Jérôme Mengin, Alexandre Niveau, Laurent Simon (coordinator), Ph. D. student Y, post-doc Z – for a total manpower of 61.2 person.month.

Schedule and deliverables. The two tasks will be performed during $[t_0+6, t_0+48]$, in parallel. They will give rise to a number of reports: a first report on the applications identified as interesting for KC at t_0+12, a second report at t_0+18 about the input languages and formalization of the requests/operations required by the two tasks, a third report at t_0+24 on “Robotics and KC”, with a prototype tool embedded in the Rhoban robot, a description of the new benchmarks provided by the two tasks at t_0+30, a fourth report at t_0+36 on “Metabolic networks”, a description of the empirical results obtained on the new benchmarks at t_0+42, and a final report at the end of the project (t_0+48).

2.3.5 WP5: Coordination of the project and dissemination of the results

This work package consists of two tasks:

Task 5.1: Coordination. Concerning global coordination, the bi-annual scientific meetings of the group will systematically include a meeting of the coordination team (comprised of the WP and laboratory leaders) for managing the project and concluding the progress reports. These meetings will be complemented by video conferences on a regular basis. We will also pay particular attention to the supervision of the two Ph.D. students and the post-doc hired by the project. Each of them is expected to spend half of his/her time in each of the two labs involved in his/her supervision.

Task 5.2: Diffusion. One of the objectives of PING/ACK is to promote research on KC by making all the results obtained during the project publicly available on a project website. We plan also to feed the “Beyond NP” collaborative website (see ⟨http://beyondnp.org/about/⟩) as this is a good means for wider, international dissemination of the results of the project (including the resources generated thanks to it: papers, software, benchmarks, etc.). A final workshop co-held with a top-ranked international conference will be organized at the end of year 4. The spirit is the same as that of the "Beyond NP" workshop held with AAAI'16 and co-organized by the project coordinator (but the PING/ACK workshop will be fully focused on KC). We also plan to prepare a special issue of an international journal, as a follow-up to the workshop.

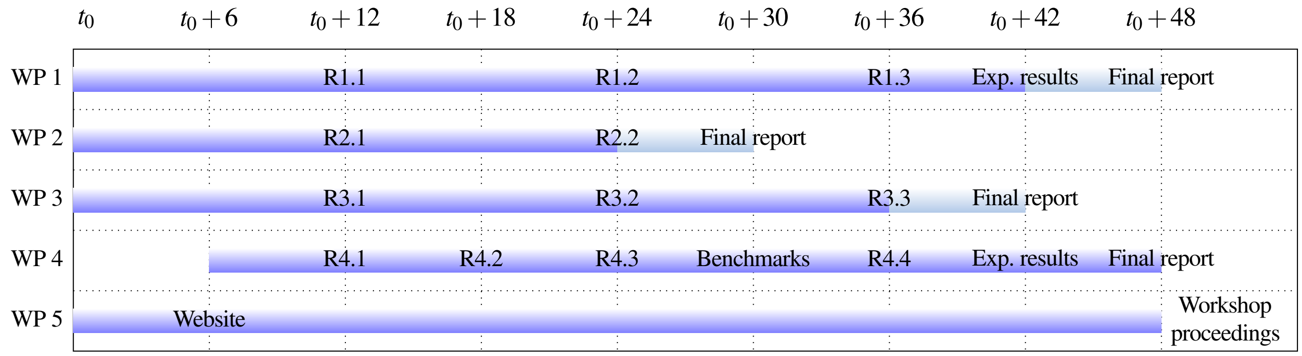
Participants. Hélène Fargier, Pierre Marquis (coordinator), Stefan Mengel, Alexandre Niveau, Laurent Simon – for a total manpower of 8.6 person.month.

Schedule and deliverables. Both Tasks 5.1 and 5.2 will last for the entire duration of the project. As for the deliverables:

- A website for the project will be built; it will serve as a repository and will include at the end of the 4 years all the material generated by the project, in particular: description of the project, deliverables, resulting publications and pieces of software, description of the benchmarks and protocols used.
- The proceedings of the final workshop will be collected and possibly published.

2.4 Partners' contributions, schedule of the project and deliverables

Gantt chart. The following chart summarizes the timing of the project and the dates planned for the deliverables; Ri.j denotes the j^{th} intermediate reports of WP i (see the description of the work packages for more details).



Distribution of the effort. The distribution of the effort (person.month) and the responsibilities per WP are reported in Table 1.

Partner	Name	First name	Implication	WP1	WP2	WP3	WP4	WP5
CRIL	Koriche	Frédéric	12	4.6	2.4	5	0	0
CRIL	Lagniez	Jean-Marie	19.2	6	0	6	7.2	0
CRIL	Marquis	Pierre	24	5	6	5	5	3 (leader)
CRIL	Mengel	Stefan	14.4	2.4	7.5 (leader)	3.5	0	1
CRIL	Wallon	Romain	6	0	6	0	0	0
GREYC	Clément	Julien	12	0	4	8	0	0
GREYC	Grandjean	Étienne	12	4	4	4	0	0
GREYC	Niveau	Alexandre	19.2	7	2	7.2 (leader)	2	1
GREYC	Zanuttini	Bruno	12	6	0	6	0	0
IRIT	Cooper	Martin	12	6	6	0	0	0
IRIT	Fargier	Hélène	14.4	5 (leader)	0	4.4	3	2
IRIT	Maris	Frédéric	14.4	10	0	0	4.4	0
IRIT	Mengin	Jérôme	12	6	0	0	6	0
LaBRI	Gimbert	Hugo	7.2	2.4	0	0	4.8	0
LaBRI	Ly	Olivier	7.2	2.4	0	0	4.8	0
LaBRI	Simon	Laurent	9.6	2	0	0	6 (leader)	1.6
CRIL/GREYC	X	X	36	0	18	18	0	0
GREYC/IRIT	Y	Y	36	30	0	0	6	0
IRIT/LaBRI	Z	Z	12	0	0	0	12	0

Table 1: Distribution of the effort (person.month) and responsibilities per WP.

2.5 Justification of the budget.

The requested budget for PING/ACK is 432 081 € (including the management costs of 8 %). This budget corresponds to the funding of two Ph.D. theses (2 × 3 years: 188 640 €); one post-doc for one year (51 935 €); two two-day project meetings for each of the 4 years (400 € each for each of the 16 participants from group A, except Romain Wallon who will participate only for 2 years in the project – 49 600 €), plus six two-day meetings per year of participation for the 3 participants from group $\bar{A}$ (16 800 €) in order to facilitate their co-supervision; participation in the budget required for attending the national and international conferences considered for disseminating the results (3600 € per participant from group A = 57 600 €, plus 2400 € per Ph.D. student from group $\bar{A}$ = 4800 €, plus 1200 € for the post-doc, for a total of 63 600 €); and additional operating costs (internships, equipment, use of the CRIL clusters: 29 500 €). This leads to a grand total of 400 075 €.

The distribution of the budget per lab (not including the management costs) is given in Table 2. The total cost (without overheads), the implication (excluding the participants funded by the project), and the funding requested (including the management costs) are given in Table 3.

	CRIL	GREYC	IRIT	LaBRI	Total
Meetings	21 600	20 000	15 200	9 600	66 400
Conferences	20 400	16 800	15 600	10 800	63 600
Hardware and operating costs	8 500	7 000	7 000	7 000	29 500
Ph. D. X	94 320				94 320
Ph. D. Y		94 320			94 320
Post-doc Z			51 935		51 935
Total	144 820	138 120	89 735	27 400	400 075

Table 2: Distribution of the budget per lab.

	CRIL	GREYC	IRIT	LaBRI	Total
Total cost	505 110	345 131	361 552	138 230	1 350 023
Implication	75.6	55.2	52.8	24	207.6
Funding requested	156 406	149 169	96 914	29 592	432 081

Table 3: Total cost, implication, and funding requested per lab.

3 Impact and benefits of the project

Immersed in the era of Big Data, our economic society is witnessing a fundamental shift to digital channels. Consumers, scientists and business professionals are increasingly using the Web to browse, discover, download, buy and recommend – a transition that is creating new opportunities, together with new challenges, for the development of on-line applications and softbots. Similarly, the development and the expectations concerning autonomous robots serving as assistants in everyday life tasks are growing quickly. Many applications are envisioned, especially the support for elderly and dependent people in their homes, which will be a major socio-economical issue in many countries for the next decades.

Many of such softbots or robots require intricate interactions with humans, and as such, they call for guaranteed response times. As an example, this is the case of product configuration applications (cars, computers, bikes, houses, menus, etc.), which aims to help a user to choose a product by navigating in a set of feasible objects which is implicitly represented (an explicit catalog gathering all feasible products would be too large to be stored). Importantly, an essential requirement for such applications is to ensure timely responses for the services used: as a matter of fact, when she interacts with an application through the Internet, the user is ready to wait only for (at most) a couple of seconds for the information she is looking for.

Thus, as users spend more and more time on the Web, their requirements of a satisfactory on-line experience are changing; high-bandwidth connections have trained users to require seamless, fast performance. Orthogonally, users have come to expect more relevant and personalized assistance that adapt to an individual based on her locations, interests, constraints, preferences, and activities. Such high added-value interactions are beyond the reach of standard systems. They require sophisticated and computationally expensive information processing techniques for extracting and integrating, in a meaningful way, the different pieces of knowledge describing the user profile and the service she is asking for. They also ask for the development of models, techniques and algorithms for ensuring efficient interactions in terms of response times.

As detailed in the previous sections, KC has been developed as a valuable approach for guaranteed response times, contributing as such to the general, large-scope issues sketched above. The PING/ACK project precisely aims to extend the scope of the KC approach to problem solving, from the theoretical side to the practical side, in order to better fit the requirements imposed by complex applications. In particular, the focus in PING/ACK will be laid on representation languages more expressive than existing ones. The applicability of KC will also be improved, via the identification of more fine-grained maps and the evaluation of additional approaches for enhancing the benefits that KC offers.

Thus, PING/ACK applies to "DÉFI 7 – SOCIÉTÉ DE L'INFORMATION ET DE LA COMMUNICATION, Axe 1 : *Socle Fondements du numérique*". From a scientific point of view, the very objective of PING/ACK is to contribute to a better understanding of the time/space trade-offs which can be reached via a preprocessing of some parts of the available information. Indeed, the study of such trade-offs is a fundamental issue in the theory of computation, and this explains why the PING/ACK project is fully relevant to Axe 1: "*Socle Fondements du numérique*". The scientific issues that will be analyzed also concern the foundations of artificial intelligence. Given that for the very recent past, KC has been spreading over a number of domains that go beyond knowledge representation (in particular, theoretical computer science, database theory, and machine learning; see e.g. ⟨http://www.dagstuhl.de/17381⟩ and [Darwiche et al., 2017]), we are confident that this is the right time for such a project. Note that an avatar of PING/ACK has been retained last year by ANR in the supplementary list of selected projects, but unfortunately it was not among those which have finally been funded.

The expected benefits of the PING/ACK project are mainly of a scientific nature, at least from a short-term perspective. Nevertheless, we hope that the project will be successful enough to have a significant impact on several, if not all, the application areas that will be considered during the project. Thus, while a large part of the results will be theory-oriented, the choices of the languages to be defined and studied in WP 1, the compilability problems considered in WP 2, the requests analyzed in WP 3, and the compilers and reasoners developed and evaluated in WPs 1 and 4 will be guided by the applications we target, which are relevant to planning, configuration, autonomous robots, and bioinformatics.

Specific care has been given to the dissemination strategy, via an additional work package (WP 5) that is partly dedicated to it. The expected outcomes of the PING/ACK project will take the form of scientific publications in top-tier international journals and conferences, as well as some pieces of software. In the AI field, the best international journals like Artificial Intelligence and the Journal of AI Research will be targeted; first-rank conferences like IJCAI, AAAI, CP, SAT will be considered as well for a quicker dissemination of the results within the AI community. The diffusion of software (compilers and reasoners) will be open-source, so that the code developed can be used by others after the end of PING/ACK. Obviously, as mentioned in WP 5, a website associated with the project will be built and maintained. In addition, we plan to organize a couple of events of the same kind as the (already mentioned) symposium on KC in Vienna (held in 2015) and the Dagstuhl seminar on KC (held in 2017). In particular, we plan to organize a final international workshop at the end of the project for promoting further the results obtained. The workshop will be co-located with a major general-scope AI conference (IJCAI or AAAI) in order to attract scientists from many AI areas (including knowledge representation, constraint programming, planning, and machine learning). As a follow-up, some of us plan to serve as the editors of a special issue of a journal collecting the best papers of the workshop. We will also take advantage of the Robocup events (and specifically the one that will be organized in 2020 and that the LaBRI group is a candidate for hosting) to promote the results of the project, and of the "Beyond NP" collaborative website for wider dissemination of the results of PING/ACK.

Finally, each of the participants in the PING/ACK project will use his/her research networks to disseminate the results to a much larger extent. Indeed, for the past few years, KC has become quite a "hot topic" that is no longer limited to the AI area but spreads over several communities within Computer Science (among others, algorithms, computational complexity, databases, and formal verification). We are connected with several internationally recognized researchers from each of these areas, and we will use those connections for leveraging the results of PING/ACK.

References

A. ALBORE, H. PALACIOS, and H. GEFFNER. 'Compiling Uncertainty Away in Non-Deterministic Conformant Planning.' In *ECAI*, vol. 215 of *Frontiers in Artificial Intelligence and Applications*, pp. 465–470. IOS Press, 2010.

J. F. ALLEN. 'Maintaining Knowledge About Temporal Intervals.' *Communicatons of the ACM*, 26(11):832–843, 1983.

A. AMARILLI, P. BOURHIS, L. JACHIET, and S. MENGEL. 'A Circuit-Based Approach to Efficient Enumeration.' In *Proceedings of ICALP'17*, vol. 80 of *LIPIcs*, pp. 111:1–111:15. 2017.

J. Amilhastre, H. Fargier, and P. Marquis. ‘Consistency Restoration and Explanations in Dynamic CSPs: Application to Configuration.’ *Artificial Intelligence*, 135(1-2):199–234, 2002.

J.-M. Astesana, L. Cosserat, and H. Fargier. ‘Constraint-based Vehicle Configuration: A Case Study.’ In *Proceedings of ICTAI’10*, pp. 68–75. 2010.

G. Audemard, J.-M. Lagniez, and L. Simon. ‘Just-In-Time Compilation of Knowledge Bases.’ In *Proceedings of IJCAI’13*. 2013.

F. Bacchus and A. Grove. ‘Graphical Models for Preference and Utility.’ In *Proceedings of UAI’95*, pp. 3–10. 1995.

G. Bagan, A. Durand, and E. Grandjean. ‘On Acyclic Conjunctive Queries and Constant Delay Enumeration.’ In *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, 2007, Proceedings*, pp. 208–222. 2007.

G. Bagan, A. Durand, E. Grandjean, and F. Olive. ‘Computing the jth solution of a first-order query.’ *ITA*, 42(1):147–164, 2008.

C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli. ‘Satisfiability Modulo Theories.’ In *Handbook of Satisfiability*, pp. 825–885. 2009.

A. Bart, F. Koriche, J.-M. Lagniez, and P. Marquis. ‘An Improved CNF Encoding Scheme for Probabilistic Inference.’ In *Proceedings of ECAI’16*, pp. 613–621. 2016.

P. L. Bartlett and S. Mendelson. ‘Rademacher and Gaussian Complexities: Risk Bounds and Structural Results.’ In *COLT/EuroCOLT*, vol. 2111 of *Lecture Notes in Computer Science*, pp. 224–240. Springer, 2001.

G. Behrmann, K. G. Larsen, J. Pearson, C. Weise, and W. Yi. ‘Efficient Timed Reachability Analysis Using Clock Difference Diagrams.’ In N. Halbwachs and D. A. Peled, eds., *CAV ’99*, vol. 1633 of *Lecture Notes in Computer Science*, pp. 341–353. Springer, 1999.

P. Bertoli, A. Cimatti, M. Roveri, and P. Traverso. ‘Strong planning under partial observability.’ *Artificial Intelligence*, 170(4-5):337–384, 2006.

M. Bienvenu, H. Fargier, and P. Marquis. ‘Knowledge Compilation in the Modal Logic S5.’ In *Proceedings of AAAI’10*. 2010.

D. Bigot, H. Fargier, J. Mengin, and B. Zanuttini. ‘Using and Learning GAI-Decompositions for Representing Ordinal Rankings.’ In J. Fürnkranz and E. Hüllermeyer, eds., *Proceedings of the ECAI’12 workshop on Preference Learning*, pp. 5–10. 2012.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. ‘Learnability and the Vapnik-Chervonenkis dimension.’ *Journal of the ACM*, 36(4):929–965, 1989.

R. Booth, Y. Chevaleyre, J. Lang, J. Mengin, and C. Sombattheera. ‘Learning conditionally lexicographic preference relations.’ In *Proceedings of ECAI 2010*, pp. 269–274. IOS Press, 2010.

C. Boutilier, F. Bacchus, and R. I. Brafman. ‘UCP-Networks: A Directed Graphical Representation of Conditional Utilities.’ In *Proceedings of UAI’01*, pp. 56–64. 2001.

C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. ‘CP-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements.’ *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

S. Bova, F. Capelli, S. Mengel, and F. Slivovsky. ‘On Compiling CNFs into Structured Deterministic DNNFs.’ In *Proceedings of SAT’15*, pp. 199–214. 2015.

S. Bova, F. Capelli, S. Mengel, and F. Slivovsky. ‘Knowledge Compilation Meets Communication Complexity.’ In *Proceedings of IJCAI’16*, pp. 1008–1014. 2016a.

S. Bova, R. de Haan, N. Lodha, and S. Szeider. ‘Positive and Negative Results for Parameterized Compilability.’ Tech. rep., Algorithms and Complexity Group, TU Wien, 2016b.

A. R. Bradley. ‘IC3 and beyond: Incremental, Inductive Verification.’ In *Proceedings of CAV 2012*, p. 4. 2012.

R. I. Brafman and C. Domshlak. ‘Introducing variable importance tradeoffs into CP-nets.’ In *Proceedings of UAI’02*, pp. 69–76. 2002.

R. Bryant. ‘Graph-based algorithms for Boolean function manipulation.’ *IEEE Transactions on Computers*, C-35(8):677–692, 1986.

J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and L. J. Hwang. ‘Symbolic Model Checking: 10ˆ20 States and Beyond.’ *Inf. Comput.*, 98(2):142–170, 1992.

M. Cadoli and F. Donini. ‘A survey on knowledge compilation.’ *AI Communications*, 10(3–4):137–150, 1998.

M. Cadoli, F. Donini, P. Liberatore, and M. Schaerf. ‘Space efficiency of propositional knowledge representation formalisms.’ *Journal of Artificial Intelligence Research*, 13:1–31, 2000.

M. CADOLI, F. M. DONINI, P. LIBERATORE, and M. SCHAERF. ‘Preprocessing of Intractable Problems.’ *Information and Computation*, 176(2):89–120, 2002.

M. CHAVIRA and A. DARWICHE. ‘On probabilistic inference by weighted model counting.’ *Artificial Intelligence*, 172(6-7):772–799, 2008.

M. CHAVIRA, A. DARWICHE, and M. JAEGER. ‘Compiling relational Bayesian networks for exact inference.’ *International Journal of Approximate Reasoning*, 42(1-2):4–20, 2006.

H. CHEN. ‘A Theory of Average-Case Compilability in Knowledge Representation.’ In *Proceedings of IJCAI’03*, pp. 455–460. 2003.

H. CHEN. ‘Parameterized Compilability.’ In *Proceedings of IJCAI’05*, pp. 412–417. 2005.

H. CHEN. ‘Parameter Compilation.’ In *IPEC*, vol. 43 of *LIPIcs*, pp. 127–137. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

Y. CHEVALEYRE, F. KORICHE, J. LANG, J. MENGIN, and B. ZANUTTINI. ‘Learning Ordinal Preferences on Multiattribute Domains: the Case of CP-nets.’ In J. FÜRNKRANZ and H. HÜLLERMEIER, eds., *Preference learning*, pp. 273–296. Springer, 2011.

J. CLÉMENT, T. H. N. THI, and B. VALLÉE. ‘Towards a Realistic Analysis of Some Popular Sorting Algorithms.’ *Combinatorics, Probability & Computing*, 24(1):104–144, 2015.

S. COSTE-MARQUIS, J. LANG, P. LIBERATORE, and P. MARQUIS. ‘Expressive Power and Succinctness of Propositional Languages for Preference Representation.’ In *Proceedings of KR’04*, pp. 203–212. 2004.

A. DARWICHE. ‘Decomposable Negation Normal Form.’ *Journal of the ACM*, 48(4):608–647, 2001.

A. DARWICHE. ‘A differential approach to inference in Bayesian networks.’ *Journal of the ACM*, 50(3):280–305, 2003.

A. DARWICHE and P. MARQUIS. ‘A Knowledge Compilation Map.’ *Journal of Artificial Intelligence Research*, 17:229–264, 2002.

A. DARWICHE and P. MARQUIS. ‘Compiling Propositional Weighted Bases.’ *Artificial Intelligence*, 157(1-2):81–113, 2004.

A. DARWICHE, P. MARQUIS, D. SUCIU, and S. SZEIDER. ‘Recent Trends in Knowledge Compilation (Dagstuhl Seminar 17381).’ *Dagstuhl Reports*, 7(9):62–85, 2017.

R. DE HAAN. ‘An Overview of Non-Uniform Parameterized Complexity.’ *Electronic Colloquium on Computational Complexity*, 22:130, 2015.

R. DECHTER, I. MEIRI, and J. PEARL. ‘Temporal Constraint Networks.’ *Artificial Intelligence*, 49(1-3):61–95, 1991.

R. G. DOWNEY and M. R. FELLOWS. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

A. DURAND and E. GRANDJEAN. ‘First-order queries on structures of bounded degree are computable with constant delay.’ *ACM Transactions on Computational Logic*, 8(4), 2007.

R. EHLERS, D. FASS, M. GERKE, and H.-J. PETER. ‘Fully Symbolic Timed Model Checking Using Constraint Matrix Diagrams.’ In *Proceedings of RTSS 2010*, pp. 360–371. 2010.

H. FARGIER, F. MARIS, and V. ROGER. ‘Temporal Constraint Satisfaction Problems and Difference Decision Diagrams: a Compilation Map.’ In *Proceedings of ICTAI’15*, pp. 429–436. 2015.

H. FARGIER and P. MARQUIS. ‘Disjunctive Closures for Knowledge Compilation.’ *Artificial Intelligence*, 216:129–162, 2014.

H. FARGIER, P. MARQUIS, and A. NIVEAU. ‘Towards a Knowledge Compilation Map for Heterogeneous Representation Languages.’ In *Proceedings of IJCAI’13*, pp. 877–883. 2013a.

H. FARGIER, P. MARQUIS, A. NIVEAU, and N. SCHMIDT. ‘A Knowledge Compilation Map for Ordered Real-Valued Decision Diagrams.’ In *Proceedings of AAAI’14*, pp. 1049–1055. 2014.

H. FARGIER, P. MARQUIS, and N. SCHMIDT. ‘Semiring Labelled Decision Diagrams, Revisited: Canonicity and Spatial Efficiency Issues.’ In *Proceedings of IJCAI’13*, pp. 884–890. 2013b.

P. FLAJOLET and R. SEDGEWICK. *Analytic Combinatorics*. Cambridge University Press, 2009.

M. A. FLIGNER and J. S. VERDUCCI. *Probability models and statistical analyses for ranking data*, vol. 80. Springer, 1993.

N. FRIEDMAN and Z. YAKHINI. ‘On the Sample Complexity of Learning Bayesian Networks.’ In *Proceedings of UAI’96*, pp. 274–282. 1996.

G. GOGIC, H. A. KAUTZ, C. H. PAPADIMITRIOU, and B. SELMAN. ‘The Comparative Linguistics of Knowledge Representation.’ In *Proceedings of IJCAI’95*, pp. 862–869. 1995.

E. GRANDJEAN and F. OLIVE. ‘Graph properties checkable in linear time in the number of vertices.’ *J. Comput. Syst. Sci.*, 68(3):546–597, 2004.

E. GRANDJEAN and T. SCHWENTICK. ‘Machine-Independent Characterizations and Complete Problems for Deterministic Linear Time.’ *SIAM J. Comput.*, 32(1):196–230, 2002.

L. HOFER and Q. ROUXEL. 'An Operational Method Toward Efficient Walk Control Policies For Humanoid Robots.' In *Proceedings of ICAPS'17*. 2017. 489-497.

J. HOFFMANN and R. I. BRAFMAN. 'Contingent Planning via Heuristic Forward Search witn Implicit Belief States.' In *Proceedings of ICAPS'05*, pp. 71–80. 2005.

J. HUANG and A. DARWICHE. 'DPLL with a Trace: From SAT to Knowledge Compilation.' In L. P. KAELBLING and A. SAFFIOTTI, eds., *Proceedings of IJCAI'05*, pp. 156–162. 2005.

D. KISA, G. V. DEN BROECK, A. CHOI, and A. DARWICHE. 'Probabilistic Sentential Decision Diagrams.' In *Proceedings of KR'14*. 2014.

F. KORICHE, J.-M. LAGNIEZ, P. MARQUIS, and S. THOMAS. 'Compiling Constraint Networks into Multivalued Decomposable Decision Graphs.' In *Proceedings of IJCAI'15*, pp. 332–338. 2015.

J. LANG and B. ZANUTTINI. 'Knowledge-Based Programs as Plans: Succinctness and the Complexity of Plan Existence.' In *Proceedings of TARK'13*. 2013.

P. LIBERATORE. 'On the compilability of diagnosis, planning, reasoning about actions, belief revision, etc.' In *Proceedings of KR'98*, pp. 144–155. 1998.

P. LIBERATORE. 'Monotonic reductions, representative equivalence, and compilation of intractable problems.' *Journal of the ACM*, 48(6):1091–1125, 2001.

P. LIBERATORE. 'The Size of MDP Factored Policies.' In *Proceedings of AAAI'02*, pp. 267–272. 2002.

P. LIBERATORE and M. SCHAERF. 'Compilability of propositional abduction.' *ACM Transaction of Computational Logics*, 8(1):2, 2007.

T. LU and C. BOUTILIER. 'Effective sampling and learning for Mallows models with pairwise-preference data.' *Journal of Machine Learning Research*, 15(1):3783–3829, 2014.

P. MARQUIS. 'Compile!' In *Proceedings of AAAI'15*, pp. 4112–4118. 2015.

S. MENGEL. 'Parameterized Compilation Lower Bounds for Restricted CNF-Formulas.' In *Proceedings of SAT'16*, vol. 9710, pp. 3–12. 2016.

P. MESEGUER, F. ROSSI, and T. SCHIEX. 'Soft Constraints.' In *Handbook of Constraint Programming*, pp. 281–328. 2006.

S. MINATO, K. SATOH, and T. SATO. 'Compiling Bayesian Networks by Symbolic Probability Calculation Based on Zero-Suppressed BDDs.' In *Proceedings of IJCAI'07*, pp. 2550–2555. 2007.

J. B. MØLLER, J. LICHTENBERG, H. R. ANDERSEN, and H. HULGAARD. 'Difference Decision Diagrams.' In *Proceedings of EACSL'99*, pp. 111–125. 1999.

A. NIVEAU, H. FARGIER, C. PRALET, and G. VERFAILLIE. 'Knowledge Compilation Using Interval Automata and Applications to Planning.' In *Proceedings of ECAI'10*, pp. 459–464. 2010.

A. NIVEAU and B. ZANUTTINI. 'Efficient Representations for the Modal Logic S5.' In *Proceedings of IJCAI'16*, pp. 1223–1229. 2016.

S. PÉRÈS, M. MORTEROL, and L. SIMON. 'SAT-Based Metabolics Pathways Analysis Without Compilation.' In *Proceedings of CMSB'14*. 2014.

K. PIPATSRISAWAT and A. DARWICHE. 'New Compilation Languages Based on Structured Decomposability.' In *Proceedings of AAAI'08*, pp. 517–522. 2008.

I. RAZGON. 'On OBDDs for CNFs of Bounded Treewidth.' In C. BARAL, G. D. GIACOMO, and T. EITER, eds., *Proceedings of KR'14*. AAAI Press, 2014.

L. SEGOUFIN. 'A glimpse on constant delay enumeration (Invited Talk).' In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, pp. 13–27. 2014.

Y. THIERRY-MIEG. 'Symbolic Model-Checking Using ITS-Tools.' In C. BAIER and C. TINELLI, eds., *Proceedings of ETAPS*, pp. 231–237. 2015.

M. B. VILAIN and H. A. KAUTZ. 'Constraint Propagation Algorithms for Temporal Reasoning.' In *Proceedings of AAAI'86*, pp. 377–382. 1986.

J. VUILLEMIN and F. BÉAL. 'On the BDD of a Random Boolean Function.' In M. J. MAHER, ed., *Proceedings of ASIAN 2004*, pp. 483–493. 2004.

F. WANG. 'Efficient verification of timed automata with BDD-like data structures.' *STTT*, 6(1):77–97, 2004.

A. WEINSTEIN and M. LITTMAN. 'Open-Loop Planning in Large-Scale Stochastic Domains.' In *Proceedings of AAAI'13*, vol. 1, p. 1436–1442. 2013.

Z. ZAMANI, S. SANNER, and C. FANG. 'Symbolic Dynamic Programming for Continuous State and Action MDPs.' In *Proceedings of AAAI'12*. 2012.